## Table of Contents

### Notational Conventions

#### A

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>aa</td>
<td>Abort acquisition with error (C)</td>
<td>38</td>
</tr>
<tr>
<td>abort</td>
<td>Terminate action of calling macro and all higher macros (C)</td>
<td>39</td>
</tr>
<tr>
<td>abortallacqs</td>
<td>Reset acquisition computer in a drastic situation (C)</td>
<td>39</td>
</tr>
<tr>
<td>abortoff</td>
<td>Terminate normal functioning of abort in a macro (C)</td>
<td>39</td>
</tr>
<tr>
<td>aborton</td>
<td>Restore normal functioning of abort in a macro (C)</td>
<td>39</td>
</tr>
<tr>
<td>abs</td>
<td>Find absolute value of a number (C)</td>
<td>39</td>
</tr>
<tr>
<td>AC1S-AC11S</td>
<td>Autocalibration macros (M)</td>
<td>40</td>
</tr>
<tr>
<td>ACbackup</td>
<td>Make backup copy of current probe file (M)</td>
<td>40</td>
</tr>
<tr>
<td>ACreport</td>
<td>Print copy of probe file after autocalibration (M)</td>
<td>40</td>
</tr>
<tr>
<td>acores</td>
<td>Find arc cosine of number (C)</td>
<td>40</td>
</tr>
<tr>
<td>acosy</td>
<td>Automatic analysis of COSY data (C)</td>
<td>40</td>
</tr>
<tr>
<td>acosyold</td>
<td>Automatic analysis of COSY data, old algorithm (C)</td>
<td>41</td>
</tr>
<tr>
<td>acqdisp</td>
<td>Display message on the acquisition status line (C)</td>
<td>41</td>
</tr>
<tr>
<td>acqi</td>
<td>Interactive acquisition display process (C)</td>
<td>41</td>
</tr>
<tr>
<td>acqmeter</td>
<td>Open Acqmeter window (M)</td>
<td>42</td>
</tr>
<tr>
<td>Acqmeter</td>
<td>Open Acqmeter window (U)</td>
<td>43</td>
</tr>
<tr>
<td>acqstat</td>
<td>Open Acquisition Status window (M)</td>
<td>44</td>
</tr>
<tr>
<td>Acqstat</td>
<td>Open Acquisition Status window (U)</td>
<td>44</td>
</tr>
<tr>
<td>acqstatus</td>
<td>Acquisition status (P)</td>
<td>45</td>
</tr>
<tr>
<td>acquire</td>
<td>Acquire data (M)</td>
<td>47</td>
</tr>
<tr>
<td>add</td>
<td>Add current FID to add/subtract experiment (C)</td>
<td>47</td>
</tr>
<tr>
<td>addAstack</td>
<td>Add stack</td>
<td>48</td>
</tr>
<tr>
<td>addfids</td>
<td>Add a series of FIDs together (M)</td>
<td>48</td>
</tr>
<tr>
<td>add</td>
<td>Start interactive add/subtract mode (C)</td>
<td>48</td>
</tr>
<tr>
<td>addnucleus</td>
<td>Add new nucleus to existing probe file (M)</td>
<td>49</td>
</tr>
<tr>
<td>addpar</td>
<td>Add selected parameters to current experiment (M)</td>
<td>49</td>
</tr>
<tr>
<td>addparams</td>
<td>Add parameter to current probe file (M)</td>
<td>51</td>
</tr>
<tr>
<td>addprobe</td>
<td>Create new probe directory and probe file (M)</td>
<td>51</td>
</tr>
<tr>
<td>addrcvrs</td>
<td>Combine data from multiple receivers (M)</td>
<td>52</td>
</tr>
<tr>
<td>adept</td>
<td>Automatic DEPT analysis and spectrum editing (C)</td>
<td>52</td>
</tr>
<tr>
<td>aexppl</td>
<td>Automatic plot of spectral expansion (M)</td>
<td>52</td>
</tr>
<tr>
<td>ai</td>
<td>Select absolute-intensity mode (C)</td>
<td>53</td>
</tr>
<tr>
<td>aig</td>
<td>Absolute-intensity group (P)</td>
<td>53</td>
</tr>
<tr>
<td>aipAnnotation</td>
<td>Annotation template name (P)</td>
<td>53</td>
</tr>
<tr>
<td>aipAutoLayout</td>
<td>Turn automatic layout on or off (P)</td>
<td>53</td>
</tr>
<tr>
<td>aipBigFrame</td>
<td>Toggle full-screen mode (C)</td>
<td>53</td>
</tr>
<tr>
<td>aipClearFrames</td>
<td>Erase all images in displayed frames (C)</td>
<td>54</td>
</tr>
<tr>
<td>aipClickedFrame</td>
<td>ID of clicked frame (P)</td>
<td>54</td>
</tr>
<tr>
<td>aipCurrentKey</td>
<td>Image key of currently drawing frame (P)</td>
<td>54</td>
</tr>
<tr>
<td>aipDeleteData</td>
<td>Unload data (C)</td>
<td>54</td>
</tr>
<tr>
<td>aipDeleteFrames</td>
<td>Clear the graphics screen (C)</td>
<td>54</td>
</tr>
<tr>
<td>aipDeleteRois</td>
<td>Delete selected ROIs (C)</td>
<td>55</td>
</tr>
<tr>
<td>aipDisplay</td>
<td>Display specified images (C)</td>
<td>55</td>
</tr>
<tr>
<td>aipDisplayByKey</td>
<td>Display a loaded image in a given frame (C)</td>
<td>56</td>
</tr>
<tr>
<td>AipDisplayMode</td>
<td>Selection mode of image display (P)</td>
<td>56</td>
</tr>
<tr>
<td>aipDupFrame</td>
<td>Move an image to another frame (C)</td>
<td>56</td>
</tr>
<tr>
<td>aipExtract</td>
<td>Extract slices from a 3D data set (C)</td>
<td>56</td>
</tr>
<tr>
<td>aipExtractMip</td>
<td>Extract MIP from a 3D data set (C)</td>
<td>57</td>
</tr>
<tr>
<td>aipGetSelectedFrames</td>
<td>Get the location and size of selected frames (C)</td>
<td>57</td>
</tr>
<tr>
<td>aipFlip</td>
<td>Reflect selected images (C)</td>
<td>57</td>
</tr>
<tr>
<td>aipGetDataKey</td>
<td>Get the key of a loaded image (C)</td>
<td>57</td>
</tr>
<tr>
<td>aipGetFrame</td>
<td>Get frame index (C)</td>
<td>58</td>
</tr>
<tr>
<td>aipGetFrameToStart</td>
<td>Get a frame to start image display (C)</td>
<td>58</td>
</tr>
<tr>
<td>aipGetHeaderParam</td>
<td>Get parameters from FDF header (C)</td>
<td>58</td>
</tr>
</tbody>
</table>
aipGetImgKey    Get image keys (C) ................................................................. 58
aipLoadDir     Load image data (C) .......................................................... 58
aipLoadFile    Load image data (C) .......................................................... 59
aipLoadRois    Load ROIs from a file to selected frames (C) .................. 59
aipMathExecute Execute an Image Math Expression (C) .......................... 59
AipMovieMode   Selection mode of movie (P) ............................................. 59
aipMovieSettings Size of movie (P) ......................................................... 59
aipNumOfCopies Get number of times an image is loaded (C) .................... 60
aipNumOfImg    Get number of loaded images (C) ..................................... 60
aipReload      Refresh image display (C) ............................................... 60
aipRotate      Rotate selected images (C) .............................................. 60
aipRQtest      Print image keys for debugging (C) ................................. 60
aipSaveHeaders Save the auxiliary header files (C) ................................. 60
aipSaveRois    Save selected ROIs to a file (C) ...................................... 61
aipSaveVs      Save intensity scaling (C) ............................................... 61
aipScreen      Query whether aip owns the graphic area (C) .................... 61
aipSegment     Segment images (C) ....................................................... 61
aipSelectFrames Select or deselect image frames (C) ............................ 62
aipSelectRois  Select or deselect ROIs (C) ........................................... 62
aipSetDebug    Enable debugging messages (C) ...................................... 62
aipSetExpression Set the image math expression template (C) ................. 62
aipSetState    Set AIP mouse state (C) ................................................... 63
aipSetVsFunction Modify intensity scaling (C) ....................................... 63
aipShow        Load and display images of a given directory (M) .......... 64
aipSomeInfoUpdate Update Point Info and Line Profile pages (C) .......... 64
aipSplitWindow Split the graphics display area into frames (C) ............... 64
aipStatPrint   Write ROI statistics to disk (C) ....................................... 65
aipStatUpdate  Update the Statistics page (C) ......................................... 65
aipWriteData   Save image data (C) ......................................................... 65
aipUpdateRQlist Update or rebuild the Review Queue list (C) ................. 66
alfa           Set alpha delay before acquisition (P) .............................. 66
aIock          Automatic lock control (P) ............................................. 66
alternateSlices Alternate slices (C) ................................................... 67
amptype        Independent control of amplifier mode (P) ...................... 67
amptype        Amplifier type (P) ........................................................... 68
analyz         Calculate standard peak height (M) .................................. 68
analyze        Generalized curve fitting (C) .......................................... 69
ap             Print out “all” parameters (C) .......................................... 70
ap             “All” parameters display control (P) .................................. 71
apa            Plot parameters automatically (M) ...................................... 71
aph            Automatic phase adjustment of spectra (C) ......................... 71
aph0           Automatic phase of zero-order term (C) ............................ 72
aphb           Auto phasing for Bruker data (C) ...................................... 72
aphx           Perform optimized automatic phasing (M) .......................... 72
appmode        Application mode (P) ..................................................... 72
apptype        Application type (P) ....................................................... 73
apt            Set up parameters for APT pulse sequence (M) .................... 73
Apt            Set up parameters for APT experiment (M) ........................ 73
APT            Change parameters for APT experiment (M) ....................... 73
aptaph         Automatic processing for APT spectra (M) ......................... 73
arccos         Calculate arc cosine of real number (M) .......................... 74
arcosin        Calculate arc sine of real number (M) ............................ 74
arctan         Calculate arc tangent of real number (M) ........................ 74
array          Easy entry of linearly spaced array values (M) ................. 75
array          Parameter order and precedence (P) ................................. 75
arraydim       Dimension of experiment (P) .......................................... 76
asin           Find arc sine of number (C) ............................................. 76
asize          Make plot resolution along f1 and f2 the same (M) .............. 76
assign         Assign transitions to experimental lines (M) ..................... 76
at             Acquisition time (P) .......................................................... 77
atan           Find arc tangent of a number (C) ....................................... 77
atan2          Find arc tangent of two numbers (C) .................................... 77
<table>
<thead>
<tr>
<th>Macro Name</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuCgrad</td>
<td>Carbon/proton gradient ratio calibration macro (M)</td>
<td>81</td>
</tr>
<tr>
<td>AuCdec</td>
<td>Carbon decoupler calibration macro (M)</td>
<td>81</td>
</tr>
<tr>
<td>AuCobs</td>
<td>Automatic Hz to DAC calibration for Z0 (M)</td>
<td>81</td>
</tr>
<tr>
<td>AuCALch3i</td>
<td>Shaped pulse information for calibration (M)</td>
<td>95</td>
</tr>
<tr>
<td>AuCALch3i1</td>
<td>B0 Magnet main static field (P)</td>
<td>95</td>
</tr>
<tr>
<td>AuCALch3ohb</td>
<td>Automatic lock gradient map generation and z0 calibration (M)</td>
<td>82</td>
</tr>
<tr>
<td>AuCALch3oh1</td>
<td>Automatic lock gradient map generation and z0 calibration (M)</td>
<td>82</td>
</tr>
<tr>
<td>AuCALch3i</td>
<td>Get autocalibration with CH3I sample (M)</td>
<td>80</td>
</tr>
<tr>
<td>Set up autocalibration with CH3I sample (M)</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Aucalibz0</td>
<td>Submit experiment to acquisition and process data (M)</td>
<td>79</td>
</tr>
<tr>
<td>Atcmd</td>
<td>Call a macro at a specified time (M)</td>
<td>78</td>
</tr>
<tr>
<td>Atext</td>
<td>Append string to current experiment text file (M)</td>
<td>79</td>
</tr>
<tr>
<td>Atval</td>
<td>Calculate pulse width (M)</td>
<td>79</td>
</tr>
<tr>
<td>Au</td>
<td>Submit experiment to acquisition and process data (M)</td>
<td>79</td>
</tr>
<tr>
<td>AuCobs</td>
<td>Carbon observe calibration macro (M)</td>
<td>81</td>
</tr>
<tr>
<td>AuCobs</td>
<td>Carbon observe calibration macro (M)</td>
<td>81</td>
</tr>
<tr>
<td>audiofilter</td>
<td>Set up and start chained acquisition (M)</td>
<td>86</td>
</tr>
<tr>
<td>Aufindz0</td>
<td>Create path for data storage (C)</td>
<td>86</td>
</tr>
<tr>
<td>AuHobs</td>
<td>Proton observe calibration macro (M)</td>
<td>83</td>
</tr>
<tr>
<td>AuHobs</td>
<td>Proton observe calibration macro (M)</td>
<td>83</td>
</tr>
<tr>
<td>Aumakegmap</td>
<td>Get parameters for a given nucleus (M)</td>
<td>83</td>
</tr>
<tr>
<td>AuNuc</td>
<td>Controlling macro for automation (M)</td>
<td>84</td>
</tr>
<tr>
<td>AuHdec</td>
<td>Preparing for an automation run (C)</td>
<td>84</td>
</tr>
<tr>
<td>AuHdec</td>
<td>Preparing for an automation run (C)</td>
<td>84</td>
</tr>
<tr>
<td>Auto</td>
<td>Automation mode active (P)</td>
<td>84</td>
</tr>
<tr>
<td>Autobackup</td>
<td>Back up current probe file (M)</td>
<td>85</td>
</tr>
<tr>
<td>autodept</td>
<td>Automated complete analysis of DEPT data (M)</td>
<td>85</td>
</tr>
<tr>
<td>autodir</td>
<td>Automation directory absolute path (P)</td>
<td>85</td>
</tr>
<tr>
<td>autogo</td>
<td>Start automation run (C)</td>
<td>85</td>
</tr>
<tr>
<td>autolist</td>
<td>Set up start chained acquisition (M)</td>
<td>86</td>
</tr>
<tr>
<td>autoname</td>
<td>Set up start chained acquisition (M)</td>
<td>86</td>
</tr>
<tr>
<td>autoname</td>
<td>Set up start chained acquisition (M)</td>
<td>86</td>
</tr>
<tr>
<td>autorora</td>
<td>Resume suspended automation run (C)</td>
<td>87</td>
</tr>
<tr>
<td>autoscale</td>
<td>Resume autoscaling after limits set by scalelimits macro (M)</td>
<td>89</td>
</tr>
<tr>
<td>autostack</td>
<td>Open Auto Test Window (C)</td>
<td>89</td>
</tr>
<tr>
<td>autotest</td>
<td>Open Auto Test Window (C)</td>
<td>89</td>
</tr>
<tr>
<td>autotime</td>
<td>Displays approximate time for automation (M)</td>
<td>89</td>
</tr>
<tr>
<td>av</td>
<td>Set abs. value mode in directly detected dimension (C)</td>
<td>90</td>
</tr>
<tr>
<td>av1</td>
<td>Set abs. value mode in 1st indirectly detected dimension (C)</td>
<td>90</td>
</tr>
<tr>
<td>av2</td>
<td>Set abs. value mode in 2nd indirectly detected dimension (C)</td>
<td>91</td>
</tr>
<tr>
<td>averag</td>
<td>Calculate average and standard deviation of input (C)</td>
<td>91</td>
</tr>
<tr>
<td>AWC</td>
<td>Additive weighting const. in directly detected dimension (P)</td>
<td>91</td>
</tr>
<tr>
<td>AWC1</td>
<td>Additive weighting const. in 1st indirectly detected dimension (P)</td>
<td>92</td>
</tr>
<tr>
<td>AWC2</td>
<td>Additive weighting const. in 2nd indirectly detected dimension (P)</td>
<td>92</td>
</tr>
<tr>
<td>axis</td>
<td>Provide axis labels and scaling factors (C)</td>
<td>92</td>
</tr>
<tr>
<td>axisf</td>
<td>Axis label for displays and plots (P)</td>
<td>92</td>
</tr>
<tr>
<td>A0604</td>
<td>Axis label for FID displays and plots (P)</td>
<td>93</td>
</tr>
<tr>
<td>B0</td>
<td>Magnet main static field (P)</td>
<td>95</td>
</tr>
<tr>
<td>B</td>
<td>Display message with large characters (C)</td>
<td>96</td>
</tr>
<tr>
<td>bandinfo</td>
<td>Display message with large characters (C)</td>
<td>96</td>
</tr>
<tr>
<td>banner</td>
<td>1D and 2D baseline correction (C)</td>
<td>96</td>
</tr>
<tr>
<td>bc</td>
<td>Turn beeper off (C)</td>
<td>96</td>
</tr>
<tr>
<td>beepoff</td>
<td>Turn beeper on (C)</td>
<td>97</td>
</tr>
<tr>
<td>beepon</td>
<td>Set up parameters for BINOM pulse sequence (M)</td>
<td>98</td>
</tr>
<tr>
<td>bootup</td>
<td>Macro executed automatically (M)</td>
<td>98</td>
</tr>
<tr>
<td>boosize</td>
<td>Magent bore size (P)</td>
<td>98</td>
</tr>
<tr>
<td>box</td>
<td>Draw a box on a plotter or graphics display (C)</td>
<td>98</td>
</tr>
<tr>
<td>boxes</td>
<td>Draw boxes selected by the mark command (M)</td>
<td>99</td>
</tr>
<tr>
<td>bpa</td>
<td>Plot boxed parameters (M)</td>
<td>100</td>
</tr>
</tbody>
</table>
br24  Set up parameters for BR24 pulse sequence (M) ......................... 104
browser  Start Image Browser application (U) ..................................... 100
bs  Block size (P) .............................................................................. 101
btune  Tune broadband channel on MERCURYplus/-Vx (M) ............... 101

c
cl13  Automated carbon acquisition (M) ............................................ 104
c13p  Process 1D carbon spectra (M) .................................................. 105
calcdim  Calculate dimension of experiment (C) .................................. 105
calfa  Recalculate alfa so that first-order phase is zero (M) ................. 105
calibflag  Correct systematic errors in DOSY experiments (P) .............. 106
calibrate  Start a dialog for autocalibration routines (M) ...................... 106
capt  Automated carbon and APT acquisition (M) .............................. 106
carbon  Set up parameters for 13C experiment (M) ............................. 107
cat  Display one or more text files in text window (C) .......................... 107
cattn  Coarse attenuator type (P) ...................................................... 107
cd  Change working directory (C) ...................................................... 107
cdc  Cancel drift correction (C) .......................................................... 107
cdept  Automated carbon and DEPT acquisition (M) ........................... 108
cdump  Prints the current graphics screen (M) ..................................... 108
celem  Completed FID elements (P) .................................................... 108
center  Set display limits for center of screen (C) .................................. 109
centersw  Move cursor to center of spectrum (M) ............................... 109
centersw1  Move cursor to center of spectrum in 1st indirect dimension (M) 109
centersw2  Move cursor to center of spectrum in 2nd indirect dimension (M) 109
cexp  Create an experiment (M) .......................................................... 109
cf  Current FID (P) ............................................................................ 110
cfpmult  Calculate first-point multiplier for 2D experiments (M) .......... 110
change  Submit a change sample experiment to acquisition (M) ........... 111
Cigar2j3j  Convert the parameter to a CIGAR2j3j experiment (M) ....... 111
cla  Clear all line assignments (M) ...................................................... 111
cla  Clear line number (M) ................................................................. 111
clamp  Clear line amplitude (M) .......................................................... 111
cleanexp  Remove old files and directories from an experiment (M) ....... 112
clear  Clear a window (C) ................................................................. 112
cleardosy  Delete temporarily saved data in current subexperiment (M) .... 112
clearStacks()  Clear stack (C) ............................................................ 112
clfreq  Calculate transition frequency (P) .......................................... 113
clindex  Index of experimental frequency of a transition (P) ................. 113
c1radd  Clear add/subtract experiment (C) ......................................... 113
color  Select plotting colors from a graphical interface (M) .................... 113
combiplate  View a color map for visual analysis of VAST microtiter plate (U) 113
combishow  Display regions (red, green, and blue) in CombiPlate window (M) 114
compressfid  Compress double-precision FID data (M, U) ..................... 114
config  Display current configuration and possibly change it (M) ........... 114
confirm  Confirm message using the mouse (C) ..................................... 119
Console  System console type (P) .......................................................... 119
contact_time  MAS cross-polarization spin-lock contact time (M) ............ 119
continueMovie  Continue movie in either forward or backward direction (C) .... 120
conv2ta  Convert imaging 3D transform to absolute value (U) .......... 120
convert  Convert data set from a VXR-style system (M, U) ..................... 120
convertBru  Convert Bruker data (M, U) ............................................. 120
copy  Copy a file (C) .......................................................................... 123
cos  Find cosine value of an angle (C) ............................................... 123
cosy  Set up parameters to a COSY pulse sequence (M) ...................... 124
Cosy  Convert the parameter to a COSY experiment (M) ................. 124
COSY  Change parameters for COSY experiment (M) .......................... 124
cosyps  Set up parameters for phase-sensitive COSY pulse sequence (M) .... 124
cp  Copy a file (C) ............................................................................ 124
cp  Cycle phase (P) ............................................................................ 124
cpmsgt2  Set up parameters for CPMGT2 pulse sequence (M) .............. 125
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpos_cvt</td>
<td>Convert data set from a VXR-style system (M,U)</td>
</tr>
<tr>
<td>cptmp</td>
<td>Copy experiment data into experiment subfile (M)</td>
</tr>
<tr>
<td>cpx</td>
<td>Create pbox shape file (M)</td>
</tr>
<tr>
<td>cr</td>
<td>Cursor position in directly detected dimension (P)</td>
</tr>
<tr>
<td>crl</td>
<td>Cursor position in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>cr2</td>
<td>Cursor position in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>crcom</td>
<td>Create user macro without using text editor (M)</td>
</tr>
<tr>
<td>create</td>
<td>Create new parameter in a parameter tree (C)</td>
</tr>
<tr>
<td>createetable</td>
<td>Generate system gradient table (M)</td>
</tr>
<tr>
<td>crf</td>
<td>Current time-domain cursor position (P)</td>
</tr>
<tr>
<td>crl1</td>
<td>Clear reference line in directly detected dimension (M)</td>
</tr>
<tr>
<td>crl2</td>
<td>Clear reference line in 1st indirectly detected dimension (M)</td>
</tr>
<tr>
<td>crmode</td>
<td>Current state of the cursors in df, ds, or dconi programs (P)</td>
</tr>
<tr>
<td>crmov</td>
<td>Remove dc offsets from FIDs in special cases (P)</td>
</tr>
<tr>
<td>decomp</td>
<td>Decompose a VXR-style directory (M)</td>
</tr>
<tr>
<td>d0</td>
<td>Overhead delay between FIDs (P)</td>
</tr>
<tr>
<td>d1</td>
<td>First delay (P)</td>
</tr>
<tr>
<td>d2</td>
<td>Incremented delay in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>d2pul</td>
<td>Set up parameters for D2PUL pulse sequence (M)</td>
</tr>
<tr>
<td>d3</td>
<td>Incremented delay in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>d4</td>
<td>Incremented delay in 3rd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>DAC_to_G</td>
<td>Store gradient calibration value in DOSY sequences (P)</td>
</tr>
<tr>
<td>da</td>
<td>Display acquisition parameter arrays (C)</td>
</tr>
<tr>
<td>da1p</td>
<td>Increment fortl dependent first-order phase correction (P)</td>
</tr>
<tr>
<td>date</td>
<td>Date (P)</td>
</tr>
<tr>
<td>daxis</td>
<td>Display horizontal LC axis (M)</td>
</tr>
<tr>
<td>Dbppste</td>
<td>Set up parameters for Dbppste pulse sequence (M)</td>
</tr>
<tr>
<td>Dbppsteinept</td>
<td>Set up parameters for Dbppsteinept pulse sequence (M)</td>
</tr>
<tr>
<td>dbset</td>
<td>Set up VnmrJ database (U)</td>
</tr>
<tr>
<td>dbupdate</td>
<td>Update the VnmrJ database (U)</td>
</tr>
<tr>
<td>dc</td>
<td>Calculate spectral drift correction (C)</td>
</tr>
<tr>
<td>dc2d</td>
<td>Apply drift correction to 2D spectra (C)</td>
</tr>
<tr>
<td>dcg</td>
<td>Drift correction group (P)</td>
</tr>
<tr>
<td>dcon</td>
<td>Display noninteractive color intensity map (C)</td>
</tr>
<tr>
<td>dconi</td>
<td>Interactive 2D data display (C)</td>
</tr>
<tr>
<td>dconi</td>
<td>Control display selection for the dconi program (P)</td>
</tr>
<tr>
<td>dconn</td>
<td>Display color intensity map without screen erase (C)</td>
</tr>
<tr>
<td>dcrmov</td>
<td>Remove dc offsets from FIDs in special cases (P)</td>
</tr>
<tr>
<td>ddf</td>
<td>Display data file in current experiment (C)</td>
</tr>
<tr>
<td>ddff</td>
<td>Display FID file in current experiment (C)</td>
</tr>
<tr>
<td>ddpf</td>
<td>Display phase file in current experiment (C)</td>
</tr>
<tr>
<td>ddif</td>
<td>Synthesize and show DOSY plot (C)</td>
</tr>
<tr>
<td>dds</td>
<td>Default display (M)</td>
</tr>
<tr>
<td>dds_seqfil</td>
<td>Sequence-specific default display (M)</td>
</tr>
<tr>
<td>debug</td>
<td>Trace order of macro and command execution (C)</td>
</tr>
<tr>
<td>deccwarnings</td>
<td>Control reporting of DECC warnings from PSG (P)</td>
</tr>
</tbody>
</table>

01-999252-00  A0604  VnmrJ 1.1D Command and Parameter Reference  7
def_osfilt Default value of osfilt parameter (P) ............................................. 149
defaultdir Default directory for Files menu system (P) .................................. 149
delcom Delete a user macro (M) ........................................................................ 150
delete Delete a file, parameter directory, or FID directory (C) .............................. 150
deleteSelected Delete selected stack or slice (C) .................................................. 150
deleteSlice Delete selected slice (C) ..................................................................... 151
delexp Delete an experiment (M) ....................................................................... 151
dels Delete spectra from \( T_1 \) or \( T_2 \) analysis (C) ........................................ 151
delta Cursor difference in directly detected dimension (P) ............................... 151
deltal Cursor difference in 1st indirectly detected dimension (P) ....................... 151
delta2 Cursor difference in 2nd indirectly detected dimension (P) .................... 152
deltaf Difference of two time-domain cursors (P) ............................................ 152
dept Set up parameters for DEPT pulse sequence (M) .................................... 152
Dept Set up parameters for DEPT experiment (M) ............................................ 152
DEPT Change parameters for DEPT experiment (M) ..................................... 152
deptgl Set up parameters for DEPTGL pulse sequence (M) ............................... 152
deptproc Process array of DEPT spectra (M) .................................................... 153
destroy Destroy a parameter (C) ...................................................................... 153
destroygroup Destroy parameters of a group in a tree (C) .................................. 153
df Display a single FID (C) ............................................................................. 154df2d Display FIDs of 2D experiment (C) .......................................................... 154
dfid Display a single FID (C) .......................................................................... 155
dfmode Current state of display of imaginary part of a FID (P) ......................... 155
dfrq Transmitter frequency of first decoupler (P) ............................................ 155
dfrq2 Transmitter frequency of second decoupler (P) ..................................... 155
dfrq3 Transmitter frequency of third decoupler (P) ...................................... 156
dfrq4 Transmitter frequency of fourth decoupler (P) ..................................... 156
dfs Display stacked FIDs (C) ........................................................................... 156
df2a Display stacked FIDs automatically (C) .................................................. 157
dfsan Display stacked FIDs automatically without screen erase (C) ............... 157
dfsh Display stacked FIDs horizontally (C) ...................................................... 157
dfshn Display stacked FIDs horizontally without screen erase (C) .................. 158
dfsn Display stacked FIDs without screen erase (C) ....................................... 158
dfww Display FIDs in whitewash mode (C) ..................................................... 158
dg Display group of acquisition/processing parameters (C) ............................ 158
dg Control dg parameter group display (P) .................................................... 159
dgl Display group of display parameters (M) .................................................. 159
dg1 Control dg1 parameter group display (P) .................................................. 159
dg2 Display group of 3rd and 4th rf channel/3D parameters (M) .................... 159
dg2 Control dg2 parameter group display (P) .................................................. 160
dga Display group of spin simulation parameters (M) ..................................... 160
DgcstecSL Set up parameters for DgcstecSL pulse sequence (M) ....................... 160
Dgcstecosy Set up parameters for Dgcstecosy pulse sequence (M) .................... 160
Dgcstehmqc Set up parameters for Dgcstehmqc pulse sequence (M) ................ 160
dglc Display group of LC-NMR parameters (M) ............................................. 161
dglc Control dglc parameter group display (P) .............................................. 161
dgm Display menu to view parameter screens (C) ............................................ 161
dgs Display group of shims and automation parameters (M) .......................... 161
dg Control dgs parameter group display (P) .................................................... 161
dhp Decoupler high-power control with class C amplifier (P) ......................... 161
dialog Display a dialog box from a macro (C) ................................................ 162
diffparams Report differences between two parameter sets (U) ...................... 162
diffshims Compare two sets of shims (M,U) .................................................... 163
digfilt Write digitally filtered FIDs to another experiment (M) ....................... 163
dir List files in directory (C) ............................................................................ 163
disCenterLines Show overlay as center lines (C) ................................................. 163
disp3d Display 3D data (U) ............................................................................ 164
display Display parameters and their attributes (C) ....................................... 164
disStripes Show overlay as stripes (C) ................................................................. 165
dla Display spin simulation parameter arrays (M) .......................................... 165
dalalong Long display of spin simulation parameter arrays (C) ....................... 165
dli Display list of integrals (C) ........................................................................ 165
dlivast  Produce text file and process wells (M)................................. 166
dll  Display listed line frequencies and intensities (C).................. 166
dlni  Display list of normalized integrals (M).............................. 167
dl  Decoupler low-power control with class C amplifier (P).............. 167
dm  Decoupler mode for first decoupler (P)................................. 167
dm2  Decoupler mode for second decoupler (P).............................. 168
dm3  Decoupler mode for third decoupler (P)............................... 168
dm4  Decoupler mode for fourth decoupler (P)......................... 168
dmf  Decoupler modulation frequency for first decoupler (P)........... 169
dmf2  Decoupler modulation frequency for second decoupler (P)...... 169
dmf3  Decoupler modulation frequency for third decoupler (P)....... 169
dmf4  Decoupler modulation frequency for fourth decoupler (P)...... 170
dmfadj  Adjust tip-angle resolution time for first decoupler (M)...... 170
dmf2adj  Adjust tip-angle resolution time for second decoupler (M) 171
dmf3adj  Adjust tip-angle resolution time for third decoupler (M) 171
dmf4adj  Adjust tip-angle resolution time for fourth decoupler (M) 171
dmg  Data display mode in directly detected dimension (P)............. 172
dmg1  Data display mode in 1st indirectly detected dimension (P).... 172
dmg2  Data display mode in 2nd indirectly detected dimension (P).... 173
dmgf  Absolute-value display of FID data or spectrum in acqi (P).... 173
dm1  Display multiple images (M).............................................. 173
dmm  Decoupler modulation mode for first decoupler (P)............... 173
dmm2  Decoupler modulation mode for second decoupler (P).......... 174
dmm3  Decoupler modulation mode for third decoupler (P)............. 174
dmm4  Decoupler modulation mode for fourth decoupler (P)........... 175
dn  Nucleus for first decoupler (P)........................................... 175
dn2  Nucleus for second decoupler (P)....................................... 175
dn3  Nucleus for third decoupler (P)......................................... 176
dn4  Nucleus for fourth decoupler (P)....................................... 176
dnode  Display list of valid limNET nodes (M,U)............................ 176
dodialog  Start a dialog window using def file (M).................... 177
dof  Frequency offset for first decoupler (P)............................. 177
dof2  Frequency offset for second decoupler (P)....................... 177
dof3  Frequency offset for third decoupler (P)........................... 177
dof4  Frequency offset for fourth decoupler (P)....................... 178
Donesth  Set up parameters for Doneshot pulse sequence (M)......... 178
dopdialog  Start a dialog with dialoglib/experiment def file (M).... 178
do_pcss  Calculate proton chemical shifts spectrum (C)............... 178
dosy  Process DOSY experiments (M)........................................ 179
dosyfrq  Gyromagnetic constant of phase encoded nucleus in DOSY (P)... 179
dosygamma  Gyromagnetic constant of phase encoded nucleus in DOSY (P) 179
dosytimecubed  Set up a T1 experiment (M)............................. 180
dot1  Display a 3D plane projection (M).................................... 180
dotflag  Display FID as connected dots (P).............................. 180
downsamp  Downsampling factor applied after digital filtering (P)....... 180
dp  Double precision (P)...................................................... 181
dpcon  Display plotted contours (C)......................................... 181
dpconn  Display plotted contours without screen erase (C)................ 182
dpf  Display peak frequencies over spectrum (C)........................ 182
dpir  Display integral amplitudes below spectrum (C)................... 182
dpirn  Display normalized integral amplitudes below spectrum (M)... 183
dp1  Default plot (M).......................................................... 183
dp1_seqfil  Sequence-specific default plot (M)............................ 183
dplane  Display a 3D plane (M).................................................. 183
dpr  Default process (M)...................................................... 184
dpr_seqfil  Sequence-specific default process (M)........................ 184
dprofile  Display pulse excitation profile (M)............................... 184
dproj  Display a 3D plane projection (M).................................... 185
dps  Display pulse sequence (C)............................................. 185
dpwr  Power level for first decoupler with linear amplifier (P)...... 186
dpwr2  Power level for second decoupler with linear amplifier (P).... 186
dpwr3  Power level for third decoupler with linear amplifier (P)........ 187
dpwr4  Power level for fourth decoupler amplifier (P)....................... 187
dpwrf  First decoupler fine power (P)............................................... 187
dpwrf2  Second decoupler fine power (P)........................................... 188
dpwrf3  Third decoupler fine power (P)............................................... 188
dpwrm  First decoupler linear modulator power (P).............................. 188
dpwrm2  Second decoupler linear modulator power (P)......................... 189
dpwrm3  Third decoupler linear modulator power (P)............................. 189
dqcosy Set up parameters for double-quantum filtered COSY (M).............. 189
Dqcosy Convert the parameter to a DQCOSY experiment (M)..................... 189
DQCOSY Change parameters for DQCOSY experiment (M)....................... 189
draw Draw line from current location to another location (C)................ 189
drawslice Display target slices (M).................................................... 190
drawvox Display target voxels (M)...................................................... 190
dres Measure linewidth and digital resolution (C)............................... 191
dres Tip-angle resolution for first decoupler (P)................................ 191
dres2  Tip-angle resolution for second decoupler (P)............................ 191
dres3  Tip-angle resolution for third decoupler (P).............................. 192
dres4  Tip-angle resolution for fourth decoupler (P)............................ 192
ds  Display a spectrum (C)............................................................... 192
ds2d  Display 2D spectra in whitewash mode (C)................................. 193
ds2dn  Display 2D spectra in whitewash mode without screen erase (C)....... 194
dscale Display scale below spectrum or FID (C)................................... 194
dscalef Digital filter coefficients for downsampling (P)......................... 195
dseq Decoupler sequence for first decoupler (P)................................ 195
dseq2 Decoupler sequence for second decoupler (P)............................ 195
dseq3 Decoupler sequence for third decoupler (P).............................. 196
dseq4 Decoupler sequence for fourth decoupler (P)............................ 196
dsfb Digital filter bandwidth for downsampling (P)............................. 196
dshape Display pulse shape or modulation pattern (M)............................ 196
dshapef Display last generated pulse shape (M)................................... 197
dshapei Display pulse shape or modulation pattern interactively (M)........ 197
dshim Display a shim "method" string (M).......................................... 197
dslsfrq Bandpass filter offset for downsampling (P)............................ 198
ds Measure signal-to-noise (C).......................................................... 198
dsn Calculate maximum signal-to-noise (M)...................................... 199
dsp Display calculated spectrum (C).................................................... 199
dsp Type of DSP for data acquisition (P).......................................... 199
dspplanes Display a series of 3D planes (M)......................................... 201
dsp Type of DSP (P).......................................................................... 201
dss  Display stacked spectra (C).......................................................... 202
dssa  Display stacked spectra automatically (C).................................. 203
dssan  Display stacked spectra automatically without erasing (C)............ 204
dssh  Display stacked spectra horizontally (C)....................................... 204
dsshn Display stacked spectra horizontally without erasing (C).............. 206
dssl Label a display of stacked spectra (M)........................................... 206
dssn  Display stacked spectra without screen erase (C).......................... 206
dsvast Display VAST data in a stacked 1D-NMR matrix format (M)............ 207
dsvast2d Display VAST data in a pseudo-2D format (M).......................... 207
dswt Display spectra in whitewash mode (C)....................................... 207
dtext Display a text file in graphics window (M)................................... 208
dtrig Delay to wait for another trigger or acquire a spectrum (P)........... 208

e Eject sample (M)............................................................................. 210
eaddr Display Ethernet address (M,U).................................................. 210
ecc Set up parameters to get eddy current compensation data (M)......... 210
ecc Put gael value and ecc file into table (M)......................................... 210
ecctabl Open eccTool window (M)....................................................... 210
echo Display strings and parameter values in text window (C).................. 211
echo Current echo index for transformed image (P)............................... 211
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>eddyout</td>
<td>Data analysis of eddy current compensation (M)</td>
<td>211</td>
</tr>
<tr>
<td>eddysend</td>
<td>Update acquisition eddy current settings (M)</td>
<td>211</td>
</tr>
<tr>
<td>edit</td>
<td>Edit a file with user-selectable editor (M)</td>
<td>212</td>
</tr>
<tr>
<td>eff_echo</td>
<td>Effective echo position in EPI experiments (P)</td>
<td>212</td>
</tr>
<tr>
<td>eject</td>
<td>Eject sample (M)</td>
<td>212</td>
</tr>
<tr>
<td>elist</td>
<td>Display directory on remote VXR-style system (M,U)</td>
<td>213</td>
</tr>
<tr>
<td>element</td>
<td>Current array index for transformed image (P)</td>
<td>213</td>
</tr>
<tr>
<td>enter</td>
<td>Enter sample information for automation run (M,U)</td>
<td>213</td>
</tr>
<tr>
<td>enterdialog</td>
<td>Start a dialog window using enterexp file (M)</td>
<td>214</td>
</tr>
<tr>
<td>epiflt</td>
<td>Process and display image in EPI experiments (M)</td>
<td>214</td>
</tr>
<tr>
<td>epiph</td>
<td>Generate phasemap file in EPI experiments (M)</td>
<td>214</td>
</tr>
<tr>
<td>epirs</td>
<td>Reverse spectral data in EPI experiments (C)</td>
<td>214</td>
</tr>
<tr>
<td>epirun</td>
<td>Collect, process, and display EPI data (M)</td>
<td>215</td>
</tr>
<tr>
<td>episet</td>
<td>Set up parameters for EPI experiments (M)</td>
<td>215</td>
</tr>
<tr>
<td>episvib</td>
<td>Save EPI images in FDF for ImageBrowser (M)</td>
<td>215</td>
</tr>
<tr>
<td>ered</td>
<td>Transfer file from remote source (M,U)</td>
<td>215</td>
</tr>
<tr>
<td>ernst</td>
<td>Calculate the Ernst angle pulse (C)</td>
<td>216</td>
</tr>
<tr>
<td>errlog</td>
<td>Display recent error messages (C)</td>
<td>216</td>
</tr>
<tr>
<td>errloglen</td>
<td>Number of lines in error message display (P)</td>
<td>216</td>
</tr>
<tr>
<td>erwrite</td>
<td>Transfer file to remote destination (M,U)</td>
<td>216</td>
</tr>
<tr>
<td>exec</td>
<td>Execute a command (C)</td>
<td>217</td>
</tr>
<tr>
<td>execpars</td>
<td>Set up the exec parameters (M)</td>
<td>217</td>
</tr>
<tr>
<td>execplot</td>
<td>Execute plotting macro (P)</td>
<td>217</td>
</tr>
<tr>
<td>execprep</td>
<td>Execute prepare macro (P)</td>
<td>217</td>
</tr>
<tr>
<td>execprescan</td>
<td>Execute prescan macro (P)</td>
<td>217</td>
</tr>
<tr>
<td>execprocess</td>
<td>Execute processing macro (P)</td>
<td>217</td>
</tr>
<tr>
<td>execsetup</td>
<td>Execute setup macro (P)</td>
<td>217</td>
</tr>
<tr>
<td>exits</td>
<td>Checks if parameter, file, or macro exists and file type (C)</td>
<td>218</td>
</tr>
<tr>
<td>exit</td>
<td>Call the vnmrexit command (M)</td>
<td>219</td>
</tr>
<tr>
<td>exp</td>
<td>Find exponential value of a number (C)</td>
<td>219</td>
</tr>
<tr>
<td>expactive</td>
<td>Determine if experiment has active acquisition (C)</td>
<td>219</td>
</tr>
<tr>
<td>expfit</td>
<td>Make least-squares fit to polynomial or exponential curve (U)</td>
<td>220</td>
</tr>
<tr>
<td>expl</td>
<td>Display exponential or polynomial curves (C)</td>
<td>222</td>
</tr>
<tr>
<td>expladd</td>
<td>Add another diffusion analysis to current display (M)</td>
<td>223</td>
</tr>
<tr>
<td>explib</td>
<td>Display experiment library (M)</td>
<td>223</td>
</tr>
<tr>
<td>explist</td>
<td>Display current experiment chain and approx. time for each (M)</td>
<td>223</td>
</tr>
<tr>
<td>explog</td>
<td>Display log file for experiment (M)</td>
<td>223</td>
</tr>
<tr>
<td>expmtime</td>
<td>Display experiment time (C)</td>
<td>223</td>
</tr>
</tbody>
</table>

**F**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>Set display parameters to full spectrum (C)</td>
<td>226</td>
</tr>
<tr>
<td>f19</td>
<td>Automated fluorine acquisition (M)</td>
<td>226</td>
</tr>
<tr>
<td>f19p</td>
<td>Process 1D fluorine spectra (M)</td>
<td>227</td>
</tr>
<tr>
<td>f1coef</td>
<td>Coefficient to construct F1 interferogram (P)</td>
<td>227</td>
</tr>
<tr>
<td>f2coef</td>
<td>Coefficient to construct F2 interferogram (P)</td>
<td>228</td>
</tr>
<tr>
<td>fattn</td>
<td>Fine attenuator (P)</td>
<td>228</td>
</tr>
<tr>
<td>fb</td>
<td>Filter bandwidth (P)</td>
<td>228</td>
</tr>
<tr>
<td>fbc</td>
<td>Apply baseline correction for each spectrum in an array (M)</td>
<td>229</td>
</tr>
<tr>
<td>fdfgluer</td>
<td>Make FDF file from header and data parts (U)</td>
<td>229</td>
</tr>
<tr>
<td>fdfspllit</td>
<td>Divide FDF file into header and data parts (U)</td>
<td>230</td>
</tr>
<tr>
<td>fdm1</td>
<td>Set, write 1D FDM parameters, run FDM (M)</td>
<td>230</td>
</tr>
<tr>
<td>fidd3d</td>
<td>3D time-domain dc correction (P)</td>
<td>231</td>
</tr>
<tr>
<td>fiddle</td>
<td>Perform reference deconvolution (M)</td>
<td>231</td>
</tr>
<tr>
<td>fiddledd</td>
<td>Perform reference deconvolution subtracting alternate FIDs (C)</td>
<td>233</td>
</tr>
<tr>
<td>fiddleeu</td>
<td>Perform reference deconvolution subtracting successive FIDs (C)</td>
<td>233</td>
</tr>
<tr>
<td>fiddle2d</td>
<td>Perform 2D reference deconvolution (C)</td>
<td>233</td>
</tr>
<tr>
<td>fiddle2dD</td>
<td>Perform 2D reference deconvolution (C)</td>
<td>233</td>
</tr>
<tr>
<td>fiddle2dd</td>
<td>2D reference deconvolution subtracting alternate FIDs (C)</td>
<td>233</td>
</tr>
<tr>
<td>fiddle2ddD</td>
<td>2D reference deconvolution subtracting alternate FIDs (C)</td>
<td>233</td>
</tr>
<tr>
<td>fddpar</td>
<td>Add parameters for FID display in current experiment (M)</td>
<td>234</td>
</tr>
<tr>
<td>fdsave</td>
<td>Save data (M)</td>
<td>234</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>fifolpsize</td>
<td>FIFO loop size (P)</td>
<td></td>
</tr>
<tr>
<td>fixgrd</td>
<td>Convert gauss/cm value to DAC (M)</td>
<td></td>
</tr>
<tr>
<td>file</td>
<td>File name of parameter set (P)</td>
<td></td>
</tr>
<tr>
<td>files</td>
<td>Interactively handle files (C)</td>
<td></td>
</tr>
<tr>
<td>filesinfo</td>
<td>Return file information for files display (C)</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Gaussian low-pass filter for image processing (M)</td>
<td></td>
</tr>
<tr>
<td>fillfile</td>
<td>File of FIR digital filter coefficients (P)</td>
<td></td>
</tr>
<tr>
<td>fitplot</td>
<td>Adjust plot parameters (M)</td>
<td></td>
</tr>
<tr>
<td>fitspec</td>
<td>Perform spectrum deconvolution (C, U)</td>
<td></td>
</tr>
<tr>
<td>fixpar</td>
<td>Correct parameter characteristics in experiment (M)</td>
<td></td>
</tr>
<tr>
<td>fixpar3rf</td>
<td>Create parameters for third rf channel (M)</td>
<td></td>
</tr>
<tr>
<td>fixpar4rf</td>
<td>Create parameters for fourth rf channel (M)</td>
<td></td>
</tr>
<tr>
<td>fixpar5rf</td>
<td>Create parameters for fifth rf channel (M)</td>
<td></td>
</tr>
<tr>
<td>fixup</td>
<td>Adjust parameter values selected by setup macros (M)</td>
<td></td>
</tr>
<tr>
<td>fxpsg</td>
<td>Update psg libraries (M)</td>
<td></td>
</tr>
<tr>
<td>flashc</td>
<td>Convert compressed 2D data to standard 2D format (C)</td>
<td></td>
</tr>
<tr>
<td>flipflop</td>
<td>Set up parameters for FLIPFLOP pulse sequence (M)</td>
<td></td>
</tr>
<tr>
<td>fliplist</td>
<td>Standard flip angle list (P)</td>
<td></td>
</tr>
<tr>
<td>Fluorine</td>
<td>Set up parameters for 19F experiment (M)</td>
<td></td>
</tr>
<tr>
<td>flush</td>
<td>Write out data in memory (C)</td>
<td></td>
</tr>
<tr>
<td>fn</td>
<td>Fourier number in directly detected dimension (P)</td>
<td></td>
</tr>
<tr>
<td>fn1</td>
<td>Fourier number in 1st indirectly detected dimension (P)</td>
<td></td>
</tr>
<tr>
<td>fn2</td>
<td>Fourier number in 2nd indirectly detected dimension (P)</td>
<td></td>
</tr>
<tr>
<td>fn2D</td>
<td>Fourier number to build up 2D DOSY display in freq. domain (P)</td>
<td></td>
</tr>
<tr>
<td>focus</td>
<td>Send keyboard focus to input window (C)</td>
<td></td>
</tr>
<tr>
<td>foldcc</td>
<td>Fold INADEQUATE data about two-quantum axis (C)</td>
<td></td>
</tr>
<tr>
<td>folddj</td>
<td>Fold J-resolved 2D spectrum about f1=0 axis (C)</td>
<td></td>
</tr>
<tr>
<td>foldt</td>
<td>Fold COSY-like spectrum along diagonal axis (C)</td>
<td></td>
</tr>
<tr>
<td>fontsselect</td>
<td>Open FontSelect window (C)</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Format a real number or convert a string for output (C)</td>
<td></td>
</tr>
<tr>
<td>fp</td>
<td>Find peak heights or phases (C)</td>
<td></td>
</tr>
<tr>
<td>fpmult</td>
<td>First point multiplier for np FID data (P)</td>
<td></td>
</tr>
<tr>
<td>fpmult1</td>
<td>First point multiplier for ni interferogram data (P)</td>
<td></td>
</tr>
<tr>
<td>fpmult2</td>
<td>First point multiplier for ni2 interferogram data (P)</td>
<td></td>
</tr>
<tr>
<td>fr</td>
<td>Full recall of a display parameter set (M)</td>
<td></td>
</tr>
<tr>
<td>fread</td>
<td>Read parameters from file and load them into a tree (C)</td>
<td></td>
</tr>
<tr>
<td>fsave</td>
<td>Save parameters from a tree to a file (C)</td>
<td></td>
</tr>
<tr>
<td>fsq</td>
<td>Frequency-shifted quadrature detection (C)</td>
<td></td>
</tr>
<tr>
<td>ft</td>
<td>Fourier transform 1D data (C)</td>
<td></td>
</tr>
<tr>
<td>ft1d</td>
<td>Fourier transform along f2 dimension (C)</td>
<td></td>
</tr>
<tr>
<td>ft1da</td>
<td>Fourier transform phase-sensitive data (M)</td>
<td></td>
</tr>
<tr>
<td>ft1dac</td>
<td>Combine arrayed 2D FID matrices (M)</td>
<td></td>
</tr>
<tr>
<td>ft2d</td>
<td>Fourier transform 2D data (C)</td>
<td></td>
</tr>
<tr>
<td>ft2da</td>
<td>Fourier transform phase-sensitive data (M)</td>
<td></td>
</tr>
<tr>
<td>ft2dac</td>
<td>Combine arrayed 2D FID matrices (M)</td>
<td></td>
</tr>
<tr>
<td>ft3d</td>
<td>Perform a 3D Fourier transform on a 3D FID data set (M,U)</td>
<td></td>
</tr>
<tr>
<td>full</td>
<td>Set display limits for a full screen (C)</td>
<td></td>
</tr>
<tr>
<td>fullsq</td>
<td>Display largest square 2D display (M)</td>
<td></td>
</tr>
<tr>
<td>fullt</td>
<td>Set display limits for a full screen with room for traces (C)</td>
<td></td>
</tr>
</tbody>
</table>

**G**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g2pul</td>
<td>Set up pulse sequence for gradient evaluation (M)</td>
</tr>
<tr>
<td>ga</td>
<td>Submit experiment to acquisition and FT the result (M)</td>
</tr>
<tr>
<td>gain</td>
<td>Receiver gain (P)</td>
</tr>
<tr>
<td>gap</td>
<td>Find gap in the current spectrum (M)</td>
</tr>
<tr>
<td>gap</td>
<td>Slice gap (P)</td>
</tr>
<tr>
<td>gaussian</td>
<td>Set up unshifted Gaussian window function (M)</td>
</tr>
<tr>
<td>gcal</td>
<td>Gradient calibration constant (P)</td>
</tr>
<tr>
<td>gcoil</td>
<td>Current gradient coil (P)</td>
</tr>
<tr>
<td>gcosy</td>
<td>Set up pulse sequence for gradient COSY (M)</td>
</tr>
<tr>
<td>gCOSY</td>
<td>Change parameters for gCOSY experiment (M)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Gcosy</td>
<td>Convert the parameter to a gradient COSY experiment (M)</td>
</tr>
<tr>
<td>gcrush</td>
<td>Crusher gradient level (P)</td>
</tr>
<tr>
<td>gdff</td>
<td>Diffusion gradient level (P)</td>
</tr>
<tr>
<td>Gdqcosy</td>
<td>Convert the parameter to a gradient DQ COSY experiment (M)</td>
</tr>
<tr>
<td>get1d</td>
<td>Select a 1D experiment for processing (M)</td>
</tr>
<tr>
<td>get2d</td>
<td>Select a 2D experiment for processing (M)</td>
</tr>
<tr>
<td>getActiveStacks</td>
<td>Get active overlay (C)</td>
</tr>
<tr>
<td>getCoronal</td>
<td>Get coronal overlay (C)</td>
</tr>
<tr>
<td>getDefaultSize</td>
<td>Get default FOV</td>
</tr>
<tr>
<td>getDefaultSlices</td>
<td>Get slices (C)</td>
</tr>
<tr>
<td>getDefaultStacks</td>
<td>Get overlay based on scout image (C)</td>
</tr>
<tr>
<td>getDefaultThk</td>
<td>Get slice thickness (C)</td>
</tr>
<tr>
<td>getdim</td>
<td>Return dimensionality of experiment (M)</td>
</tr>
<tr>
<td>getFile</td>
<td>Get information about directories and files (C)</td>
</tr>
<tr>
<td>getGapMode</td>
<td>Get gap mode (C)</td>
</tr>
<tr>
<td>getgcal</td>
<td>Get gcal value from table (M)</td>
</tr>
<tr>
<td>getll</td>
<td>Get intensity and line frequency of line (C)</td>
</tr>
<tr>
<td>getMilestoneStacks</td>
<td>Get overlay from saved parameters (C)</td>
</tr>
<tr>
<td>getparam</td>
<td>Retrieve parameter from probe file (M)</td>
</tr>
<tr>
<td>getplane</td>
<td>Extract planes from a 3D spectral data set (M)</td>
</tr>
<tr>
<td>getPrevStacks</td>
<td>Start planning with previous stacks</td>
</tr>
<tr>
<td>getreg</td>
<td>Get frequency limits of a specified region (C)</td>
</tr>
<tr>
<td>getSagittal</td>
<td>Get sagittal overlay (C)</td>
</tr>
<tr>
<td>getSlices</td>
<td>Get slices (C)</td>
</tr>
<tr>
<td>getTransverse</td>
<td>Get transverse overlay (C)</td>
</tr>
<tr>
<td>gettext</td>
<td>Get text file from VnmrJ data file (C)</td>
</tr>
<tr>
<td>getType</td>
<td>Get the type of a variable (C)</td>
</tr>
<tr>
<td>getValue</td>
<td>Get value of parameter in a tree (C)</td>
</tr>
<tr>
<td>gf</td>
<td>Prepare parameters for FID/spectrum display in acqi (M)</td>
</tr>
<tr>
<td>gf1</td>
<td>Gaussian function in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>gf2</td>
<td>Gaussian function in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>gflow</td>
<td>Flow encoding gradient level (P)</td>
</tr>
<tr>
<td>gfs</td>
<td>Gaussian shift const. in directly detected dimension (P)</td>
</tr>
<tr>
<td>gfs1</td>
<td>Gaussian function in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>gfs2</td>
<td>Gaussian function in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>gHMBC</td>
<td>Change parameters for gHMBC experiment (M)</td>
</tr>
<tr>
<td>Ghmbc</td>
<td>Convert the parameter to a gradient HMBC experiment (M)</td>
</tr>
<tr>
<td>ghmqc</td>
<td>Set up a PFG HMQC pulse sequence (M)</td>
</tr>
<tr>
<td>gHMQC</td>
<td>Set up parameters for gHMQC experiment (M)</td>
</tr>
<tr>
<td>Ghmqc</td>
<td>Convert the parameter to a gradient HMQC experiment (M)</td>
</tr>
<tr>
<td>gHMQC15</td>
<td>Set up parameters for (^{15}\text{N}) gHMQC experiment (M)</td>
</tr>
<tr>
<td>gHMQC_d2</td>
<td>Set up parameters for (^{15}\text{N}) gHMQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>gHMQC_d213</td>
<td>Set up parameters for (^{13}\text{C}) gHMQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>ghmQCTOXY</td>
<td>Set up a PFG HMQC phase-sensitive pulse sequence (M)</td>
</tr>
<tr>
<td>gHQCTOXY</td>
<td>Change parameters for gHQCTOXY experiment (M)</td>
</tr>
<tr>
<td>ghqsc</td>
<td>Set up a PFG HSQC pulse sequence (M)</td>
</tr>
<tr>
<td>ghqsc</td>
<td>Set up parameters for gHSQC experiment (M)</td>
</tr>
<tr>
<td>ghqsc</td>
<td>Convert the parameter to a gradient HSQC experiment (M)</td>
</tr>
<tr>
<td>gHSQC15</td>
<td>Set up parameters for (^{15}\text{N}) gHSQC experiment (M)</td>
</tr>
<tr>
<td>gHSQC_d2</td>
<td>Set up parameters for (^{15}\text{N}) gHSQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>gHSQC_d213</td>
<td>Set up parameters for (^{13}\text{C}) gHSQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>gHSQCTOXY</td>
<td>Set up parameters for gHSQCTOXY experiment (M)</td>
</tr>
<tr>
<td>Ghqctoxy</td>
<td>Change parameters for gradient HSQCTOXY experiment (M)</td>
</tr>
<tr>
<td>gilson</td>
<td>Open the Gilson Liquid Handler window (C)</td>
</tr>
<tr>
<td>gin</td>
<td>Return current mouse position and button values (C)</td>
</tr>
<tr>
<td>globalauto</td>
<td>Automation directory name (P)</td>
</tr>
<tr>
<td>glue</td>
<td>Create a pseudo-2D dataset (M)</td>
</tr>
<tr>
<td>gmapshim</td>
<td>Start gradient autoshimming (M)</td>
</tr>
<tr>
<td>gmapshim-au</td>
<td>Start acquisition with gradient shimming (M)</td>
</tr>
<tr>
<td>gmapsysts</td>
<td>Run gradient autoshimming, set parameters, map shims (M)</td>
</tr>
<tr>
<td>gmapz</td>
<td>Get parameters and files for gmapz pulse sequence (M)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gmap_fndtof</td>
<td>Gradient shimming flag to first find tof (P)</td>
</tr>
<tr>
<td>gmap_z14</td>
<td>Gradient shimming flag to first shim z1-z4 (P)</td>
</tr>
<tr>
<td>gmax</td>
<td>Maximum gradient strength (P)</td>
</tr>
<tr>
<td>gmqcosy</td>
<td>Set up PFG absolute-value MQF COSY parameter set (M)</td>
</tr>
<tr>
<td>gnoesy</td>
<td>Set up a PFG NOESY parameter set (M)</td>
</tr>
<tr>
<td>go_</td>
<td>Submit experiment to acquisition (M)</td>
</tr>
<tr>
<td>gpat-gpat3</td>
<td>Gradient shape (P)</td>
</tr>
<tr>
<td>gpe</td>
<td>Phase encoding gradient increment (P)</td>
</tr>
<tr>
<td>gpe2</td>
<td>2nd phase encode gradient increment</td>
</tr>
<tr>
<td>gpe3</td>
<td>3rd phase encode gradient increment</td>
</tr>
<tr>
<td>gped</td>
<td>Phase encode dephasing gradient in the EPI sequence (P)</td>
</tr>
<tr>
<td>gpmult</td>
<td>Phase encode gradient increment multiplier (P)</td>
</tr>
<tr>
<td>gplan</td>
<td>Start interactive image planning (C)</td>
</tr>
<tr>
<td>gradaxis</td>
<td>Gradient axis (P)</td>
</tr>
<tr>
<td>gradientdisable</td>
<td>Disable PFG gradients (P)</td>
</tr>
<tr>
<td>gradstpsz</td>
<td>Gradient step size (P)</td>
</tr>
<tr>
<td>gradtype</td>
<td>Gradients for X, Y, and Z axes (P)</td>
</tr>
<tr>
<td>graphis</td>
<td>Return the current graphics display status (C)</td>
</tr>
<tr>
<td>grayctr</td>
<td>Gray level window adjustment (P)</td>
</tr>
<tr>
<td>grayls1</td>
<td>Gray level slope (contrast) adjustment (P)</td>
</tr>
<tr>
<td>grecovr</td>
<td>Eddy current testing (M)</td>
</tr>
<tr>
<td>grid</td>
<td>Draw a grid on a 2D display (M)</td>
</tr>
<tr>
<td>grierate</td>
<td>Gradient rise rate (P)</td>
</tr>
<tr>
<td>gro_</td>
<td>Readout gradient strength (P)</td>
</tr>
<tr>
<td>groa</td>
<td>Readout gradient adjuster in EPI experiment (P)</td>
</tr>
<tr>
<td>grof</td>
<td>Fine tune readout gradient compensation (P)</td>
</tr>
<tr>
<td>gropat</td>
<td>Readout gradient shape (P)</td>
</tr>
<tr>
<td>gror</td>
<td>Read out compensation gradient (P)</td>
</tr>
<tr>
<td>grora</td>
<td>Readout dephasing gradient adjuster in EPI experiment (P)</td>
</tr>
<tr>
<td>groupcopy</td>
<td>Copy parameters of group from one tree to another (C)</td>
</tr>
<tr>
<td>gh2pul</td>
<td>Set up parameters for shaped gradients tests (M)</td>
</tr>
<tr>
<td>gsoil</td>
<td>Spoiler gradient level (P)</td>
</tr>
<tr>
<td>gss</td>
<td>Slice selection gradient strength (P)</td>
</tr>
<tr>
<td>gssf</td>
<td>Slice selection fractional refocusing (P)</td>
</tr>
<tr>
<td>gsspat</td>
<td>Slice-select gradient shape (P)</td>
</tr>
<tr>
<td>gssr</td>
<td>Slice selection refocusing gradient (P)</td>
</tr>
<tr>
<td>gss2,gss3</td>
<td>Slice selection gradient level (P)</td>
</tr>
<tr>
<td>gtnnoesy</td>
<td>Set up a PFG TNOESY parameter set (M)</td>
</tr>
<tr>
<td>gtnroesy</td>
<td>Set up a PFG absolute-value ROESY parameter set (M)</td>
</tr>
<tr>
<td>gtotlim</td>
<td>Gradient total limit (P)</td>
</tr>
<tr>
<td>gtrim</td>
<td>Trim gradient level (P)</td>
</tr>
<tr>
<td>gvox1-gvox3</td>
<td>Gradient strength for voxel selection (P)</td>
</tr>
<tr>
<td>gx, gy, gz</td>
<td>Gradient strength for X, Y, and Z gradients (P)</td>
</tr>
<tr>
<td>gxcal,gycal,gzcal</td>
<td>Gradient calibration constants (P)</td>
</tr>
<tr>
<td>gxmax, gymax, gmax</td>
<td>Maximum gradient strength for each axis (P)</td>
</tr>
<tr>
<td>gzlv1</td>
<td>Pulsed field gradient strength (P)</td>
</tr>
<tr>
<td>gzsiz</td>
<td>Number of z-axis shims used by gradient shimming (P)</td>
</tr>
<tr>
<td>gzwln</td>
<td>Spectral width percentage used for gradient shimming (P)</td>
</tr>
</tbody>
</table>

### H

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>Automated proton acquisition (M)</td>
<td>294</td>
</tr>
<tr>
<td>h1freq</td>
<td>Proton frequency of spectrometer (P)</td>
<td>295</td>
</tr>
<tr>
<td>h1p</td>
<td>Process 1D proton spectra (M)</td>
<td>295</td>
</tr>
<tr>
<td>h2cal</td>
<td>Calculate strength of the decoupler field (C)</td>
<td>295</td>
</tr>
<tr>
<td>halt</td>
<td>Abort acquisition with no error (C)</td>
<td>296</td>
</tr>
<tr>
<td>hc</td>
<td>Automated proton and carbon acquisition (M)</td>
<td>296</td>
</tr>
<tr>
<td>hcapt</td>
<td>Automated proton, carbon, and APT acquisition (M)</td>
<td>296</td>
</tr>
<tr>
<td>hchtocsy</td>
<td>Set up parameters for HCCHTOSY pulse sequence (M)</td>
<td>297</td>
</tr>
<tr>
<td>hccorr</td>
<td>Automated proton, carbon, and HETCOR acquisition (M)</td>
<td>297</td>
</tr>
<tr>
<td>hcddept</td>
<td>Automated proton, carbon, and DEPT acquisition (M)</td>
<td>297</td>
</tr>
</tbody>
</table>
hsqc  Automated proton and COSY acquisition (M)................................. 298
hcmult  Execute protocol actions of apptype hcmult (M).......................... 298
hdwshim  Hardware shimming (P).............................................................. 298
hdwshimlist  List of shims for hardware shimming (P).............................. 298
het2dj  Set up parameters for HET2DJ pulse sequence (M)....................... 299
HETCOR  Change parameters for HETCOR experiment (M)......................... 299
hetcor  Set up parameters for HETCOR pulse sequence (M).......................... 299
hetcorcpl  Set up parameters for solids HETCOR pulse sequence (M).......... 299
hetcorps  Set up parameters for HETCORPS pulse sequence (M)................... 299
hidecommand  Execute macro instead of command with same name (C)........... 299
hetero2d  Execute protocol actions of apptype hetero2d (M)..................... 300
Hmbc  Convert the parameter to a HMBC experiment (M).......................... 300
HMBC  Change parameters for HMBC experiment (M)................................ 300
hmqc  Set up parameters for HMQC pulse sequence (M)............................ 300
Hmqc  Convert the parameter to a HMQC experiment (M)............................ 300
HMQC  Set up parameters for HMQC experiment (M)................................ 300
HMQC_d2  Set up parameters for $^{15}$N HMQC experiment (M).................... 300
HMQC_d213  Set up parameters for $^{13}$C HMQC experiment using dec. 2 (M)... 301
hmqcrr  Set up parameters for HMQCR pulse sequence (M)........................ 301
hmqctocsy  Convert the parameter to a HMQCTOCSY pulse sequence (M).... 301
HMqctoxy  Convert the parameter to a HMQCTOXY experiment (M)............. 301
HMQCTOXY  Set up parameters for HMQCTOXY experiment (M)................... 301
HMQCTOXY15  Set up parameters for $^{15}$N HMQCTOXY experiment (M)....... 301
HMQCTOXY_d2  Set up parameters for $^{15}$N HMQCTOXY using decoupler 2 (M) 301
HMQCTOXY_d213  Set up parameters for $^{13}$C HMQCTOXY using decoupler 2 (M) 301
hmqctoxy3d  Set up parameters for HMQC-TOCSY 3D pulse sequence (M)...... 301
ho  Horizontal offset (P)...................................................................... 302
hold  Post-trigger delay (P).................................................................... 302
hom2dj  Set up parameters for HOM2DJ pulse sequence (M)...................... 302
HOMODEC  Change parameters for HOMODEC experiment (M).................. 302
homedec  Proton homonuclear decoupler present (P)................................ 302
homo  Homodecoupling control for first decoupler (P).............................. 302
homo2d  Execute protocol actions of apptype homo2d (M)....................... 303
homo2  Homodecoupling control for second decoupler (P)........................ 303
homo3  Homodecoupling control for third decoupler (P)............................ 303
homo4  Homodecoupling control for fourth decoupler (P).......................... 303
hout  Set parameters alfa and rof2 according to Houtl (M)....................... 303
hpa  Select integral regions in proton spectrum (M).................................. 304
hs  Homospoil pulses (P)........................................................................ 304
hsqc  Set up parameters for HSQC pulse sequence (M)............................. 304
Hsqc  Convert the parameter to a HSQC experiment (M)............................ 305
HSCC  Set up parameters for HSQC experiment (M).................................. 305
HSCC15  Set up parameters for $^{15}$N HSQC experiment (M)....................... 305
HSCC_d2  Set up parameters for $^{15}$N HSQC experiment using dec. 2 (M)... 305
HSCC_d213  Set up parameters for $^{13}$C HSQC experiment using dec. 2 (M)... 305
Hsqctoxy  Convert parameters to a HSQCTOXY experiment (M)................. 305
HMQCTOXY  Set up parameters for HMQCTOXY experiment (M)................... 305
HMQCTOXY15  Set up parameters for $^{15}$N HMQCTOXY experiment (M)....... 305
HMQCTOXY_d2  Set up parameters for $^{15}$N HMQCTOXY using decoupler 2 (M) 305
HMQCTOXY_d213  Set up parameters for $^{13}$C HMQCTOXY using decoupler 2 (M) 305
hsqctoxySE  Set up parameters for HSQC-TOCSY 3D pulse sequence (M)...... 305
hsrotor  Display rotor speed for solids operation (P)................................ 305
hst  Homospoil time (P).......................................................................... 306
hzmmp  Scaling factor for plots (P).......................................................... 306
hztomm  Convert locations from Hz or ppm to plotter units (C)................... 306

i

ihwinfo  Hardware status of $\text{UNITY/INOVA}$ console (U)......................... 310
il
Interleave arrayed and 2D experiments (P) .................................................. 310

ilfid
Interleave FIDs during data processing (C) ................................................. 310

image
Display noninteractive gray scale image (M) ............................................. 311

image
Control phase encoding gradient in EPI experiments (P) .......................... 311

imageprint
Plot noninteractive gray scale image (M) .................................................... 311

imark
Annotate an image display (M) ................................................................. 311

imcalc
Calculate 2D phasefiles (M,U) ................................................................. 312

imcalc
Format arguments for imcalc macro (M) .................................................... 313

imcon
Display 2D data in interactive grayscale mode (M) .................................. 313

imfit
Fit arrayed imaging data to $T_1$ or $T_2$ exponential data (M,U) ............... 313

imprep
Set up rf pulses, imaging and voxel selection gradients (M) ...................... 314

in
Lock and spin interlock (P) ....................................................................... 314

inadq
Set up parameters for INADEQUATE pulse sequence (M) ..................... 315

index2
Projection or 3D plane index selected (P) .................................................. 315

inplot
Set up parameters for INEPT pulse sequence (M) ...................................... 315

initialize_iterate
Set iterate string to contain relevant parameters (M) ................................ 315

input
Receive input from keyboard (C) ............................................................... 316

ins
Integral normalization scale (P) ................................................................. 316

ins2
2D volume value (P) ................................................................................. 316

insref
Fourier number scaled value of an integral (P) .......................................... 316

ins2ref
Fourier number scaled volume of a peak (P) ............................................. 317

insert
Insert sample (M) ..................................................................................... 317

inset
Display an inset spectrum (C) ................................................................. 317

integ
Find largest integral in a specified region (C) ............................................ 317

integrate
Automatically integrate 1D spectrum (M) ................................................ 318

intmod
Integral display mode (P) ....................................................................... 318

intvast
Produces a text file of integral regions (M) ................................................ 318

iplan
Open interactive image planning tools (M) ................................................. 318

io
Integral offset (P) ..................................................................................... 319

ir
Inversion recovery mode (P) ..................................................................... 319

is
Integral scale (P) ..................................................................................... 319

isadj
Automatic integral scale adjustment (M) ................................................. 319

isadj2
Automatic integral scale adjustment by powers of two (M) ...................... 320

iterate
Parameters to be iterated (P) .................................................................. 320

J

jdesign
Start Plot Designer Program (M) ............................................................. 321

jexp
Join existing experiment (C) ................................................................... 321

jexp1–jexp9999
Join existing experiment and display new parameters (M) ...................... 322

jplot
Plot from Plot Designer program (C) ....................................................... 322

jplotscale
Scale plot parameters (M) .................................................................... 322

jplotunscale
Restore current experiment parameters (M) ............................................ 322

jprint
Prints the selected images to a printer or file (M) ..................................... 323

jumpret
Set up parameters for JUMPRET pulse sequence (M) ............................... 323

jwin
Activate and record activity in current window (M) .................................. 323

K

killft3d
Terminate any ft3d process started in an experiment (M,U) ....................... 325

killplot
Stop plot jobs and remove from plot queue (M) ........................................ 325

killprint
Stop print jobs and remove from print queue (M) ..................................... 325

kind
Kinetics analysis, decreasing intensity (M) ............................................. 326

kinds
Kinetics analysis, decreasing intensity, short form (M) ............................. 326

kini
Kinetics analysis, increasing intensity (M) .............................................. 326

kinis
Kinetics analysis, increasing intensity, short form (M) ............................. 327

L

lastlk
Last lock solvent used (P) ......................................................................... 330

lastmenu
Menu to display when Return button is selected (P) ................................ 331

latch
Frequency synthesizer latching (P) .......................................................... 331

lb
Line broadening in directly detected dimension (P) .................................. 331
1b1  Line broadening in 1st indirectly detected dimension (P)........311
1b2  Line broadening in 2nd indirectly detected dimension (P)....322
1c1d  Pulse sequence for LC-NMR (M).................................332
1cpar2d  Create 2D LC-NMR acquisition parameters (M)........332
1ceak  Peak number (P).........................................................333
1cplot  Plot LC-NMR data (M).................................................333
1cpsgset  Set up parameters for various LC-NMR pulse sequences (M)....333
1cset2d  General setup for 2D LC-NMR experiments (M)..........333
leaf  Set display limits to left half of screen (C)....................333
legrelay  Independent control of magnet leg relay (P)..............334
length  Determine length of a string (C).................................334
list  List files in directory (C)..............................................335
llamp  Amplitudes of integral reset points (P)..........................334
llfrq  Frequencies of integral reset points (P)..........................335
listenoff  Disable receipt of messages from send2Vnmr (M).........335
listenon  Enable receipt of messages from send2Vnmr (M).........335
1kof  Track changes in lock frequency (P)...............................335
1l2d  Automatic and interactive 2D peak picking (C)..............336
1l2dbackup  Copy current 1l2d peak file to another file (M)........338
1l2dmode  Control display of peaks picked by 1l2d (P).............339
1lamp  List of line amplitudes (P)...........................................339
1lfrq  List of line frequencies (P)...........................................339
ln  Find natural logarithm of a number (C)..............................339
load  Load status of displayed shims (P).................340
loadcolors  Load colors for graphics window and plotters (M)....340
loadPrescription  Load prescription (C).................................340
loc  Location of sample in tray (P)........................................341
location  Get coordinate information from an image display (M)..........................................................341
lock  Submit an Autolock experiment to acquisition (C)...........341
lockacqtc  Lock loop time constant during acquisition (P).......342
lockfreq  Lock frequency (P)...................................................342
lockgain  Lock gain (P)..........................................................343
lockphase  Lock phase (P).......................................................343
lockpower  Lock power (P)........................................................343
locktc  Lock time constant (P)................................................343
logate  Transmitter local oscillator gate (P)............................344
lookup  Look up words and lines from a text file (C)..................344
lp  First-order phase in directly detected dimension (P)............345
lp1  First-order phase in 1st indirectly detected dimension (P)....346
lp2  First-order phase in 2nd indirectly detected dimension (P)....346
lpalg  LP algorithm in np dimension (P)..................................346
lpalg1  LP algorithm in ni dimension (P).................................347
lpalg2  LP algorithm in ni2 dimension (P).................................347
lpe  Field of view size for phase-encode axis (P).....................347
lpe2  Field of view size for 2nd phase-encode axis (P)...............348
lpeext  LP data extension in np dimension (P).........................348
lpeext1  LP data extension in ni dimension (P).........................348
lpeext2  LP data extension in ni2 dimension (P).......................348
lpfilt  LP coefficients to calculate in np dimension (P)..........349
lpfilt1  LP coefficients to calculate in ni dimension (P)..........349
lpfilt2  LP coefficients to calculate in ni2 dimension (P).........349
lpnupts  LP number of data points in np dimension (P).............349
lpnupts1  LP number of data points in ni dimension (P).............350
lpnupts2  LP number of data points in ni2 dimension (P)...........350
lpoct  LP algorithm data extension in np dimension (P)...........350
lpoct1  LP algorithm data extension in ni dimension (P)...........351
lpoct2  LP algorithm data extension in ni2 dimension (P)..........351
lprint  LP print output for np dimension (P)............................351
lprint1  LP print output for ni dimension (P)............................352
lprint2  LP print output for ni2 dimension (P)...........................352
lptrace  LP output spectrum in np dimension (P).......................352
lptrace1  LP output spectrum in ni dimension (P)......................353
make3dcoef

Make a 3D coefficients file from 2D coefficients (M) ................... 362

Edit a user macro with the vi text editor (M).................................. 361

Remove a system macro (C)........................................................... 361

List system macros (C) ................................................................... 361

Copy a system macro to become a user macro (C)......................... 361

Display a system macro file in text window (C) ............................ 360

Remove a user macro (C) ............................................................... 360

Load a macro into memory (C)....................................................... 360

Edit a macro with user-selectable editor (M).................................. 359

List user macro files (C) ................................................................. 359

Copy a user macro file (C).............................................................. 358

Display online description of command or macro (M)................... 365

Move FIDs between experiments (C). ............................................ 370

Path to user’s macro directory (P) ................................................ 358

Macro name (P)............................................................................. 358

Display a user macro file in text window (C)..................... ............... 358

Copy a system macro to become a user macro (C) ......................... 361

List system macros (C).................................................................. 361

Remove a system macro (C)......................................................... 361

Edit a macro with the vi text editor (M),............................... 361

Make a 3D coefficients file from 2D coefficients (M)............. 362

Create parameters for DOSY processing (M) ...................... 363

Make a FID element using numeric text input (C) .................. 363

Transform and save images as phasefiles (M).......................... 364

Synthesize 2D projection of 3D DOSY experiment (C).............. 364

Display online description of command or macro (M)............. 365

Update user files (U) ................................................................. 365

Path to user’s manual directory (P)................................. 365

Edit online description of a command or macro (M) .......... 365

List of experiment numbers (P) .................................................. 365

Determine intensity of spectrum at a point (C)........................... 365

Type of variable temperature system (P) ................................. 367

Maximum limit for attenuator setting for rf channel 1-4 (P)..... 368

Maximum number of pens to use (P) .............................................. 368

Maximum spectral width of Input board (P) .............................. 368

Move display parameters between experiments (C)................ 368

Change status of menu system (C)........................................... 369

Path to user’s menu directory (P)............................................ 369

Edit a menu with vi text editor (M) ............................................. 369

Autoshim method (P).................................................................. 370

Move FIDs between experiments (C)........................................... 370

Copy FID block (C).................................................................. 370

Close memory map FID (C).......................................................... 371

Move FID data (C).................................................................... 371

Memory map open FID file (C).................................................... 372

Move FID trace (C).................................................................... 373

Reduce spectral width to minimum required (M)...................... 374

Create new directory (C)............................................................ 374

Menu label (P)............................................................................ 374

Move to an absolute location to start a line (C)......................... 374

Set downsampling parameters for selected spectral region (M).... 375

Set oversampling parameters for selected spectral region (M)..... 375

Move the imaging readout position (C).................................. 375

Move spectral window according to cursors (M)......................... 376

Move spectral window according to cursors (M)......................... 376
**N**

- **n1, n2, n3**: Name storage for macros (P) ............................................................ 382
- **nactivecvrs**: Return number of receivers currently active (M) ...................... 382
- **nD**: Number of dimensions (P) ......................................................................... 382
- **ne**: Number of echoes to be acquired (P) ...................................................... 382
- **newmenu**: Select a menu without immediate activation (C) .......................... 382
- **newshm**: Interactively create a shim method with options (M) ...................... 383
- **newtpl**: Display the next 3D plane (M) ......................................................... 384
- **nf**: Number of FIDs (P) ................................................................................... 384
- **ni**: Number of increments in 1st indirectly detected dimension (P) .......... 384
- **ni2**: Number of increments in 2nd indirectly detected dimension (P) .... 384
- **ni3**: Number of increments in 3rd indirectly detected dimension (P) .... 385
- **niter**: Number of iterations (P) ....................................................................... 385
- **nl**: Number of line (C) ................................................................................... 385
- **nli**: Find integral values (C) ........................................................................... 385
- **nlivast**: Produces a text file of integral regions without a sum region (M) . 386
- **nlivast2**: Produces a text file with normalized integral regions (M) .......... 386
- **nlivast3**: Produces a text file with normalized integral regions (M) .......... 386
- **nll**: Find line frequencies and intensities (C) .................................................. 386
- **nm**: Select normalized intensity mode (C) ...................................................... 387
- **nm2d**: Select Automatic 2D normalization (M) ............................................. 387
- **noDconI**: Disable image planning (C) .............................................................. 388
- **noedit**: Convert parameters for NOE difference experiment (M) ............ 388
- **NOESY**: Change parameters for NOESY experiment (M) ......................... 388
- **Noesy**: Convert the parameter to a NOESY experiment (M) ..................... 388
- **noesy**: Set up parameters for NOESY pulse sequence (M) ......................... 388
- **NOESY1D**: Change parameters for NOESY1D experiment (M) .......... 389
- **Noesy1d**: Convert the parameter set to a NOesy1d experiment (M) ....... 389
- **noise**: Measure noise level of FID (C) ............................................................. 389
- **noisemult**: Control noise multiplier for automatic 2D processing (M) ....... 389
- **noislm**: Limit noise in spectrum (M) .............................................................. 390
- **notebook**: Notebook name (P) ....................................................................... 390
- **np**: Number of data points (P) ...................................................................... 390
- **npoint**: Number of points for fp peak search (P) ......................................... 391
- **nrecords**: Determine number of lines in a file (M) ........................................ 391
- **ns**: Number of slices to be acquired (P) ....................................................... 391
- **nscans**: Number of scout scan or real scan repetitions (P) ....................... 391
- **nt**: Number of transients (P) ......................................................................... 391
- **ntrig**: Number of trigger signals to wait before acquisition (P) .......... 392
- **ntype3d**: Specify whether f1 or f2 display expected to be N-type (P) .... 392
- **numcvrs**: Number of receivers in the system (P) ........................................... 392
- **numreg**: Return the number of regions in a spectrum (C) ............................ 392
- **numrfch**: Number of rf channels (P) .............................................................. 393
- **nv**: Number of phase encode steps (P) .......................................................... 393

**O**

- **off**: Make a parameter inactive (C) ............................................................... 395
- **offset**: Calculate frequency offset of cursor (M) ......................................... 395
- **on**: Make a parameter active or test its state (C) ........................................... 396
- **operatorlogin**: Sets workspace and parameters for the operator (M) ....... 396
- **opx**: Open shape definition file for Pbox (M) ............................................. 396
**P**

- **p1**
  - Enter pulse width for p1 in degrees (C) ........................................... 405
  - First pulse width (P) .......................................................................... 405
- **p1pat**
  - Shape of excitation pulse (P) ............................................................. 406
- **p2**
  - 180° refocus pulse width (P) ............................................................... 406
- **p2pat**
  - RF pulse pattern of 180° refocus pulse p2 (P) ...................................... 406
- **p2pul**
  - Set up sequence for PFG testing (M) .................................................. 406
- **p3**
  - Automated phosphorus acquisition (M) .............................................. 406
- **p3lp**
  - Process 1D phosphorus spectra (M) ................................................... 407
- **pa**
  - Set phase angle mode in directly detected dimension (C) ................. 407
- **pal**
  - Set phase angle mode in 1st indirectly detected dimension (C) ............ 408
- **pacosy**
  - Plot automatic COSY analysis (C) ..................................................... 408
- **pad**
  - Preacquisition delay (P) .................................................................... 409
- **padept**
  - Perform adept analysis and plot resulting spectra (C) ......................... 409
- **page**
  - Submit plot and change plotter page (C) ............................................ 410
- **page**
  - Name of page (P) ............................................................................... 410
- **panellevel**
  - Display level for VnmrJ interface pages (P) ........................................ 410
- **pap**
  - Plot out “all” parameters (C) ............................................................ 410
- **par2d**
  - Create 2D acquisition, processing, and display parameters (M) .......... 411
- **par3d**
  - Create 3D acquisition, processing, and display parameters (M) .......... 411
- **par3rf**
  - Get display templates for 3rd rf channel parameters (M) ................. 412
- **par4d**
  - Create 4D acquisition parameters (M) .............................................. 412
- **paramedit**
  - Edit a parameter and its attributes with user-selected editor (C) ......... 412
- **paramvi**
  - Edit a parameter and its attributes with vi editor (M) .......................... 412
- **pards**
  - Create additional parameters used by downsampling (M) ................... 413
- **parfidss**
  - Create parameters for time-domain solvent subtraction (M) ............... 413
- **parll2d**
  - Create parameters for LC-NMR experiments (M) .................................. 414
- **parfix**
  - Update parameter sets (M) ............................................................... 414
- **parlc**
  - Create parameters for LC-NMR experiments (M) ................................. 415
- **parl12d**
  - Create parameters for 2D peak picking (M) ....................................... 415
- **parlp**
  - Create parameters for linear prediction (M) ........................................ 415
- **parmax**
  - Parameter maximum values (P) ......................................................... 416
- **parmin**
  - Parameter minimum values (P) .......................................................... 416
- **paros**
  - Create additional parameters used by oversampling (M) .................... 416
- **parstep**
  - Parameter step size values (P) .......................................................... 417
- **parversion**
  - Version of parameter set (P) .............................................................. 417
- **path3d**
  - Path to currently displayed 2D planes from a 3D data set (P) ............... 417
- **patlist**
  - Active pulse template parameter list (P) ........................................... 418
- **paxis**
  - Plot horizontal LC axis (M) ............................................................... 418
- **Pbox**
  - Pulse shaping software (U) ............................................................... 418
- **pbox_bw**
  - Define excitation band (M) ............................................................... 420
- **pbox_bws**
  - Define excitation band for solvent suppression (notch) pulses (M) ....... 420
- **pbox_dmf**
  - Extract dmf value from pbox.cal or Pbox shape file (M) ..................... 420
- **pbox_dres**
  - Extract dres value from pbox.cal or Pbox shape file (M) ....................... 420
- **pbox_name**
  - Extract name of last shape generated by Pbox from pbox.cal (M) ....... 421
- **pbox_pw**
  - Extract pulse length from pbox.cal or Pbox shape file (M) .................... 421
- **pbox_pwr**
  - Extract power level from Pbox.cal or Pbox shape file (M) ....................... 421
- **pbox_pwrfl**
  - Extract fine power level from pbox.cal or Pbox shape file (M) ............... 421
- **pboxget**
  - Extract Pbox calibration data (M) .................................................... 422
- **pboxpar**
  - Add parameter definition to the Pbox.inp file (M) ................................ 422
- **pboxreset**
  - Reset temporary Pbox variables (M) ................................................ 423
- **pboxunits**
  - Converts to Pbox default units (M) .................................................. 423
- **pcmapclose**
  - Apply phase correction map to data in EPI experiments (C) ............... 423
- **pcmapclip**
  - Close phase correction map in EPI experiments (C) ............................ 423
- **osfb**
  - Digital filter bandwidth for oversampling (P) .................................... 397
- **oslsfrq**
  - Bandpass filter offset for oversampling (P) ......................................... 398
- **oslsfrq**
  - Oversampling filter for real-time DSP (P) .......................................... 398
- **oscoef**
  - Digital filter coefficients for oversampling (P) .................................... 397
- **osfilt**
  - Oversampling filter for real-time DSP (P) .......................................... 398

Reference: VnmrJ 1.1D Command and Parameter Reference 01-999252-00 A0604
Calculate and show proton chemical shifts spectrum (M)...................... 425
Open phase correction map in EPI experiments (C)............................. 423
Plot contours on a plotter (C).......................................................... 424
Calculate and show proton chemical shifts spectrum (M)...................... 425
Find tallest peak in specified region (C)......................................... 425
Return information about maximum in 2D data (C).......................... 426
Select a pen or color for drawing (C)............................................. 426
Plot exponential or polynomial curves (C)..................................... 427
Add another diffusion analysis to current plot (M)............................ 427
Pulsed field gradient amplifiers on/off control (P)......................... 428
Plot FIDs in whitewash mode (C)................................................. 428
Convert parameter set to PGE pulse sequence (M)......................... 428
Calibrate gradient strengths for PGE pulse sequence (M)............... 429
Extract data from single element of PGE pulse sequence (M)........... 429
Automated processing of data from PGE pulse sequence (M)........... 429
Calculate diffusion constant for integral region (M)....................... 430
Set up gradient control parameters for PGE pulse sequence (M)....... 430
Set phased mode in directly detected dimension (C)....................... 430
Set phased mode in 1st indirectly detected dimension (C)............... 431
Set phased mode in 2nd indirectly detected dimension (C)............... 431
Change frequency-independent phase rp (M).................................... 432
Phase selection (P)........................................................................ 432
Phase of first pulse (P).................................................................. 433
Phase selection for 3D acquisition (P)........................................... 433
Phase selection for 4D acquisition (P)........................................... 433
Control update region during interactive phasing (P)...................... 433
Zero-order phasing constant for the np FID (P)............................... 433
Zero-order phasing constant for ni interferogram (P)..................... 434
Zero-order phasing constant for ni2 interferogram (P).................... 434
Euler angle phi from magnet frame (P)......................................... 435
Set up parameters for $^{31}$P experiment (M)................................. 435
Inversion pulse length (P)............................................................. 435
Set up pi/3 shifted sinebell-squared window function (M).............. 435
Set up pi/4 shifted sinebell-squared window function (M).............. 436
Automatic sequence setup (P)....................................................... 436
Plots of integral regions (M).......................................................... 436
Shape of an inversion pulse (P)...................................................... 436
Plot integral amplitudes below spectrum (C).................................. 437
Plot normalized integral amplitudes below spectrum (M)............... 437
Plot spectra (C)........................................................................... 437
Plot 2D spectra in whitewash mode (C)......................................... 438
Display menu for planning a target scan (M).................................. 439
Currently displayed 3D plane type (P)......................................... 440
Planner lock (P)............................................................................ 440
Plot APT-type spectra automatically (M)................................. 440
Plotting macro for arrayed 1D spectra (M).................................... 441
Define a glue order for plotting and display (U)............................. 441
Plot a carbon spectrum (M)......................................................... 441
Plot COSY- and NOESY-type spectra automatically (M)............... 442
Plot DEPT data, edited or unedited (M)........................................ 442
Plot FIDs (C)................................................................................ 443
Plot deconvolution analysis (M).................................................... 443
Plot a grid on a 2D plot (M)......................................................... 443
Plot proton spectrum (M)............................................................... 444
Plot heteronuclear J-resolved 2D spectra automatically (M)......... 444
Plot homonuclear J-resolved 2D spectra automatically (M)........... 445
Plot X,H-correlation 2D spectrum (M).......................................... 446
Active pulse length parameter list (P).......................................... 447
Plot a line list (M)........................................................................ 447
Plot results of 2D peak picking (C)............................................. 447
Automatically plot spectra (M)...................................................... 448
01-999252-00  A0604  VnmrJ 1.1D Command and Parameter Reference  21
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>plot1d</td>
<td>Plotting macro for simple (non-arrayed) 1D spectra (M)</td>
<td>448</td>
</tr>
<tr>
<td>plot2D</td>
<td>Plot 2D spectra (M)</td>
<td>449</td>
</tr>
<tr>
<td>plotside</td>
<td>Plot spectrum on side (M)</td>
<td>449</td>
</tr>
<tr>
<td>plotter</td>
<td>Plotter device (P)</td>
<td>449</td>
</tr>
<tr>
<td>plottop</td>
<td>Plot spectrum on top (M)</td>
<td>449</td>
</tr>
<tr>
<td>plottopside</td>
<td>Plot spectrum on top and side (M)</td>
<td>450</td>
</tr>
<tr>
<td>pltplanes</td>
<td>Plot a series of 3D planes (M)</td>
<td>450</td>
</tr>
<tr>
<td>plttext</td>
<td>Plot text file (M)</td>
<td>451</td>
</tr>
<tr>
<td>pltmod</td>
<td>Plotter display mode (P)</td>
<td>451</td>
</tr>
<tr>
<td>plvast</td>
<td>Plot VAST data in a stacked 1D-NMR matrix format (M)</td>
<td>452</td>
</tr>
<tr>
<td>plvast2d</td>
<td>Plot VAST data in a stacked pseudo-2D format (M)</td>
<td>452</td>
</tr>
<tr>
<td>plww</td>
<td>Plot spectra in whitewash mode (C)</td>
<td>452</td>
</tr>
<tr>
<td>pmode</td>
<td>Processing mode for 2D data (P)</td>
<td>453</td>
</tr>
<tr>
<td>poly0</td>
<td>Display mean of the data in regression.inp file (M)</td>
<td>454</td>
</tr>
<tr>
<td>pos1=pos3</td>
<td>Position of voxel center (P)</td>
<td>454</td>
</tr>
<tr>
<td>pp</td>
<td>Decoupler pulse length (P)</td>
<td>454</td>
</tr>
<tr>
<td>ppa</td>
<td>Plot a parameter list in plain English (M)</td>
<td>455</td>
</tr>
<tr>
<td>ppcal</td>
<td>Proton decoupler pulse calibration (M)</td>
<td>455</td>
</tr>
<tr>
<td>ppe</td>
<td>Position of image center on 2D phase encode axis (P)</td>
<td>455</td>
</tr>
<tr>
<td>ppf</td>
<td>Plot peak frequencies over spectrum (C)</td>
<td>455</td>
</tr>
<tr>
<td>pph</td>
<td>Print pulse header (M)</td>
<td>456</td>
</tr>
<tr>
<td>pplvl</td>
<td>Proton pulse power level (P)</td>
<td>456</td>
</tr>
<tr>
<td>pmm</td>
<td>Resolution on printers and plotters (P)</td>
<td>457</td>
</tr>
<tr>
<td>pprofile</td>
<td>Plot pulse excitation profile (M)</td>
<td>457</td>
</tr>
<tr>
<td>pps</td>
<td>Plot pulse sequence (C)</td>
<td>457</td>
</tr>
<tr>
<td>prep</td>
<td>Prepare a scan (M)</td>
<td>458</td>
</tr>
<tr>
<td>presat</td>
<td>Set up parameters for PRESAT pulse sequence (M)</td>
<td>458</td>
</tr>
<tr>
<td>Presat</td>
<td>Set up parameters for presat (^1)H experiment (M)</td>
<td>458</td>
</tr>
<tr>
<td>presig</td>
<td>Preamplifier signal level selection (P)</td>
<td>458</td>
</tr>
<tr>
<td>prevpl</td>
<td>Display the previous 3D plane (M)</td>
<td>458</td>
</tr>
<tr>
<td>printer</td>
<td>Printer device (P)</td>
<td>459</td>
</tr>
<tr>
<td>printfile</td>
<td>Path to the print-to-file image (P)</td>
<td>459</td>
</tr>
<tr>
<td>printformat</td>
<td>Format of saved-to-file image (P)</td>
<td>459</td>
</tr>
<tr>
<td>printlayout</td>
<td>Layout of printed image (P)</td>
<td>459</td>
</tr>
<tr>
<td>printoff</td>
<td>Stop sending text to printer and start print operation (C)</td>
<td>459</td>
</tr>
<tr>
<td>printon</td>
<td>Direct text output to printer (C)</td>
<td>459</td>
</tr>
<tr>
<td>printregion</td>
<td>Screen region to be printed (P)</td>
<td>460</td>
</tr>
<tr>
<td>printsize</td>
<td>Size of printed image (P)</td>
<td>460</td>
</tr>
<tr>
<td>printsend</td>
<td>Defines where image will print (P)</td>
<td>460</td>
</tr>
<tr>
<td>pro</td>
<td>Position of image center on the readout axis (P)</td>
<td>460</td>
</tr>
<tr>
<td>probe</td>
<td>Probe type (P)</td>
<td>460</td>
</tr>
<tr>
<td>Probe_edit</td>
<td>Edit probe for specific nucleus (U)</td>
<td>460</td>
</tr>
<tr>
<td>probe_edit</td>
<td>Edit probe for specific nucleus (M)</td>
<td>461</td>
</tr>
<tr>
<td>probe_protection</td>
<td>Probe protection control (P)</td>
<td>461</td>
</tr>
<tr>
<td>proc</td>
<td>Type of processing on np FID (P)</td>
<td>461</td>
</tr>
<tr>
<td>proc1</td>
<td>Type of processing on ni interferogram (P)</td>
<td>462</td>
</tr>
<tr>
<td>proc1d</td>
<td>Processing macro for simple (non-arrayed) 1D spectra (M)</td>
<td>462</td>
</tr>
<tr>
<td>proc2</td>
<td>Type of processing on ni2 interferogram (P)</td>
<td>462</td>
</tr>
<tr>
<td>proc2d</td>
<td>Process 2D spectra (M)</td>
<td>463</td>
</tr>
<tr>
<td>procarray</td>
<td>Process arrayed 1D spectra (M)</td>
<td>463</td>
</tr>
<tr>
<td>process</td>
<td>Generic automatic processing (M)</td>
<td>464</td>
</tr>
<tr>
<td>procplot</td>
<td>Automatically process FIDs (M)</td>
<td>464</td>
</tr>
<tr>
<td>profile</td>
<td>Set up pulse sequence for gradient calibration (M)</td>
<td>465</td>
</tr>
<tr>
<td>proj</td>
<td>Project 2D data (C)</td>
<td>465</td>
</tr>
<tr>
<td>Proton</td>
<td>Set up parameters for (^1)H experiment (M)</td>
<td>465</td>
</tr>
<tr>
<td>prune</td>
<td>Prune extra parameters from current tree (C)</td>
<td>466</td>
</tr>
<tr>
<td>pscale</td>
<td>Plot scale below spectrum or FID (C)</td>
<td>466</td>
</tr>
<tr>
<td>pseudo</td>
<td>Set default parameters for pseudo-echo weighting (M)</td>
<td>467</td>
</tr>
<tr>
<td>psg</td>
<td>Display pulse sequence generation errors (M)</td>
<td>467</td>
</tr>
<tr>
<td>psggen</td>
<td>Compile a user PSG object library (M,U)</td>
<td>467</td>
</tr>
<tr>
<td>psgset</td>
<td>Set up parameters for various pulse sequences (M)</td>
<td>467</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>psg updateon</td>
<td>Enable update of acquisition parameters (C)</td>
<td></td>
</tr>
<tr>
<td>psg updateoff</td>
<td>Prevent update of acquisition parameters (C)</td>
<td></td>
</tr>
<tr>
<td>pshape</td>
<td>Plot pulse shape or modulation pattern (M)</td>
<td></td>
</tr>
<tr>
<td>pshapef</td>
<td>Plot the last created pulse shape (M)</td>
<td></td>
</tr>
<tr>
<td>psi</td>
<td>Euler angle psi from magnet frame (P)</td>
<td></td>
</tr>
<tr>
<td>pslabel</td>
<td>Pulse sequence label (P)</td>
<td></td>
</tr>
<tr>
<td>pss</td>
<td>Slice position (P)</td>
<td></td>
</tr>
<tr>
<td>pss0</td>
<td>Stack center shift along z axis (P)</td>
<td></td>
</tr>
<tr>
<td>ptext</td>
<td>Print out a text file (M)</td>
<td></td>
</tr>
<tr>
<td>ptspec3d</td>
<td>Region-selective 3D processing (P)</td>
<td></td>
</tr>
<tr>
<td>ptsval</td>
<td>PTS frequency synthesizer value (P)</td>
<td></td>
</tr>
<tr>
<td>pulse cal</td>
<td>Update and display pulse calibration data file (M)</td>
<td></td>
</tr>
<tr>
<td>pulse info</td>
<td>Shaped pulse information for calibration (M)</td>
<td></td>
</tr>
<tr>
<td>pulsetool</td>
<td>RF pulse shape analysis (U)</td>
<td></td>
</tr>
<tr>
<td>purge</td>
<td>Remove macro from memory (C)</td>
<td></td>
</tr>
<tr>
<td>put txt</td>
<td>Put text file into a data file (C)</td>
<td></td>
</tr>
<tr>
<td>put wave</td>
<td>Write a wave into Pbox.inp file (M)</td>
<td></td>
</tr>
<tr>
<td>pw</td>
<td>Pulse width (P)</td>
<td></td>
</tr>
<tr>
<td>pw90</td>
<td>90° pulse width (P)</td>
<td></td>
</tr>
<tr>
<td>pwsd</td>
<td>Display current working directory (C)</td>
<td></td>
</tr>
<tr>
<td>pwp</td>
<td>Shape of refocusing pulse (P)</td>
<td></td>
</tr>
<tr>
<td>pr</td>
<td>Set power mode in directly detected dimension (C)</td>
<td></td>
</tr>
<tr>
<td>pr1</td>
<td>Set power mode in 1st indirectly detected dimension (C)</td>
<td></td>
</tr>
<tr>
<td>pr2</td>
<td>Set power mode in 2nd indirectly detected dimension (C)</td>
<td></td>
</tr>
<tr>
<td>pr1list</td>
<td>Active pulse power level parameter list (P)</td>
<td></td>
</tr>
<tr>
<td>pr adj</td>
<td>Adjust pulse interval time (M)</td>
<td></td>
</tr>
<tr>
<td>pxecal</td>
<td>Decoupler pulse calibration (M)</td>
<td></td>
</tr>
<tr>
<td>px set</td>
<td>Assign Pbox calibration data to experimental parameters (M)</td>
<td></td>
</tr>
<tr>
<td>px shape</td>
<td>Generates a single-band shape file (M)</td>
<td></td>
</tr>
<tr>
<td>px sim</td>
<td>Simulate Bloch profile for a shaped pulse (U)</td>
<td></td>
</tr>
<tr>
<td>px spy</td>
<td>Create shape definition using Fourier coefficients (U)</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>Set up quick experiment (M)</td>
<td></td>
</tr>
<tr>
<td>QK exp</td>
<td>Tune probe using swept-tune graphical tool (C)</td>
<td></td>
</tr>
<tr>
<td>tune</td>
<td>Display individual parameter value (C)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Recall display parameter set (M)</td>
</tr>
<tr>
<td>r</td>
<td>Real-value storage for macros (P)</td>
</tr>
<tr>
<td>r1-r7</td>
<td>Resume acquisition stopped with sa command (C)</td>
</tr>
<tr>
<td>ra</td>
<td>Radial slice fan angle (P)</td>
</tr>
<tr>
<td>rcvr s</td>
<td>Which receivers to use (P)</td>
</tr>
<tr>
<td>rcvr w</td>
<td>Weighting for different receivers (P)</td>
</tr>
<tr>
<td>rcvr y</td>
<td>Pre-trigger delay (P)</td>
</tr>
<tr>
<td>react</td>
<td>Recover from error conditions during werr processing (M)</td>
</tr>
<tr>
<td>read all shims</td>
<td>Read all shims from hardware (M)</td>
</tr>
<tr>
<td>read brutape</td>
<td>Read Bruker data files from 9-track tape (U)</td>
</tr>
<tr>
<td>read file</td>
<td>Read the contents of a text file into two parameters (C)</td>
</tr>
<tr>
<td>read hw</td>
<td>Read current values of acquisition hardware (C)</td>
</tr>
<tr>
<td>read lk</td>
<td>Read current lock level (C)</td>
</tr>
<tr>
<td>read param</td>
<td>Read one of more parameters from a file (C)</td>
</tr>
<tr>
<td>read ultra</td>
<td>Read shim coil setting for Ultra•nmr shim system (M)</td>
</tr>
<tr>
<td>real</td>
<td>Create a real variable without a value (C)</td>
</tr>
<tr>
<td>recon all</td>
<td>Reconstruct images from 2D MRI fid data (C)</td>
</tr>
<tr>
<td>record</td>
<td>Record keyboard entries as a macro (M)</td>
</tr>
<tr>
<td>redor l</td>
<td>Set up parameters for REDOR1 pulse sequence (M)</td>
</tr>
<tr>
<td>redosy</td>
<td>Restore 2D DOSY display from subexperiment (M)</td>
</tr>
<tr>
<td>refresh</td>
<td>Redraw, refresh overlay (C)</td>
</tr>
<tr>
<td>reffrq</td>
<td>Reference frequency of reference line (P)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>removeAStack</td>
<td>Remove stack (C)</td>
</tr>
<tr>
<td>rename</td>
<td>Move and/or rename a file (C)</td>
</tr>
<tr>
<td>rescal</td>
<td>Calculate pixel size and spatial resolution (M)</td>
</tr>
<tr>
<td>resetf3</td>
<td>Reset parameters after a partial 3D Fourier transform (M)</td>
</tr>
<tr>
<td>resetMovie</td>
<td>Reset movie to the beginning and restore original display (C)</td>
</tr>
<tr>
<td>resto</td>
<td>NMR resonance offset frequency (P)</td>
</tr>
<tr>
<td>restoreStack</td>
<td>Restore stack (C)</td>
</tr>
<tr>
<td>return</td>
<td>Terminate execution of a macro (C)</td>
</tr>
<tr>
<td>rev</td>
<td>System software revision level (P)</td>
</tr>
<tr>
<td>revdate</td>
<td>System software preparation date (P)</td>
</tr>
<tr>
<td>rfband</td>
<td>RF band in use (P)</td>
</tr>
<tr>
<td>rfb1k</td>
<td>Reverse FID block (C)</td>
</tr>
<tr>
<td>rfchannel</td>
<td>Independent control of rf channel selection (P)</td>
</tr>
<tr>
<td>rfctype</td>
<td>Type of rf channel (P)</td>
</tr>
<tr>
<td>rfcoil</td>
<td>RF pulse calibration identity (P)</td>
</tr>
<tr>
<td>rfdata</td>
<td>Reverse FID data (C)</td>
</tr>
<tr>
<td>rf1</td>
<td>Reference peak position in directly detected dimension (P)</td>
</tr>
<tr>
<td>rf1l</td>
<td>Reference peak position in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>rf2</td>
<td>Reference peak position in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>rfp</td>
<td>Reference peak frequency in directly detected dimension (P)</td>
</tr>
<tr>
<td>rfp1</td>
<td>Reference peak freq. in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>rfp2</td>
<td>Reference peak freq. in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>rftrace</td>
<td>Reverse FID trace (C)</td>
</tr>
<tr>
<td>rtftype</td>
<td>Type of rf generation (P)</td>
</tr>
<tr>
<td>rfwg</td>
<td>RF waveform generator (P)</td>
</tr>
<tr>
<td>right</td>
<td>Set display limits to right half of screen (C)</td>
</tr>
<tr>
<td>rinput</td>
<td>Input data for a regression analysis (M)</td>
</tr>
<tr>
<td>r1</td>
<td>Set reference line in directly detected dimension (M)</td>
</tr>
<tr>
<td>r1l</td>
<td>Set reference line in 1st indirectly detected dimension (M)</td>
</tr>
<tr>
<td>r12</td>
<td>Set reference line in 2nd indirectly detected dimension (M)</td>
</tr>
<tr>
<td>rm</td>
<td>Delete file (C)</td>
</tr>
<tr>
<td>rmdir</td>
<td>Remove directory (C)</td>
</tr>
<tr>
<td>rmsAddData</td>
<td>Add transformed data files with weighting (U)</td>
</tr>
<tr>
<td>ROESY</td>
<td>Change parameters for ROESY experiment (M)</td>
</tr>
<tr>
<td>Roesy</td>
<td>Convert the parameter to a ROESY experiment (M)</td>
</tr>
<tr>
<td>roesy</td>
<td>Set up parameters for ROESY pulse sequence (M)</td>
</tr>
<tr>
<td>roesy1d</td>
<td>Convert the parameter set to a Roesy1d experiment (M)</td>
</tr>
<tr>
<td>rof1</td>
<td>Receiver gating time preceding pulse (P)</td>
</tr>
<tr>
<td>rof2</td>
<td>Receiver gating time following pulse (P)</td>
</tr>
<tr>
<td>rotate</td>
<td>Rotate 2D data (C)</td>
</tr>
<tr>
<td>rotorsync</td>
<td>Rotor synchronization (P)</td>
</tr>
<tr>
<td>rp</td>
<td>Zero-order phase in directly detected dimension (P)</td>
</tr>
<tr>
<td>rp1</td>
<td>Zero-order phase in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>rp2</td>
<td>Zero-order phase in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>RQdisplay</td>
<td>Display images selected by aipDisplayMode (M)</td>
</tr>
<tr>
<td>rqfull</td>
<td>Review Queue table width (P)</td>
</tr>
<tr>
<td>rqselection</td>
<td>Select images in the Review Queue (P)</td>
</tr>
<tr>
<td>rqsort</td>
<td>Sort images in the Review Queue (P)</td>
</tr>
<tr>
<td>rqtype</td>
<td>Review Queue type (P)</td>
</tr>
<tr>
<td>rsliceplan</td>
<td>Generate absolute magnet frame data (M)</td>
</tr>
<tr>
<td>rt</td>
<td>Retrieve FIDs (M)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>rtcmx</td>
<td>Return Spinsight data into current experiment (C)</td>
</tr>
<tr>
<td>rtp</td>
<td>Retrieve parameters (M)</td>
</tr>
<tr>
<td>rtpuff</td>
<td>Return stored phasefile to current phasefile (C)</td>
</tr>
<tr>
<td>rts</td>
<td>Retrieve shim coil settings (C)</td>
</tr>
<tr>
<td>rttmp</td>
<td>Retrieve experiment data from experiment subfile (M)</td>
</tr>
<tr>
<td>rtv</td>
<td>Retrieve individual parameters (C)</td>
</tr>
<tr>
<td>rt2</td>
<td>Retrieve parameters based on rtx rules (C)</td>
</tr>
<tr>
<td>S</td>
<td>s</td>
</tr>
<tr>
<td>s2pul</td>
<td>Set up parameters for standard two-pulse sequence (M)</td>
</tr>
<tr>
<td>sa</td>
<td>Stop acquisition (C)</td>
</tr>
<tr>
<td>sample</td>
<td>Submit change sample, Autoshim experiment to acquisition (M)</td>
</tr>
<tr>
<td>save</td>
<td>Save data (M)</td>
</tr>
<tr>
<td>savefile</td>
<td>Base file name for saving files (P)</td>
</tr>
<tr>
<td>samplename</td>
<td>Sample name (P)</td>
</tr>
<tr>
<td>saveglobal</td>
<td>Save selected parameters from global tree (P)</td>
</tr>
<tr>
<td>saveMilestoneStacks</td>
<td>Save current planning as milestone (C)</td>
</tr>
<tr>
<td>savePrescription</td>
<td>Save current planning to file (C)</td>
</tr>
<tr>
<td>sb</td>
<td>Sinebell constant in directly detected dimension (P)</td>
</tr>
<tr>
<td>sb1</td>
<td>Sinebell constant in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>sb2</td>
<td>Sinebell constant in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>abs</td>
<td>Sinebell shift in directly detected dimension (P)</td>
</tr>
<tr>
<td>abs1</td>
<td>Sinebell shift in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>abs2</td>
<td>Sinebell shift in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>sc</td>
<td>Start of chart (P)</td>
</tr>
<tr>
<td>sc2</td>
<td>Start of chart in second direction (P)</td>
</tr>
<tr>
<td>scalelimits</td>
<td>Set limits for scales in regression (M)</td>
</tr>
<tr>
<td>scalesw</td>
<td>Set scaling factor for multipulse experiments (M)</td>
</tr>
<tr>
<td>scalesw1</td>
<td>Scale spectral width in directly detected dimension (P)</td>
</tr>
<tr>
<td>scalesw1</td>
<td>Set f1 scaling factor for 2D multipulse experiments (M)</td>
</tr>
<tr>
<td>scalesw2</td>
<td>Scale spectral width in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>scalesw2</td>
<td>Scale spectral width in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>sd</td>
<td>Set first decoupler frequency to cursor position (M)</td>
</tr>
<tr>
<td>sd2</td>
<td>Set second decoupler frequency to cursor position (M)</td>
</tr>
<tr>
<td>sd3</td>
<td>Set third decoupler frequency to cursor position (M)</td>
</tr>
<tr>
<td>sda</td>
<td>Set first decoupler frequency array (M)</td>
</tr>
<tr>
<td>sd2a</td>
<td>Set second decoupler frequency array (M)</td>
</tr>
<tr>
<td>sd3a</td>
<td>Set third decoupler frequency array (M)</td>
</tr>
<tr>
<td>sdp</td>
<td>Show diffusion projection (M)</td>
</tr>
<tr>
<td>sediff</td>
<td>Set up spin-echo diffusion imaging sequence (M)</td>
</tr>
<tr>
<td>selld</td>
<td>Execute protocol actions of apttype selld (M)</td>
</tr>
<tr>
<td>select</td>
<td>Select spectrum, FID, trace, or 2D plane without display (C)</td>
</tr>
<tr>
<td>selext</td>
<td>Defines excitation band (M)</td>
</tr>
<tr>
<td>sens</td>
<td>Set up PFG selective excitation pulse sequence (M)</td>
</tr>
<tr>
<td>send2vnmr</td>
<td>Send a command to VnmrJ (U)</td>
</tr>
<tr>
<td>seqcon</td>
<td>Acquisition loop control (P)</td>
</tr>
<tr>
<td>seqfil</td>
<td>Pulse sequence name (P)</td>
</tr>
<tr>
<td>seqgen</td>
<td>Initiate compilation of user's pulse sequence (M,U)</td>
</tr>
<tr>
<td>set2D</td>
<td>General setup for 2D experiments (M)</td>
</tr>
<tr>
<td>set2d</td>
<td>General setup for 2D experiments (M)</td>
</tr>
<tr>
<td>set3dproc</td>
<td>Set 3D processing (C)</td>
</tr>
<tr>
<td>setallshims</td>
<td>Set all shims into hardware (M)</td>
</tr>
<tr>
<td>setarray</td>
<td>Set up a parameter array (M)</td>
</tr>
<tr>
<td>setcenter</td>
<td>Set up parameters for center sequence calibration (M)</td>
</tr>
<tr>
<td>setcolor</td>
<td>Set colors for graphics window and for plotters (C)</td>
</tr>
<tr>
<td>setdec2pars</td>
<td>Set decoupler parameter values from probe file (C)</td>
</tr>
<tr>
<td>setdec2pars</td>
<td>Set decoupler 2 parameter values from probe file (M)</td>
</tr>
<tr>
<td>setDefaultSize</td>
<td>Set FOV to default size (C)</td>
</tr>
<tr>
<td>setDefaultSlices</td>
<td>Set default number of slices (C)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>shell</td>
<td>Start an interactive UNIX shell (C)</td>
</tr>
<tr>
<td>shim</td>
<td>Submit an Autoshim experiment to acquisition (C)</td>
</tr>
<tr>
<td>shimset</td>
<td>Type of shim set (P)</td>
</tr>
<tr>
<td>shimpath</td>
<td>Path to user’s shims directory (P)</td>
</tr>
<tr>
<td>showconsole</td>
<td>Show UNITY/INOVA console configuration parameters (U)</td>
</tr>
<tr>
<td>showfit</td>
<td>Display numerical results of deconvolution (M)</td>
</tr>
<tr>
<td>showloginbox</td>
<td>Shows operator login dialog (M)</td>
</tr>
<tr>
<td>showoriginal</td>
<td>Restore first 2D spectrum in 3D DOSY experiment (M)</td>
</tr>
<tr>
<td>showplotter</td>
<td>Show list of currently defined plotters and printers (M)</td>
</tr>
<tr>
<td>showplotq</td>
<td>Display plot jobs in plot queue (M)</td>
</tr>
<tr>
<td>showprintq</td>
<td>Display print jobs in print queue (M)</td>
</tr>
<tr>
<td>showstat</td>
<td>Display information about status of acquisition (M,U)</td>
</tr>
<tr>
<td>sin</td>
<td>Find sine value of an angle (C)</td>
</tr>
<tr>
<td>sine</td>
<td>Find values for a sine window function (M)</td>
</tr>
<tr>
<td>sinebell</td>
<td>Select default parameters for sinebell weighting (M)</td>
</tr>
<tr>
<td>sinesq</td>
<td>Find values for a sine-squared window function (M)</td>
</tr>
<tr>
<td>size</td>
<td>Returns the number of elements in an arrayed parameter (O)</td>
</tr>
<tr>
<td>slfreq</td>
<td>Measured line frequencies (P)</td>
</tr>
<tr>
<td>sliceorder</td>
<td>Reorder the slice position list (M)</td>
</tr>
<tr>
<td>sliceplan</td>
<td>Set slice parameters for target slice (M)</td>
</tr>
<tr>
<td>slp</td>
<td>Family of offset Frequencies of SLP shapes (P)</td>
</tr>
<tr>
<td>slw</td>
<td>Spin simulation linewidth (P)</td>
</tr>
<tr>
<td>smaxf</td>
<td>Maximum frequency of any transition (P)</td>
</tr>
<tr>
<td>sminf</td>
<td>Minimum frequency of any transition (P)</td>
</tr>
<tr>
<td>smsport</td>
<td>Sample Management System serial port connection (P)</td>
</tr>
<tr>
<td>sn</td>
<td>Signal-to-noise ratio (P)</td>
</tr>
<tr>
<td>solppm</td>
<td>Return ppm and peak width of solvent resonances (M)</td>
</tr>
<tr>
<td>solvent</td>
<td>Lock solvent (P)</td>
</tr>
<tr>
<td>solvinfo</td>
<td>Retrieve information from solvent table (C)</td>
</tr>
<tr>
<td>sort</td>
<td>Sort real values of a parameter (M)</td>
</tr>
<tr>
<td>sp</td>
<td>Start of plot in directly detected dimension (P)</td>
</tr>
<tr>
<td>sp1</td>
<td>Start of plot in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>sp2</td>
<td>Start of plot in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>spadd</td>
<td>Add current spectrum to add/subtract experiment (C)</td>
</tr>
<tr>
<td>spcfrq</td>
<td>Display frequencies of rf channels (M)</td>
</tr>
<tr>
<td>specdc3d</td>
<td>3D spectral dc correction (P)</td>
</tr>
<tr>
<td>spin</td>
<td>Submit a spin setup experiment to acquisition (C)</td>
</tr>
<tr>
<td>spinopt</td>
<td>Spin automation (P)</td>
</tr>
<tr>
<td>spins</td>
<td>Perform spin simulation calculation (C)</td>
</tr>
<tr>
<td>split</td>
<td>Split difference between two cursors (M)</td>
</tr>
<tr>
<td>spmax</td>
<td>Take the maximum of two spectra (C)</td>
</tr>
<tr>
<td>spmin</td>
<td>Take minimum of two spectra in add/subtract experiment (C)</td>
</tr>
<tr>
<td>spsm</td>
<td>Enter spin system (M)</td>
</tr>
<tr>
<td>spsub</td>
<td>Subtract current spectrum from add/subtract experiment (C)</td>
</tr>
<tr>
<td>sqcosine</td>
<td>Set up unshifted cosine-squared window function (M)</td>
</tr>
<tr>
<td>sqdir</td>
<td>Study queue directory (P)</td>
</tr>
<tr>
<td>sqname</td>
<td>Study queue parameter template (P)</td>
</tr>
<tr>
<td>sqrt</td>
<td>Return square root of a real number (O)</td>
</tr>
<tr>
<td>sqsinebell</td>
<td>Set up unshifted sinebell-squared window function (M)</td>
</tr>
<tr>
<td>srate</td>
<td>Spinning rate for magic angle spinning (P)</td>
</tr>
<tr>
<td>sread</td>
<td>Read converted data into VnmrJ (C)</td>
</tr>
<tr>
<td>ss</td>
<td>Steady-state transients (P)</td>
</tr>
<tr>
<td>ssecho</td>
<td>Set up solid-state echo pulse sequence (M)</td>
</tr>
<tr>
<td>ssechol</td>
<td>Set up parameters for SSECHO1 pulse sequence (M)</td>
</tr>
<tr>
<td>ssfilter</td>
<td>Full bandwidth of digital filter to yield a filtered FID (P)</td>
</tr>
<tr>
<td>sslsfrq</td>
<td>Center of solvent-suppressed region of spectrum (P)</td>
</tr>
<tr>
<td>ssntaps</td>
<td>Number of coefficients in digital filter (P)</td>
</tr>
</tbody>
</table>
ssorder
Set slice parameters for target slice (M)............. 581
ssplan
Order of polynomial to fit digitally filtered FID (P)........ 581
sslist
Set up volume localized spectroscopy sequence (M)............. 581
ssprep
Calculate slice gradient and slice selection parameters (M)..... 581
stack
Stacking mode for processing and plotting arrayed spectra (M).... 582
stackmode
Stacking control for processing arrayed 1D spectra (P)............. 583
startplan
Start/restart image planning (C)......................................... 581
startMovie
Start running a movie (C)........................................................ 583
status
Display status of sample changer (C,U)................................. 583
std1d
Execute protocol actions of apptype std1d (M)..................... 584
stdshm
Interactively create a method string for autoshimming (M)........ 584
steam
Set up volume localized spectroscopy sequence (M).................. 584
stepMovie
Step one frame in a movie (C).................................................. 584
sw1
Spectral width in 1st indirectly detected dimension (P).............. 595
sw2
Spectral width in 2nd indirectly detected dimension (P).............. 596
sw3
Spectral width in 3rd indirectly detected dimension (P).............. 596
sysgcoil
System gradient coil (P).................................................................. 596
system
type
System type (P)............................................................................... 596
systemdir
VnmrJ system directory (P).......................................................... 597
sslist
Conjugate gradient list (P).............................................................. 581
ssprep
Set up volume localized spectroscopy sequence (M)............. 581
steam
Set up volume localized spectroscopy sequence (M).................. 584
stepMovie
Step one frame in a movie (C).................................................. 584
sw1
Spectral width in 1st indirectly detected dimension (P).............. 595
sw2
Spectral width in 2nd indirectly detected dimension (P).............. 596
sw3
Spectral width in 3rd indirectly detected dimension (P).............. 596
sysgcoil
System gradient coil (P).................................................................. 596
system
type
System type (P)............................................................................... 596
systemdir
VnmrJ system directory (P).......................................................... 597
T
T1
T1 exponential analysis (M)........................................................ 597
t1image
Fit arrayed imaging data to T1 exponential data (M)................. 597
t1s
T1 exponential analysis with short output table (M)............... 597
t2
T2 exponential analysis (M)........................................................... 597
t2image
Fit arrayed imaging data to T2 exponential data (M)................. 597
t2s
T2 exponential analysis with short output table (M)............... 597
tabc
Convert data in table order to linear order (M)...................... 602
tan
Find tangent value of an angle (C).......................................... 603
tape
Read tapes from VXR-style system (M,U)................................. 603
tape
Control tape options of files program (P).............................. 604
tbox
Draw a tilted box (C)..................................................................... 604
tcapply
Apply table conversion reformating to data (C)....................... 605
tcclose
Close table conversion file (C).................................................. 605
tcl
Send Tcl script to Tcl version of dg window (C) ........................................... 606
tcopen
Open table conversion file (C)...................................................................... 606
te
Echo time (P) ............................................................................................ 606
techron
Set up parameters for gradient amplifier tests (M) ..................................... 606
temp
Open the Temperature Control window (C) .............................................. 607
temp
Sample temperature (P) ............................................................................ 607	tempcal
Temperature calculation (C) ................................................................. 607
tep
Post-acquisition delay in EPI experiments (P) ............................................. 608
testct
Check ct for resuming signal-to-noise testing (M) ....................................... 608
testsn
Test signal-to-noise of a spectrum (M) ...................................................... 608
teststr
Find which array matches a string (M) ...................................................... 609
text
Display text or set new text for current experiment (C) ............................. 609
textis
Return the current text display status (C) ................................................. 610
textvi
Edit text file of current experiment (M) .................................................... 610
th
Threshold (P) ............................................................................................ 610
th2d
Threshold for integrating peaks in 2D spectra (P) ......................................... 611
thadj
Adjust threshold for peak printout (M) ..................................................... 611
theta
Euler angle theta from magnet frame (P) .................................................... 611
thk
Slice thickness (P) .................................................................................... 612
ti
Inversion recovery time (P) ....................................................................... 612
ticks
Number of trigger pulses (P) .................................................................... 612
time
Display experiment time or recalculate number of transients (M) ............... 612
tin
Temperature interlock (P) ....................................................................... 613
title
Plot a title on a plotter (M) ...................................................................... 613
tlt
First-order baseline correction (P) ........................................................... 613
tmove
Left-shift FID to time-domain cursor (M) ................................................ 614
tmsref
Reference 1D proton or carbon spectrum to TMS (M) ................................ 614
tn
Nucleus for observe transmitter (P) ............................................................ 614
tncosyps
Set up parameters for TNCOSYPS pulse sequence (M) ............................... 614
tndqcosy
Set up parameters for TNDQCOSY pulse sequence (M) ............................... 614
tnmqcosy
Set up parameters for TNMQCOSY pulse sequence (M) ............................ 615
tnnoesy
Set up parameters for TNNOE SY pulse sequence (M) ................................. 615
troesy
Set up parameters for TROE SY pulse sequence (M) .................................. 615
TOCSY
Change parameters for TOCSY experiment (M) ........................................ 615
Tocsy
Convert the parameters to a TOCSY experiment (M) .................................. 615
tocsy
Set up parameters for TOCSY pulse sequence (M) ...................................... 615
Tocsy1D
Convert the parameter set to a Tocsy1D experiment (M) ............................ 616
TOCSY1D
Change parameters for TOCSY1D experiment (M) .................................... 616
tof
Frequency offset for observe transmitter (P) ............................................. 616
tpe
Duration of the phase encoding gradient pulse (P) ....................................... 616
tpe2,tpe3
Duration of second and third phase encoding gradient periods (P) .......... 616
tpwr
Observe transmitter power level with linear amplifiers (P) .......................... 617
tpwr1
Intensity of an excitation pulse (P) ............................................................ 617
tpwr2
Intensity of an excitation pulse (P) ............................................................ 617
tpwrcale
Calibrate power levels of 90° and 180° pulse (M) ....................................... 617
tpwrfr
Observe transmitter fine power (P) ............................................................ 618
tpwi
Intensity of inversion pulse (P) ................................................................. 618
tpwrnl
Observe transmitter linear modulator power (P) ........................................ 619
tr
Repetition time in imaging and localized spectroscopy (P) .......................... 619	race
Mode for n-dimensional data display (P) .................................................... 619
transfer
Move parameters to target experiment (M) ................................................. 619
traymax
Sample changer tray slots (P) ................................................................. 620
trfunc
Translate screen coordinates (M) ............................................................. 621
trfuncd
Translate screen distance (M) ................................................................. 621
trise
Gradient rise time (P) ........................................................... 621
troesy
Set up parameters for TROE SY pulse sequence (M) .................................. 622
trunc
Truncate real numbers (O) ............................................................... 622
tshift
Adjust tau2 to current cursor position (M) ................................................. 622
tspoil
Gradient spoiling time (P) ............................................................... 622
tugain
Amount of receiver gain used by tqu e (P) .................................................. 623
tune
Assign a frequency to a channel for probe tuning (C) ............................... 623
**U**

- **undospins**
  - Restore spin system as before last iterative run (M)..................625
- **undosy**
  - Restore original 1D NMR data from subexperiment (M)..............625
- **unit**
  - Define conversion units (C)..................................................625
- **unlock**
  - Remove inactive lock and join experiment (C)........................626
- **updatepars**
  - Update all parameter sets saved in a directory (M).................627
- **updateprobe**
  - Update probe file (M)..............................................................627
- **updaterev**
  - Update after installing new VnmrJ version (M).......................628
- **updtgcoil**
  - Update gradient coil (M).........................................................628
- **updpqparam**
  - Update specified acquisition parameters (C)..........................628
- **usemark**
  - Use “mark” output as deconvolution starting point (M)............628
- **userdir**
  - VnmrJ user directory (P).........................................................629
- **usergo**
  - Macro called by fixpar (M)......................................................629
- **userfixpar**
  - Macro called by fixpar (M)......................................................629
- **usersel**
  - Selection for images and frames (P).......................................629

**V**

- **vast1d**
  - Set up initial parameters for VAST experiments (M)..................631
- **vastget**
  - Selects and displays VAST spectra (M)....................................632
- **vastglue**
  - Assemble 1D datasets into a 2D (or pseudo-2D) dataset (M)........632
- **vastglue2**
  - Assemble 1D datasets into a 2D (or pseudo-2D) dataset (M)........632
- **vastgo**
  - Turn off LC stop flow automation, start VAST automation (M)...633
- **vbg**
  - Run VNMR processing in background (U)................................633
- **vf**
  - Vertical scale of FID (P).........................................................633
- **vi**
  - Edit text file with vi text editor (M)........................................634
- **vjsel**
  - Display VnmrJ help (U)..............................................................636
- **vn**
  - Start VNMR directly (U)............................................................636
- **vnmr**
  - Start VNMR in current windowing system (U).............................636
- **vnmr2sc**
  - VNMR to SpinCAD pulse sequence translator (M)......................637
- **vnmr_accounting**
  - Open Accounting window (U).....................................................637
- **vnmrexit**
  - Exit from the VNMR system (C)................................................638
- **vnmrj**
  - Start VnmrJ (U)..........................................................................638
- **vnmrplot**
  - Plot files (U)...........................................................................638
- **vnmrprint**
  - Print text files (U).................................................................638
- **vo**
  - Vertical offset (P)...................................................................639
- **vorient**
  - Voxel orientation....................................................................639
- **vox1 - vox3**
  - Voxel dimensions (P)...............................................................639
- **voxplan**
  - Set voxel parameters for voxel defined by 2D box cursor (M).....640
- **vp**
  - Vertical position of spectrum (P)..............................................640
- **vphi,vpsi,vtheta**
  - Euler angles for voxel orientation............................................641
- **vs**
  - Vertical scale (P)......................................................................641
- **vs2d**
  - Vertical scale for 2D displays (P)..............................................641
- **vsadj**
  - Automatic vertical scale adjustment (M)..................................642
- **vsadj2**
  - Automatic vertical scale adjustment by powers of 2 (M)............642
- **vsadjc**
  - Automatic vertical scale adjustment for $^{13}$C spectra (M)......642
- **vsadjh**
  - Automatic vertical scale adjustment for $^1$H spectra (M).........643
- **vsproj**
  - Vertical scale for projections and traces (P)............................643
- **vtc**
  - Variable temperature cutoff point (P).......................................643
- **vtttype**
  - Variable temperature controller present (P).............................644
- **vtwait**
  - Variable temperature wait time (P)..........................................644
- **vxxr_unix**
  - Convert VXR-style text files to UNIX format (M,U)....................644

**W**

- **who**
  - Who is using system (C).............................................................648
- **walkup**
  - Walkup automation (M)..............................................................648
waltz
Specify action when waltz decoupling present (P).................................648
wbs
Specify action when bs transients accumulate (C).................................648
wbs
When block size (P) ............................................................................649
wc
Width of chart (P) ................................................................................649
wc2
Width of chart in second direction (P)....................................................649
wcmax
Maximum width of chart (P) ..................................................................650
wc2max
Maximum width of chart in second direction (P).....................................650
werr
Specify action when error occurs (C)....................................................650
werr
When error (P) .....................................................................................650
wet
flag to turn on or off wet solvent suppression ((P) ..............................651
wet1d
Set up parameters for a WET1D pulse sequence (M) .........................651
wetdqcossy
Set up parameters for a WETDCQOSY pulse sequence (M).................651
wetgcosy
Set up parameters for a WETGCOUYY pulse sequence (M)...............651
wetghmqqos
Set up parameters for a WETGHMQCPS pulse sequence (M).............651
wetgmmqcosy
Set up parameters for a WETGHSQC pulse sequence (M)..................651
wetnoesyy
Set up parameters for a WETNOESY pulse sequence (M)..................652
wetpmwctal
Set up parameters for a WETPWXCAL pulse sequence (M)................652
wetntntcosy
Set up parameters for a WETNTNTCSY pulse sequence (M).............652
wetshape
Shape for pwwet pulses (P) ................................................................652
wexp
Specify action when experiment completes (C)..................................652
wexp
When experiment completes (P) ..........................................................653
wf
Width of FID (P)..................................................................................653
wf1
Width of interferogram in 1st indirectly detected dimension (P) ...653
wf2
Width of interferogram in 2nd indirectly detected dimension (P)......654
wfgtest
Waveform generator test (M)...............................................................654
wft
Weight and Fouier transform 1D data (C)............................................654
wft1d
Weight and Fouier transform f1 for 2D data (C)..................................654
wft1da
Weight and Fouier transform phase-sensitive data (M).....................655
wft1dac
Combine arrayed 2D FID matrices (M)..............................................655
wft2d
Weight and Fouier transform 2D data (C)............................................655
wft2da
Weight and Fouier transform phase-sensitive data (M).....................656
wft2dac
Combine arrayed 2D FID matrices (M)..............................................656
wft3
Process f3 dimension during 3D acquisition (M)...............................656
which
Display which command or macro is used (M)..................................657
wnt
Specify action when nt transients accumulate (C)..............................657
wnt
When number of transients (P) ..........................................................658
wp
Width of plot in directly detected dimension (P).................................658
wp1
Width of plot in 1st indirectly detected dimension (P).........................658
wp2
Width of plot in 2nd indirectly detected dimension (P).........................658
write
Write formatted text to a device (C)..................................................658
writefid
Write numeric text file using a FID element (C) .................................660
writeparam
Write one of more parameters to a file (C) ........................................660
wrtp
Command string executed after rtp command (P) ...............................661
wsram
Send hardware configuration to acquisition console (C).....................661
wshim
Conditions when shimming is performed (P).....................................661
wtfle
User-defined weighting in directly detected dimension (P) ...............662
wtfle1
User-defined weighting in 1st indirectly detected dimension (P)......662
wtfle2
User-defined weighting in 2nd indirectly detected dimension (P)......662
wtgen
Compile user-written weighting functions (M,U) ..............................663
wti
Interactive weighting (C) .................................................................663
wtia
Interactive weighting for 2D absorptive data (M) ...............................664
wysiwyg
Set plot display or full display (P) .....................................................664

X
x0
X-zero position of HP pen plotter or Postscript device (P) ...............665
x1
X1 shim gradient (P) ...........................................................................665
x2y2
X2Y2 shim gradient (P) .......................................................................665
x3
X3 shim gradient (P) ...........................................................................665
x4
X4 shim gradient (P) ...........................................................................666
Add integral reset point at cursor position (C) .........................671
z0  Z0 field position (P).............................................................672
z1  Z1 shim gradient (P)..........................................................672
z1c Z1C shim gradient (P).......................................................672
z2  Z2 shim gradient (P)..........................................................672
z2c Z2C shim gradient (P).......................................................673
z2x2y2 Z2X2Y2 shim gradient (P)...........................................673
z2x3 Z2X3 shim gradient (P)....................................................673
z2xy Z2XY shim gradient (P)...................................................673
z2y3 Z2Y3 shim gradient (P)....................................................673
z3  Z3 shim gradient (P)..........................................................673
z3c Z3C shim gradient (P).......................................................673
z3x Z3X shim gradient (P).......................................................673
z3x2y2 Z3X2Y2 shim gradient (P)...........................................674
z3x3 Z3X3 shim gradient (P)....................................................674
z3xy Z3XY shim gradient (P)...................................................674
z3y Z3Y shim gradient (P)........................................................674
z3y3 Z3Y3 shim gradient (P)....................................................674
z4  Z4 shim gradient (P)..........................................................674
z4c Z4C shim gradient (P).......................................................674
z4x Z4X shim gradient (P).......................................................674
z4x2y2 Z4X2Y2 shim gradient (P)...........................................675
z4xy Z4XY shim gradient (P)...................................................675
z4y Z4Y shim gradient (P)........................................................675
z5  Z5 shim gradient (P)..........................................................675
z5x Z5X shim gradient (P).......................................................675
z5y Z5Y shim gradient (P)........................................................675
z6  Z6 shim gradient (P)..........................................................675
z7  Z7 shim gradient (P)..........................................................675
z8  Z8 shim gradient (P)..........................................................676
zap Set up for gradient refocused high-speed imaging sequences (M).676
zeroneg Set all negative intensities of 2D spectra to zero (C).......676
zoom Adjust display to given width (M).................................676
zx2y2 ZX2Y2 shim gradient (P)..............................................676
zx3 ZX3 shim gradient (P).......................................................676
zxy ZXY shim gradient (P).......................................................676
zy3 ZY3 shim gradient (P).......................................................677

Index ........................................................................................................679
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)`. 
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 Varian, Inc. 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):

VnmrJ Walkup NMR
VnmrJ Liquids NMR
VnmrJ Installation and Administration
VnmrJ Imaging NMR
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aa</td>
<td>Abort acquisition with error (C)</td>
</tr>
<tr>
<td>abort</td>
<td>Terminate action of calling macro and all higher macros (C)</td>
</tr>
<tr>
<td>abortallacqs</td>
<td>Reset acquisition computer in a drastic situation (C)</td>
</tr>
<tr>
<td>abortoff</td>
<td>Terminate normal functioning of abort in a macro (C)</td>
</tr>
<tr>
<td>aborton</td>
<td>Restore normal functioning of abort in a macro (C)</td>
</tr>
<tr>
<td>abs</td>
<td>Find absolute value of a number (C)</td>
</tr>
<tr>
<td>AC1S-AC11S</td>
<td>Autocalibration macros (M)</td>
</tr>
<tr>
<td>ACbackup</td>
<td>Make backup copy of current probe file (M)</td>
</tr>
<tr>
<td>ACreport</td>
<td>Print copy of probe file after autocalibration (M)</td>
</tr>
<tr>
<td>acos</td>
<td>Find arc cosine of number (C)</td>
</tr>
<tr>
<td>acosy</td>
<td>Automatic analysis of COSY data (C)</td>
</tr>
<tr>
<td>acosyold</td>
<td>Automatic analysis of COSY data, old algorithm (C)</td>
</tr>
<tr>
<td>acqdisp</td>
<td>Display message on the acquisition status line (C)</td>
</tr>
<tr>
<td>acqi</td>
<td>Interactive acquisition display process (C)</td>
</tr>
<tr>
<td>acqmeter</td>
<td>Open Acqmeter window (M)</td>
</tr>
<tr>
<td>Acqmeter</td>
<td>Open Acqmeter window (U)</td>
</tr>
<tr>
<td>acqstat</td>
<td>Open Acquisition Status window (M)</td>
</tr>
<tr>
<td>Acqstat</td>
<td>Open Acquisition Status window (U)</td>
</tr>
<tr>
<td>acqstatus</td>
<td>Acquisition status (P)</td>
</tr>
<tr>
<td>acquire</td>
<td>Acquire data (M)</td>
</tr>
<tr>
<td>add</td>
<td>Add current FID to add/subtract experiment (C)</td>
</tr>
<tr>
<td>addAstack</td>
<td>Add stack</td>
</tr>
<tr>
<td>addfids</td>
<td>Add a series of FIDs together (M)</td>
</tr>
<tr>
<td>addi</td>
<td>Start interactive add/subtract mode (C)</td>
</tr>
<tr>
<td>addnucleus</td>
<td>Add new nucleus to existing probe file (M)</td>
</tr>
<tr>
<td>addpar</td>
<td>Add selected parameters to current experiment (M)</td>
</tr>
<tr>
<td>addparams</td>
<td>Add parameter to current probe file (M)</td>
</tr>
<tr>
<td>addprobe</td>
<td>Create new probe directory and probe file (M)</td>
</tr>
<tr>
<td>addrvcvs</td>
<td>Combine data from multiple receivers (M)</td>
</tr>
<tr>
<td>adept</td>
<td>Automatic DEPT analysis and spectrum editing (C)</td>
</tr>
<tr>
<td>aexppl</td>
<td>Automatic plot of spectral expansion (M)</td>
</tr>
<tr>
<td>ai</td>
<td>Select absolute-intensity mode (C)</td>
</tr>
<tr>
<td>aig</td>
<td>Absolute-intensity group (P)</td>
</tr>
<tr>
<td>aipAnnotation</td>
<td>Annotation template name (P)</td>
</tr>
<tr>
<td>aipAutoLayout</td>
<td>Turn automatic layout on or off (P)</td>
</tr>
<tr>
<td>aipBigFrame</td>
<td>Toggle full-screen mode (C)</td>
</tr>
<tr>
<td>aipClearFrames</td>
<td>Erase all images in displayed frames (C)</td>
</tr>
<tr>
<td>aipClickedFrame</td>
<td>ID of clicked frame (P)</td>
</tr>
<tr>
<td>aipCurrentKey</td>
<td>Image key of currently drawing frame (P)</td>
</tr>
<tr>
<td>aipDeleteData</td>
<td>Unload data (C)</td>
</tr>
<tr>
<td>aipDeleteFrames</td>
<td>Clear the graphics screen (C)</td>
</tr>
<tr>
<td>aipDeleteRois</td>
<td>Delete selected ROIs (C)</td>
</tr>
<tr>
<td>aipDisplay</td>
<td>Display specified images (C)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>aipDisplayByKey</td>
<td>Display a loaded image in a given frame (C)</td>
</tr>
<tr>
<td>AipDisplayMode</td>
<td>Selection mode of image display (P)</td>
</tr>
<tr>
<td>aipDupFrame</td>
<td>Move an image to another frame (C)</td>
</tr>
<tr>
<td>aipExtract</td>
<td>Extract slices from a 3D data set (C)</td>
</tr>
<tr>
<td>aipExtractMip</td>
<td>Extract MIP from a 3D data set (C)</td>
</tr>
<tr>
<td>aipGetSelectedFrames</td>
<td>Get the location and size of selected frames (C)</td>
</tr>
<tr>
<td>aipFlip</td>
<td>Reflect selected images (C)</td>
</tr>
<tr>
<td>aipGetDataKey</td>
<td>Get the key of a loaded image (C)</td>
</tr>
<tr>
<td>aipGetFrame</td>
<td>Get frame index (C)</td>
</tr>
<tr>
<td>aipGetFrameToStart</td>
<td>Get a frame to start image display (C)</td>
</tr>
<tr>
<td>aipGetHeaderParam</td>
<td>Get parameters from FDF header (C)</td>
</tr>
<tr>
<td>aipGetImgKey</td>
<td>Get image keys (C)</td>
</tr>
<tr>
<td>aipLoadDir</td>
<td>Load image data (C)</td>
</tr>
<tr>
<td>aipLoadFile</td>
<td>Load image data (C)</td>
</tr>
<tr>
<td>aipLoadRois</td>
<td>Load ROIs from a file to selected frames (C)</td>
</tr>
<tr>
<td>aipMathExecute</td>
<td>Execute an Image Math Expression (C)</td>
</tr>
<tr>
<td>AipMovieMode</td>
<td>Selection mode of movie (P)</td>
</tr>
<tr>
<td>aipMovieSettings</td>
<td>Size of movie (P)</td>
</tr>
<tr>
<td>aipNumOfCopies</td>
<td>Get number of times an image is loaded (C)</td>
</tr>
<tr>
<td>aipNumOfImgs</td>
<td>Get number of loaded images (C)</td>
</tr>
<tr>
<td>aipRedisplay</td>
<td>Refresh image display (C)</td>
</tr>
<tr>
<td>aipRotate</td>
<td>Rotate selected images (C)</td>
</tr>
<tr>
<td>aipRQtest</td>
<td>Print image keys for debugging (C)</td>
</tr>
<tr>
<td>aipSaveHeaders</td>
<td>Save the auxiliary header files (C)</td>
</tr>
<tr>
<td>aipSaveRois</td>
<td>Save selected ROIs to a file (C)</td>
</tr>
<tr>
<td>aipSaveVs</td>
<td>Save intensity scaling (C)</td>
</tr>
<tr>
<td>aipScreen</td>
<td>Query whether aip owns the graphic area (C)</td>
</tr>
<tr>
<td>aipSegment</td>
<td>Segment images (C)</td>
</tr>
<tr>
<td>aipSelectFrames</td>
<td>Select or deselect image frames (C)</td>
</tr>
<tr>
<td>aipSelectRois</td>
<td>Select or deselect ROIs (C)</td>
</tr>
<tr>
<td>aipSetDebug</td>
<td>Enable debugging messages (C)</td>
</tr>
<tr>
<td>aipSetExpression</td>
<td>Set the image math expression template (C)</td>
</tr>
<tr>
<td>aipSetState</td>
<td>Set AIP mouse state (C)</td>
</tr>
<tr>
<td>aipSetVsFunction</td>
<td>Modify intensity scaling (C)</td>
</tr>
<tr>
<td>aipShow</td>
<td>Load and display images of a given directory (M)</td>
</tr>
<tr>
<td>aipSomeInfoUpdate</td>
<td>Update Point Info and Line Profile pages (C)</td>
</tr>
<tr>
<td>aipSplitWindow</td>
<td>Split the graphics display area into frames (C)</td>
</tr>
<tr>
<td>aipStatPrint</td>
<td>Write ROI statistics to disk (C)</td>
</tr>
<tr>
<td>aipStatUpdate</td>
<td>Update the Statistics page (C)</td>
</tr>
<tr>
<td>aipWriteData</td>
<td>Save image data (C)</td>
</tr>
<tr>
<td>aipUpdateRQlist</td>
<td>Update or rebuild the Review Queue list (C)</td>
</tr>
<tr>
<td>alfa</td>
<td>Set alfa delay before acquisition (P)</td>
</tr>
<tr>
<td>alock</td>
<td>Automatic lock control (P)</td>
</tr>
<tr>
<td>alternateSlices</td>
<td>Alternate slices (C)</td>
</tr>
<tr>
<td>ampmode</td>
<td>Independent control of amplifier mode (P)</td>
</tr>
<tr>
<td>amptype</td>
<td>Amplifier type (P)</td>
</tr>
<tr>
<td>analyz</td>
<td>Calculate standard peak height (M)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>analyze</td>
<td>Generalized curve fitting (C)</td>
</tr>
<tr>
<td>ap</td>
<td>Print out “all” parameters (C)</td>
</tr>
<tr>
<td>ap</td>
<td>“All” parameters display control (P)</td>
</tr>
<tr>
<td>apa</td>
<td>Plot parameters automatically (M)</td>
</tr>
<tr>
<td>aph</td>
<td>Automatic phase adjustment of spectra (C)</td>
</tr>
<tr>
<td>aph0</td>
<td>Automatic phase of zero-order term (C)</td>
</tr>
<tr>
<td>aphb</td>
<td>Auto phasing for Bruker data (C)</td>
</tr>
<tr>
<td>aphx</td>
<td>Perform optimized automatic phasing (M)</td>
</tr>
<tr>
<td>appmode</td>
<td>Application mode (P)</td>
</tr>
<tr>
<td>aptype</td>
<td>Application type (P)</td>
</tr>
<tr>
<td>apt</td>
<td>Set up parameters for APT pulse sequence (M)</td>
</tr>
<tr>
<td>Apt</td>
<td>Set up parameters for APT experiment (M)</td>
</tr>
<tr>
<td>APT</td>
<td>Change parameters for APT experiment (M)</td>
</tr>
<tr>
<td>aptaph</td>
<td>Automatic processing for APT spectra (M)</td>
</tr>
<tr>
<td>arccos</td>
<td>Calculate arc cosine of real number (M)</td>
</tr>
<tr>
<td>arcsin</td>
<td>Calculate arc sine of real number (M)</td>
</tr>
<tr>
<td>arctan</td>
<td>Calculate arc tangent of real number (M)</td>
</tr>
<tr>
<td>array</td>
<td>Easy entry of linearly spaced array values (M)</td>
</tr>
<tr>
<td>arraydim</td>
<td>Dimension of experiment (P)</td>
</tr>
<tr>
<td>asin</td>
<td>Find arc sine of number (C)</td>
</tr>
<tr>
<td>asize</td>
<td>Make plot resolution along f1 and f2 the same (M)</td>
</tr>
<tr>
<td>assign</td>
<td>Assign transitions to experimental lines (M)</td>
</tr>
<tr>
<td>at</td>
<td>Acquisition time (P)</td>
</tr>
<tr>
<td>atan</td>
<td>Find arc tangent of a number (C)</td>
</tr>
<tr>
<td>atan2</td>
<td>Find arc tangent of two numbers (C)</td>
</tr>
<tr>
<td>atcmd</td>
<td>Call a macro at a specified time (M)</td>
</tr>
<tr>
<td>atext</td>
<td>Append string to current experiment text file (M)</td>
</tr>
<tr>
<td>attval</td>
<td>Calculate pulse width (M)</td>
</tr>
<tr>
<td>au</td>
<td>Submit experiment to acquisition and process data (M)</td>
</tr>
<tr>
<td>AuCALch3i</td>
<td>Set up autocalibration with CH3I sample (M)</td>
</tr>
<tr>
<td>AuCALch3i1</td>
<td>Get autocalibration with CH3I sample (M)</td>
</tr>
<tr>
<td>AuCALch3oh</td>
<td>Set up autocalibration with Autotest sample (M)</td>
</tr>
<tr>
<td>AuCALch3oh1</td>
<td>Get autocalibration with Autotest sample (M)</td>
</tr>
<tr>
<td>AuCalibz0</td>
<td>Automatic Hz to DAC calibration for Z0 (M)</td>
</tr>
<tr>
<td>AuCdec</td>
<td>Carbon decoupler calibration macro (M)</td>
</tr>
<tr>
<td>AuCgrad</td>
<td>Carbon/proton gradient ratio calibration macro (M)</td>
</tr>
<tr>
<td>AuCobs</td>
<td>Carbon observe calibration macro (M)</td>
</tr>
<tr>
<td>audiofilter</td>
<td>Audio filter board type (P)</td>
</tr>
<tr>
<td>Aufindz0</td>
<td>Automatic adjustment of Z0 (M)</td>
</tr>
<tr>
<td>Augcal</td>
<td>Probe gcal calibration macro (M)</td>
</tr>
<tr>
<td>Augmap</td>
<td>Automated gradient map generation (M)</td>
</tr>
<tr>
<td>Augmapz0</td>
<td>Automatic lock gradient map generation and z0 calibration (M)</td>
</tr>
<tr>
<td>AuHdec</td>
<td>Proton decoupler calibration (M)</td>
</tr>
<tr>
<td>AuHobs</td>
<td>Proton observe calibration macro (M)</td>
</tr>
<tr>
<td>Aumakegmap</td>
<td>Auto lock gradient map generation (M)</td>
</tr>
<tr>
<td>AuNuc</td>
<td>Get parameters for a given nucleus (M)</td>
</tr>
</tbody>
</table>
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:  VnmrJ Liquids NMR

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)
          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)

**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: *VnmrJ Liquids NMR*

**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:  
\texttt{abs(-25)}  
\texttt{abs(n):abs\_val}

See also: \textit{VnmrJ User Programming}

\textbf{AC1S-AC11S} \hspace{1cm} \textbf{Autocalibration macros (M)}

Syntax: \texttt{ACnS}, where \texttt{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 \texttt{AC1S} determines $^1\text{H}$ pw90, \texttt{AC5S} begins $^{13}\text{C}$ calibration, including decoupler power calibrations. \texttt{AC10S} performs $^{19}\text{F}$ calibration, and \texttt{AC11S} performs $^{31}\text{P}$ calibration.

See also: \textit{VnmrJ Liquids NMR}

\textbf{ACbackup} \hspace{1cm} \textbf{Make backup copy of current probe file (M)}

Syntax: \texttt{ACbackup}

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

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{AC1S-AC11S} \hspace{1cm} Autocalibration macros (M)

\textbf{ACreport} \hspace{1cm} \textbf{Print copy of probe file after autocalibration (M)}

Syntax: \texttt{ACreport}

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

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{AC1S-AC11S} \hspace{1cm} Autocalibration macros (M)

\textbf{acos} \hspace{1cm} \textbf{Find arc cosine of number (C)}

Syntax: \texttt{acos(value)<:n>}

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

Arguments: \texttt{value} is a number in the range of $\pm 1.0$ to +1.0.

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

Examples: \texttt{acos(.5)}  
\texttt{acos(value):acos\_val}

See also: \textit{VnmrJ User Programming}

Related: \texttt{sin} \hspace{1cm} Find sine value of an angle (C)

\textbf{acosy} \hspace{1cm} \textbf{Automatic analysis of COSY data (C)}

Syntax: \texttt{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: VnmrJ Liquids NMR

Related: `acosyold` Automatic analysis of COSY data, old algorithm (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.

Related: `acosy` Automatic analysis of COSY data (C)

`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: VnmrJ Liquids NMR

`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 UNITY/NOVA systems only, you can use the Acquisition window to shim on the sample while an acquisition is in progress. This feature is not available on other systems. 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 VnmrJ Liquids NMR 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, ve, 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: VnmrJ Liquids NMR

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)
f 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)
spn Sample spin rate (P)
w 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)>

42 VnmrJ 1.1D Command and Parameter Reference 01-999252-00 A0604
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('inova500')
```

See also: `VnmrJ Liquids NMR`

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 inova400 &
Acqmeter gem300 -f inova500.lisa &
```
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 *VnmrJ Liquids NMR*.

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: *VnmrJ Liquids NMR*

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

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 *VnmrJ Liquids NMR*.

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 inova400 &  
Acqstat gem300 -f inova500.lisa &  

See also:  VnmrJ Liquids NMR  
Related:  Acqstat  Open the Acquisition Status window (U)  
showstat  Display information about status of acquisition (C,U)  

acqstatus  

Acquisition status (P)  

Applicability:  All systems, except codes marked with an asterisk (*) are not used on MERCURYplus/Vx systems.  

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 vtt ime 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 \texttt{pw} (AGAINFAIL)

Hard errors
901. Incorrect PSG version for acquisition
See also: *VnmrJ Liquids NMR*

**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.

**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:**

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

**See also:** *VnmrJ Liquids NMR*
**addAstack**  
**Add stack**

**Applicability:** Systems with imaging capabilities.

**Syntax:** `addAstack`

**Description:** Adds a stack of the given type. If `type` is not given, `type=0`; if `type =-1`, the default type will be used.

**See also:** *VnmrJ Liquids NMR*

**Related:** `gplan`  
Start interactive image planning (C)

**addfids**  
**Add a series of FIDs together (M)**

**Applicability:** Systems with LC-NMR accessory.

**Syntax:** `addfids<(start,finish)>`

**Description:** Improves signal-to-noise by adding adjacent FIDs that represent the same peak.

Given a series of FIDs that represent separate data, such as occur during an LC-NMR run, some of the adjacent FIDs can actually represent the same peak in the LC run.

To obtain the FID numbers to use, you can enter `dss` or `dsww` (e.g., enter `dsww(25,35)` and then determine that peak numbers 28 to 31 contain the peaks of interest), or you can enter `dconi` and then read the Index counter on line 1 of the display.

**Arguments:**
- `start` is the number of the first FID to be co-added. The default is that you are prompted for the value.
- `finish` is the number of the last FID to be co-added. The default is that you are prompted for the value.

**Examples:**
- `addfids`
- `addfids(25,28)`

**See also:** *VnmrJ Liquids NMR*

**addi**  
**Start interactive add/subtract mode (C)**

**Syntax:** `addi`

**Description:** Starts the interactive add/subtract mode. Before entering `addi`, start the process with `ciradd` 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 \texttt{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 \texttt{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 \texttt{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 (\texttt{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 \texttt{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: \textit{VnmrJ Liquids NMR}

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

\textbf{addnucleus} \textit{Add new nucleus to existing probe file (M)}

\textbf{Syntax}: \texttt{addnucleus<(nucleus)>}

\textbf{Description}: Appends entries for nuclei not in the default probe file to the end of the file.

\textbf{Arguments}: If no argument is entered, a prompt is displayed requesting the nucleus entry.

\textit{nucleus} is a nucleus entry in the \texttt{nuctable}.

\textbf{Examples}: \texttt{addnucleus}
\texttt{addnucleus('Si29')}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{addprobe} Create new probe directory and probe file (M)
\texttt{getparam} Receive parameter from probe file (M)
\texttt{probe} Probe type (P)
\texttt{setparams} Write parameter to current probe file (M)

\textbf{addpar} \textit{Add selected parameters to current experiment (M)}

\textbf{Syntax}: \texttt{addpar<('2d'|'3d'|'3rf'|'4d'|'downsamp'|'fid'|'image'|'ll2d'|'lp',dim>|'oversamp'|'ss'>}

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

\textbf{Description}: Creates selected parameters in the current experiment.

\textbf{Arguments}: If no argument is entered, \texttt{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 be 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, deltax, dotflag, vpf, and vpfj 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 ssrlsfrq 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: VnmrJ Liquids NMR; 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)
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: VnmrJ Liquids NMR

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 1H, 19F, 13C, and 15N. The information is saved in the user’s directory vnmrjsys/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 /vnmrj/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: VnmrJ Liquids NMR; VnmrJ Walkup NMR

Related:
- addnucleus  Add new nucleus to existing probe file (M)
- getparam  Receive parameter from probe file (M)
addrcvrs Combine data from multiple receivers (M)

Applicability: Imaging systems with multiple receivers.

Syntax: addrcvrs

Description: Combines image data that has been acquired by multiple receivers. First transforms the data from each receiver separately with 'wft2d'. Weights the individual images by the factors specified in the 'rcvrwt' parameter and forms the RMS average.

Related: rcvrwt Weighting for different receivers (M)
wft2d Weight and Fourier Transform 2D data (C)
rmsAddData Add transformed data files with weighting (U)

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 CH₃ carbons only
- #3 is CH₂ 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
ddept('coef')
ddept('theory','noll')

See also: VnmrJ Liquids NMR

Related: autodept Automated complete analysis of DEPT data (M)
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:

```c
aexppl
aexppl(20)
```

See also: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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)

### aipAnnotation

Annotation template name (P)

**Description:** Name of annotation template.

**Values:**

- `'short'`
- `'full'`
- `'none'`

See also: *VnmrJ Imaging NMR: Image Processing*

### aipAutoLayout

Turn automatic layout on or off (P)

**Description:** Integer parameter to turn on/off automatic layout.

**Values:**

- `1`, for auto layout
- `0`, for no auto layout

### aipBigFrame

Toggle full-screen mode (C)

**Syntax:** `aipBigFrame`

**Description:** Toggle the display between multiple-image display and full-screen display of one image. If the current display is multiple-image, and at least one frame is
selected, the first selected frame is expanded to fill the display area. If no frame is selected, the command has no effect. If the current display is full-screen, the previous multiple-image display is restored. After the command, all frames are unselected.

See also: VnmrJ Imaging User Guide: Image Processing

Related: aipSplitWindow Split the graphics display into frames (C)

**aipClearFrames** Erase all images in displayed frames (C)

Syntax: aipClearFrames

Description: Clears all the currently displayed frames of images. The image data remains loaded. If the current display is full-screen mode, it is toggled to multiple-image mode first.

See also: VnmrJ Imaging User Guide: Image Processing

Related: aipBigFrame Toggle full-screen mode (C)
aipDeleteData Unload data (C)
aipDeleteFrames Clear the graphics screen

**aipClickedFrame** ID of clicked frame (P)

Description: ID (an integer) of the clicked frame.

Values: 1, 2, 3

**aipCurrentKey** Image key of currently drawing frame (P)

Description: Image key of the currently drawing frame. This key is used to get parameter values for annotation drawing.

Values: directory + space + filename + space + n

**aipDeleteData** Unload data (C)

Syntax: (1) aipDeleteData
(2) aipDeleteData('sel')
(3) aipDeleteData(key)

Description: Unloads all data or selected data. Does not delete the data files.

Using syntax 1, all loaded data is unloaded, and the screen is cleared (no frames displayed).

Using syntax 2, the data displayed in all the selected frames is unloaded, and the frames are displayed empty.

Arguments: key is a string data key or a file containing a list of keys. It must begin with a "/".

Examples: aipDeleteData('/usr/vnmr1/vnmrsys/data/keylist')


Related: aipClearFrames Erase all images in displayed frames (C)

**aipDeleteFrames** Clear the graphics screen (C)

Syntax: aipDeleteFrames
**Description:** Deletes all frames and displays a blank graphics screen. If the current display is full-screen mode, it is toggled to multiple-image mode first.

**See also:** *VnmrJ Imaging User Guide: Image Processing*

**Related:**
- `aipClearFrames` Erase all images in displayed frames (C)
- `aipDeleteData` Unload data (C)
- `aipSplitWindow` Split the graphics display into frames (C)

### aipDeleteRois Delete selected ROIs (C)

**Syntax:** `aipDeleteRois`

**Description:** Deletes all selected ROIs.

**See also:** *VnmrJ Imaging User Guide: Image Processing*

**Related:**
- `aipSelectRois` Select or deselect ROIs (C)
- `aipLoadRois`
- `aipSaveRois`

### aipDisplay Display specified images (C)

**Syntax:**
1. `aipDisplay`
2. `aipDisplay(['reset',] 'redisplay')`
3. `aipDisplay(['reset',] 'all')`
4. `aipDisplay(['reset',] 'batch' [,,'show'] [,,'next' | 'previous' | 'first' | 'last'])`

**Description:** For any syntax, the optional 'reset' argument forces VnmrJ to recalculate all the display buffers for the images, rather than using cached values.

Using syntax 1, as many images are displayed as fit in the current frames. The first image is displayed in the first selected frame, and successive images are displayed in successive frames until either the frames are all full or there are no more images. The display will wrap around from the last to the first frame if appropriate.

Using syntax 2, the current display is refreshed.

Using syntax 3, the screen is split into enough frames to hold all the loaded images, and they are all displayed simultaneously.

Using syntax 4, a "batch" of images is displayed in the existing frames according to the values in the parameter `aipDisplay[1:3]`. The 'show' option must be present for anything to actually be displayed.

The 'first' option initializes the display to the first batch of images.

The 'last' option initializes the display to the last batch of images.

The 'next' option updates `aipDisplay[1]` (the number of the first image to show) to show the next batch before the images are displayed. This is done after any first/last operation.

The 'previous' option updates `aipDisplay[1]` to show the previous batch before the images are displayed. This is done after any first/last operation.

The options can be used in any order. Normally, use the 'show' option alone to do a redisplay function, or one of the first/last/next/previous options plus the 'show' option.

**Examples:**
- `aipDisplay('batch','next','show')` Displays the next "batch" of images.
- `aipDisplay[2]=2` `aipDisplay('batch','show')` Sets the number of images to display to 2, and displays them in the current frames.
aipDisplayByKey
Display a loaded image in a given frame (C)
Syntax: aipDisplayByKey($key,$frame)
Description: Display the image defined by $key in $frame.
Applicability: $key

AipDisplayMode
Selection mode of image display (P)
Description: Integer parameter to hold selection mode of image display.
Values: 1, all loaded images
2, images in a group
3, images selected in the Review Queue
4, images in selected frames
5, images selected by user using:
vnmrjcmd('RQ Rqupdate',aipRoiBind,rgsort,userselection)

aipDupFrame
Move an image to another frame (C)
Syntax: aipDupFrame(srcFrame, dstFrame)
Description: Moves an image from one frame to another. (The Dup is really a misnomer.)
The n frames are numbered from 1 to n, from left to right and top to bottom. If there is no image in the source frame, the effect is to clear the destination frame. If either argument is outside the range 1 <= arg <= n, the command does nothing.
Arguments: srcFrame is the number of the frame containing the source image.
dstFrame is the number of the frame in which to put the image.
Examples: aipDupFrame(1, 3)
See also: VnmrJ Imaging User Guide: Image Processing

aipExtract
Extract slices from a 3D data set (C)
Syntax: aipExtract(['xy'|'yz'|'xz'], first [, last [, incr]])
Description: When a 3D data set is loaded, it is not displayed, but just saved in memory. Only one 3D data set can be loaded at a time. The aipExtract command extracts slices from the current 3D data set that are then displayed.
Arguments: xy, yz, xz are the three possible plane orientations to extract. The X dimension is the fastest data direction and Z is the slowest.
first is the number of the first slice to extract, counting from 1. The slice order is always from the start of the data set.
last is the maximum slice number to extract. If absent, only 1 slice is extracted.
incr is the increment between extracted slice numbers. If absent, it defaults to 1.
Examples: aipExtract('xy', 10, 22)
See also: VnmrJ Imaging User Guide: Image Processing
Related: aipExtractMip Extract MIP from a 3D data set (C)
**aipExtractMip** Extract MIP from a 3D data set (C)

Syntax:  
aipExtractMip(['xy'|'yz'|'xz'], first [, last [, incr]])

Description:  Like aipExtract, but instead of extracting a set of slices, constructs one slice in which each pixel contains the maximum value for that pixel in any of the specified slices.

Arguments:  xy, yz, xz are the three possible plane orientations to extract. The X dimension is the fastest data direction and Z is the slowest.

first is the number of the first slice to extract, counting from 1. The slice order is always from the start of the data set.

last is the maximum slice number to extract. If absent, only 1 slice is extracted.

incr is the increment between extracted slice numbers. If absent, it defaults to 1.

Examples:  aipExtractMip('xy', 10, 22)

See also: *VnmrJ Imaging User Guide: Image Processing*

Related:  aipExtract

**aipGetSelectedFrames** Get the location and size of selected frames (C)

Syntax:  
aipGetSelectedFrames:$str

Description:  Return a string that contains 1+n*4 integer numbers. First integer is the number of selected frames, next 4 numbers are the location (x, y) and FOV (width, height) of the first frame, and so on.

**aipFlip** Reflect selected images (C)

Syntax:  
aipFlip('0' | '90' | '45' | '135')

Description:  Reflects all selected images about one of 4 axes. Axes are defined relative to the screen view. This reflection can be reset by displaying the image with aipDisplay('reset', ...).

Arguments:  
'0' reflects about the Y axis.

'90' reflects about the X axis.

'45' reflects about the line X = -Y.

'135' reflects about the line X = Y.

Examples:  aipFlip('0')

See also: *VnmrJ Imaging User Guide: Image Processing*

Related:  aipRotate  Rotate selected images (C)

**aipGetDataKey** Get the key of a loaded image (C)

Syntax:  
aipGetDataKey:$key

aipGetDataKey(x,y):$key

Description:  Return the key of a loaded image. If mouse position (x, y) is not explicitly given, the key of the image last clicked is returned. If (x, y) is not on an image, or no image has been clicked, an empty string is returned.

Arguments:  x, y, a (mouse) position to determine the frame.
**aipGetFrame**  
Get frame index (C)  

Syntax:     aipGetFrame(x, y):$frame  

Description: Return frame index for mouse position (x, y) Used by Review Queue to drop an image, scan, or study to a frame.  

Arguments: x, y, mouse position.  

**aipGetFrameToStart**  
Get a frame to start image display (C)  

Syntax:     aipGetFrameToStart:$frame  

Description: Return a frame to start image display. If a frame is selected, return that frame, otherwise if there is empty frame(s), return the first empty frame; otherwise, return the first frame.  

**aipGetHeaderParam**  
Get parameters from FDF header (C)  

Syntax:     aipGetHeaderParam(key,name,[index]):$value,$type  

Description: Return value and type of a fdf header parameter. If a parameter does not exist, $value and $type are empty. This command is used by annotation in macro annPar, where the key of the image being drawing is aipCurrentKey. In other cases, use aipGetDataKey to get the key of a displayed image.  

Arguments: key is the key of a loaded image, i.e., directory + space + filename + n, where n is a number distinguishing different copies of the same image. By default, n is zero when the image is loaded the first time. It increases by one when the same image is loaded next time.  

index is the index of an array parameter, can be omitted if the parameter is not arrayed.  

See also: aipCurrentKey (P), aipGetDataKey (C).  

**aipGetImgKey**  
Get image keys (C)  

Syntax:     aipGetImgKey(mode):$n  
  aipGetImgKey(mode, image_index):$key  

Description: This command returns the number of images for a given selection mode, or the key of an image in the selected list.  

Arguments: mode is an integer for different selection modes:  
1, all loaded images  
2, images of a group (scan)  
3, images selected in the Review Queue  
4, images in selected frames  
5, images selected by user using:  
  vnmrjcmd('RQ RQupdate',aipRoiBind,rqsort,userselection)  
5, currently displayed images  
image_index is the index of an image in the selected list.  

**aipLoadDir**  
Load image data (C)  

Description: Same as aipLoadFile
**aipLoadFile**  Load image data (C)

Syntax: (1) `aipLoadFile(filepath [, frame])`
(2) `aipLoadFile(dirpath)`

Description: Loads a single file or a directory full of files.

Arguments: 
- `filepath` is the full path to an FDF file.
- `dirpath` is the full path to a directory with FDF files.
- `frame` is the index of the frame in which to display the image, starting from 0.

Examples: `aipLoadFile('/vnmr/fidlib/monkey.dat/')`

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `aipDisplay` Display selected images (C)

**aipLoadRois**  Load ROIs from a file to selected frames (C)

Syntax: `aipLoadRois(fullpath)`

Description: Load ROIs from a file to selected frames. If an ROI is loaded to multiple frames, it is bound.

Arguments: full path

**aipMathExecute**  Execute an Image Math Expression (C)

Syntax: `aipMathExecute('gstring' [, 'parm'])`

Description: Executes the expression string contained in the given parameter. If the `parm` argument is present, the string `pnew 1 parm` is sent to VnmrJ after the math expression is executed.

Arguments: 
- `gstring` is a global string parameter that contains a legal image math expression.
- `parm` is any parameter name.

Examples: `aip2CExp='#8=#1+#7' aipMathExecute('aip2CExp')`

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `aipSetExpression` Set the image math expression template (C)

**AipMovieMode**  Selection mode of movie (P)

Description: Integer parameter to hold selection mode of movie run.

Values: 
- 1, all loaded images
- 2, images in a group
- 3, images selected in the Review Queue
- 4, images in selected frames.
- 5, images selected by user using: `vnmrjcmd('RQ RQupdate',aipRoiBind,rqsort,userselection)`
- 6, currently displayed images

**aipMovieSettings**  Size of movie (P)

Description: Arrayed parameters of size 3, for image movie settings.

Values: 
- `aipMovieSettings[1]`, 1/0, repeat/not repeat the movie.
- `aipMovieSettings[2]`, 1/0, show/not show graphics on movie.

Examples: `aipMovieSetting=0,1,0`
**aipNumOfCopies** Get number of times an image is loaded (C)

Syntax: `aipNumOfCopies(fdfPath):$n`  
Description: Return number of times an FDF image file is loaded.  
Arguments: full FDF path

**aipNumOfImgs** Get number of loaded images (C)

Syntax: `aipNumOfImgs:$n`  
Description: Return a number that represents the total number of loaded images. If the same FDF file is loaded n times, it is counted as n images.

**aipRedisplay** Refresh image display (C)

Syntax: `aipRedisplay`  
Description: Refreshes the image display area, allowing for change in the window size. If the parameter `aipFrameResplitOnResize` is non-zero, the frames are laid out anew to fit a new window shape. (This feature is buggy, in that the wrong images and the wrong number of images might be displayed.)

See also: VnmrJ Imaging User Guide: Image Processing

Related: `aipDisplay` Display selected images (C)

**aipRotate** Rotate selected images (C)

Syntax: `aipRotate('90' | '180' | '270' | '-90')`  
Description: Rotates all selected images counterclockwise by the number of degrees indicated by the argument. Note that the '270' and '-90' arguments are equivalent. This rotation can be reset by displaying the image with `aipDisplay('reset', ...)`.  
Examples: `aipRotate('90')`

See also: VnmrJ Imaging User Guide: Image Processing

Related: `aipFlipp` Reflect selected images (C)  
`aipDisplay` Display selected images (C)

**aipRQtest** Print image keys for debugging (C)

Syntax: `aipRQtest(mode)`  
Description: Print selected image keys as a list (as ordered in the Review Queue) to a window specified by the VnmrJ command `jFunc(55,...)`.  
Arguments:  
`mode = 1`, all loaded images  
`mode = 2`, images of the group  
`mode = 3`, images selected in Review Queue  
`mode = 4`, images of selected frames  
`mode = 5`, images selected by user with the command: `vnmrjcmd('RQ RQupdate',aipRoiBind,rqsort,userselection)`  
currently displayed images

**aipSaveHeaders** Save the auxiliary header files (C)

Syntax: `aipSaveHeaders`
Description: Write auxiliary header files for all loaded images. This contains whatever is in
the auxiliary symbol table for the data, currently only VS (intensity scaling) information.

See also: *VnmrJ Imaging User Guide: Image Processing*
Related: `aipSaveVs` Save intensity scaling (C)

**aipSaveRois**  
**Save selected ROIs to a file (C)**

Syntax: `aipSaveRois(fullpath)`

Description: Save selected ROIs to a file in the format of:

- **Box**
  - `x1 y1`
  - `x2 y2`

- **Oval**
  - `x1 y1`
  - `x2 y2`

- **Polygon**
  - `n`
  - `xi yi`
  - `....`

  where `xi` and `yi` are coordinates of point `i`.

If ROIs are bound or are identical, only one copy is saved.

Arguments: `full path`

**aipSaveVs**  
**Save intensity scaling (C)**

Syntax: `aipSaveVs`

Description: Updates the VS (intensity scaling) information in the auxiliary symbol tables of all loaded images and writes out the auxiliary headers.

See also: *VnmrJ Imaging User Guide: Image Processing*
Related: `aipSaveHeaders` Save the auxiliary header files (C)

**aipScreen**  
**Query whether aip owns the graphic area (C)**

Syntax: `aipScreen:$b`

Description: Return 1 if `aip` owns the screen; otherwise 0.

**aipSegment**  
**Segment images (C)**

Syntax: (1)`aipSegment[('i')]`
(2) `aipSegment('r')`
(3) `aipSegment('R')`

Description: Segment all selected images, i.e., set all the pixels whose values are outside a given range to 0. The range is defined by the two parameters `aipStatCursMin` and `aipStatCursMax`. If either is inactive, there is no limit for the minimum or maximum value, respectively.

Using syntax 1, all selected images are segmented. This is the default, so the argument is optional.

Using syntax 2, all selected ROIs are segmented, with the region outside the ROI being entirely cleared.

Using syntax 3, all selected ROIs are segmented, with the region outside the ROI being unaffected.
See also: *VnmrJ Imaging User Guide: Image Processing*

**aipSelectFrames**  
Select or deselect image frames (C)

Syntax:  
1. `aipSelectFrames('all')`  
2. `aipSelectFrames('none')`

Description:  
Using syntax 1, all the displayed frames are selected. This is the default action if there is no argument.  
Using syntax 2, all displayed frames are deselected.

See also: *VnmrJ Imaging User Guide: Image Processing*

**aipSelectRois**  
Select or deselect ROIs (C)

Syntax:  
1. `aipSelectRois('all')`  
2. `aipSelectRois('none')`

Description:  
Using syntax 1, all ROIs are selected. This is the default action if there is no argument.  
Using syntax 2, all ROIs are deselected.

See also: *VnmrJ Imaging User Guide: Image Processing*

**aipSetDebug**  
Enable debugging messages (C)

Syntax:  
`aipSetDebug([flag,] bit# ... [, flag, bit# ...] ...)`

Description: Enables and disables debugging messages from the VnmrJ AIP modules. The parameters are processed successively to build up the final bit mask. The flag parameter tells what succeeding bit numbers do.

Arguments:  
flag is a string indicating the mode for the following bit# parameters.
Possible values are:
- 'off' -- the following indicated bits are turned off.
- 'on' -- the following indicated bits are turned on. An initial 'on' flag is the default.
- 'none' -- the bit mask is cleared, and the following indicated bits are turned on.

bit# is the bit number to turn off or on. Meanings of some bits are:
- 0 -- Log construction and destruction of major classes
- 1 -- not used
- 2 -- Log construction and destruction of ROIs
- 3 -- Print out the rotation matrices for images
- 4 -- Log calls to aipRedisplay
- 5 -- Print time required to draw images
- 6 -- Check memory management of DDL symbol tables
- 7 -- Track loading and unloading of data files
- 8 -- Log progress of image math evaluations

Examples:  
aipSetDebug('none', 3, 6) Sets bits 3 and 6 only.
aipSetDebug(4) Adds bit 4 to the bits set.
aipSetDebug('off', 3, 'on' 5) Turns bit 3 off and 5 on.

See also: *VnmrJ Imaging User Guide: Image Processing*

**aipSetExpression**  
Set the image math expression template (C)

Syntax:  
aipSetExpression(expr)
Description: Used for initializing the entry box in the image math panel when an expression is selected from the menu. First, the string \#= is prepended to the expr string. Then the parameters aip2CExp and aip2JExp are both set to the resulting expression. Also, the parameters aip2CCaret and aip2JCaret are set to 0. Finally, the command pnew 1 aip2JExp is sent to VnmrJ.

Arguments: expr is the right-hand-side of an image math expression. Normally the frame numbers are not filled in.

Examples: aipSetExpression('#+#)

See also: VnmrJ Imaging User Guide: Math Processing

Related: aipMathExecute Execute image math expression (C)

aipSetState Set AIP mouse state (C)

Syntax: aipSetState(state)

Description:

Arguments: state is the number of the state. Values are:
1 -- select
2 -- vs
3 -- createPoint
4 -- createLine
5 -- createBox
6 -- createPolyline
7 -- createPolygon
8 -- zoom
10 -- createOval
99 -- dragImage
100 -- imageMath

Examples: aipSetState(1)

See also: VnmrJ Imaging User Guide: Image Processing

aipSetVsFunction Modify intensity scaling (C)

Syntax: (1) aipSetVsFunction('hist')
(2) aipSetVsFunction('file')
(3) aipSetVsFunction('cmd', function)

Description: Sets the intensity scaling (the VS) of images. Which images are affected is determined by the value of the aipVsMode parameter.

Using syntax 1, updates the histogram display on the VnmrJ imaging VScale page.

Using syntax 2, modifies the scaling to that specified in the file whose path is given by the aipVsFunctionFile parameter.

Arguments: function is a string that specifies the VS function. Must begin with the word "curve" and be of the following form:
"curve" x y "imin" dark "imax" light "dmin" min "dmax" max where the items in quotes are entered literally, and x and y are the coordinates of the control point in the graph of the VS function, in the range [0, 1]. For a linear function set x = y = 0.5.

dark and light are the relative screen intensities for the lowest and highest data values, respectively. Black is 0 and 1 is as bright as it can get.

min and max are the minimum and maximum data values that are mapped. Data values outside this range are colored the same as if they were at the limit.
Examples: \texttt{aipSetVsFunction('cmd', 'curve .5 .5 imin 0 imax 1 dmin 0 dmax .01')}

See also: \textit{VnmrJ Imaging User Guide: Image Processing}

Related: \texttt{aipVsMode}
\texttt{aipVsFunctionFile}

\textbf{aipShow} \hspace{1cm} \textbf{Load and display images of a given directory (M)}

\textbf{Syntax:} \texttt{aipShow(dir, <framelayout>, <action>)}

\textbf{Description:} This macro loads and displays images in a given path. In the Review viewport, it also adds the data to the Review Queue.

\textbf{Arguments:} \texttt{dir}, full path of image directory \texttt{framelayout}, 'all' to display all images \texttt{n}, to automatically layout \texttt{n} frames default, use current frame layout. \texttt{action}, 'dnd', or 'DragNDrop' to keep currently displayed images default, unload currently displayed images

\textbf{Examples:} \texttt{aipShow(sqdir+’/data/sems_001.img’, 'all', 'dnd')} to append images in the given directory to current display, and show all images.

See also: \texttt{aipDeleteData, aipDisplay, aipSplitWindow}

\textbf{aipSomeInfoUpdate} \hspace{1cm} \textbf{Update Point Info and Line Profile pages (C)}

\textbf{Syntax:} \texttt{aipSomeInfoUpdate}

\textbf{Description:} Updates the point and line information displays. For the line information, this involves updating the file pointed to by the \texttt{aipProfileFile} and all the relevant \texttt{aipProfile...} parameters. For the point information, the \texttt{aipPoint...} parameters are updated.

See also: \textit{VnmrJ Imaging User Guide: Math Processing}

Related: \texttt{aipStatUpdate} \hspace{1cm} Update the statistics page (C)

\textbf{aipSplitWindow} \hspace{1cm} \textbf{Split the graphics display area into frames (C)}

\textbf{Syntax:} (1) \texttt{aipSplitWindow}
(2) \texttt{aipSplitWindow('all')}
(3) \texttt{aipSplitWindow(nframes)}
(4) \texttt{aipSplitWindow(nframes, width, height)}
(5) \texttt{aipSplitWindow(nrows, ncols)}

\textbf{Description:} Using syntax 1, the window is split into enough frames to hold all the loaded images, up to the maximum set by the parameter \texttt{aipFrameDefaultMax}. The ratio of the number of rows to the number of columns is chosen to keep the frames as square as possible.

Using syntax 2, the \texttt{aipFrameDefaultMax} parameter is ignored, and the maximum number of frames is limited only by the minimum frame size, 10x10 pixels. Otherwise, this is the same as syntax 1.

Using syntax 3, the window is split into at least the indicated number of frames.

Using syntax 4, the window is split into at least the indicated number of frames, and the aspect ratio of the frames is kept close to the specified width/height ratio.

Using syntax 5, the window is split into the indicated number of rows and columns, subject only to the limit on the minimum allowable frame size.
Arguments: `nframes` is an integer greater than 0 specifying the minimum number of frames to show. The actual number of frames may be greater, because some numbers will be attainable with a reasonable split of rows and columns.

Examples:
- `aipSplitWindow(10)` Makes 10 or more roughly square frames.
- `aipSplitWindow(10,1,2)` Makes the frames tall and skinny.
- `aipSplitWindow(3,4)` Makes 12 frames in 3 rows and 4 columns.

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `aipDeleteFrames` Clear the graphics screen (C)

---

**aipStatPrint**  
**Write ROI statistics to disk (C)**

**Syntax:** `aipStatPrint(path [, 'w' | 'a'])`

**Description:** Writes the current ROI statistics that are displayed in the Statistics page. The data is written to the specified file in a human readable, tabular format. If the optional 'a' argument is given, the new data is appended to any data already in the file. With the 'w' argument, or no second argument, any previous data in the file is deleted.

**Arguments:** `path` is the full path and file name.

**Examples:**
- `aipStatPrint('/tmp/statistics', 'a')`

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `aipStatUpdate` Update the statistics page (C)

---

**aipStatUpdate**  
**Update the Statistics page (C)**

**Syntax:** `aipStatUpdate`

**Description:** Updates the ROI statistics information page. This involves updating all the relevant `aipStat...` parameters.

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `aipSomeInfoUpdate` Update point info and line profile pages (C)

---

**aipWriteData**  
**Save image data (C)**

**Syntax:** `aipWriteData[(filename)]:path`

**Description:** Saves image data for the selected frames. The data format is that indicated by the `aipWriteFmtConvert` parameter. An index number is appended to the given `filename` in the form `.nnnn`, where the index number starts at 1 for the first selected image. Another suffix is appended after the index to indicate the data type. This will depend on the format, and is specified in the `aipWriteFmtConvert` string. This command may overwrite previously written images. If the file name is not specified, the name in the `aipWritePath` parameter is used. Returns the full path of the last file that was written.

**Arguments:** `filename` is either the full path or just the name of the file where the data is written. If it is not a full path, it is relative to the current experiment directory.

**Examples:**
- `aipWriteData('head'):n1`

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `aipWritePath`
aipUpdateRQlist

Update or rebuild the Review Queue list (C)

Syntax: aipUpdateRQlist('update'/'rebuild')

Description: This command called by Review Queue display commands. It updates or rebuilds Review Queue list (a list of images and their information as maintained in the Review Queue) for the Browser to reflect changes in Review Queue table. Current Review Queue table will be written out to a file, the Browser will read the file and update the Rqlist.

Arguments: a string of value 'update' or 'rebuild'.

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 MERCURYplus/Vx broadband systems, beta is 2.
- 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 UNITY/NOVA 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: VnmrJ Liquids NMR

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)

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: VnmrJ Liquids NMR

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)

**alternateSlices** Alternate slices (C)

**Applicability:** Systems with imaging capabilities.

**Syntax:** alternateSlices(intmode)

**Description:** If `mode=0`, restores the order; `mode=1` alternates slices; `mode=-1` toggles between the two modes.

**Related:** gplan Start interactive image planning (C)

**ampmode** Independent control of amplifier mode (P)

**Applicability:** UNITY/INOVA systems.

**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 CONFIG window (opened from config) using the label Type of Amplifier.

On UNIT/INOVA systems, 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'.

On MERCURYplus/Vx systems, amptype specifies the type of amplifier on each rf channel of the spectrometer. The value is set in the CONFIG window (opened from config) using the label Type of Amplifier.

Values: On UNIT/INOVA Systems:
- '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.

On MERCURYplus/Vx systems:
- 'aa' indicates the system has a linear 4-Nucleus amplifier with two outputs: $^{13}$C/$^{31}$P and $^{19}$F/$^1$H at a nominal 35W each.
- 'bb' indicates the system has a linear broadband amplifier with two outputs: $^{15}$N to $^{31}$P and $^{19}$F/$^1$H at a nominal 125W and 75W respectively.
- 'cc' indicates the system has a linear CP/MAS amplifier with two outputs: $^{15}$N to $^{31}$P and $^{19}$F/$^1$H at a nominal 300W and 100W respectively.

See also: Software Installation and MERCURYplus CP/MAS Installation, Testing, and Operation

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

analyze  | Calculate standard peak height (M)
---|---
Syntax: analyze($option,$title)
Description: Macro to calculate average peak height and std deviation and or average phase and std 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: analyze – 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 VnmrJ Liquids NMR 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.
- 'increment' sets kinetics analysis, with increasing peak height.
- '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')
analyze('expfit','p2','contact_time','list')
analyze('expfit','regression','poly1','list')

See also: VnmrJ Liquids NMR

Related: contact_time MAS cross-polarization spin-lock contact time (M)
expfit Least squares fit to polynomial or exponential curve (U)
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)

ap

Print out "all" parameters (C)

Syntax: ap<(template)>

Description: Prints a parameter list containing “all” parameter names and values.
Arguments:  template is the name of the template. The default is a template controlled by the parameter ap, which can be modified with the command paramvi('ap'). See the manual VnmrJ User Programming for rules on building a template.

Examples:  ap
            ap('newap')

See also:  VnmrJ Liquids NMR, 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:  VnmrJ Liquids NMR, 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:  VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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)

**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: VnmrJ Liquids NMR

Related:
- `aph` Automatic phase adjustment of spectra (C)
- `aph0` Automatic phase of zero-order term only (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`, `sysmenulibpath`, and `syshelpopath`.

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` can be set by entering its value directly from the command line. New applications modes can be added by creating the appropriate subdirectories in `/vnmr/maclib`, `/vnmr/ menus`, and `/vnmr/help`, and adding the desired applications mode name to the `_appmode` macro. Subdirectories should be named by adding the file extension `.appmodename` to the corresponding parent directory name (e.g., `maclib.solids`, `menulib.automation`).

Values: 'standard' sets standard application mode.  
'imaging' sets imaging 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: `execpars(M)`, `execsetup(P)`, `execprep(P)`, `execprescan(P)`, `execprocess(P)`, `execplot(P)`

**apt**  
**Set up parameters for APT pulse sequence (M)**  
Syntax: `apt<(solvent)>`  
Description: Converts a parameter set to the APT (attached proton test) experiment.  
Arguments: `solvent` is the name of the solvent used. The default for `solvent` is CDCl3 or, if in the automation mode, the default is read from the file `sampleinfo`.

See also: *VnmrJ Liquids NMR*

Related: `aptaph` Automatic processing for APT spectra (M)

**Apt**  
**Set up parameters for APT experiment (M)**  
Description: Set up parameters for APT experiment

**APT**  
**Change parameters for APT experiment (M)**  
Syntax: `APT`  
Description: Converts the current parameter set to an APT experiment.

Related: `apt` Set up parameters for APT experiment (M)

**aptaph**  
**Automatic processing for APT spectra (M)**  
Syntax: `aptaph`  
Description: Automatically phases APT spectra.

See also: *VnmrJ Liquids NMR*

Related: `apt` Set up parameters for APT pulse sequence (M)
**arccos**  
**Calculate arc cosine of real number (M)**

Applicability: Systems with imaging capabilities.

Syntax: `arccos(x<,'silent'>)<:rad,deg>`

Description: Calculates the arc cosine value of a real number. The answer is given, in radians and degrees, in the top VnmrJ display window and is optionally returned to two destination variables. The calculation is based on the identity \( \arccos(x) = \arctan(\sqrt{1-x^2}) \). Since arccos calls the macro `arctan` rather than the built-in math function `atan`, the calculation is somewhat slow.

Arguments: `x` is a real number in the range of \( \pm 1.0 \).

- `'silent'` is a keyword to suppress the display of the results in the top VnmrJ display window.
- `rad` is a return value in radians.
- `deg` is a return value in degrees.

Examples: `arccos(.5)`  
`arccos(-.2,'silent'):r1,d1`

See also: *VnmrJ Imaging NMR*

Related:  
- `acos` Find arc cosine of number (C)
- `arcsin` Calculate arc sine of a real number (M)
- `arctan` Calculate arc tangent of a real number (M)
- `atan` Find arc tangent of a number (C)

**arcsin**  
**Calculate arc sine of real number (M)**

Applicability: Systems with imaging capabilities.

Syntax: `arcsin(x<,'silent'>)<:rad,deg>`

Description: Calculates the arc sine value of a real number. The answer is given, in radians and degrees, in the top VnmrJ display window and is optionally returned to two destination variables. The calculation is based on the identity \( \arcsin(x) = \arctan(x / \sqrt{1-x^2}) \). Since arcsin calls the macro `arctan` rather than the built-in math function `atan`, the calculation is somewhat slow.

Arguments: `x` is a real number in the range of \( \pm 1.0 \).

- `'silent'` is a keyword to suppress the display of the results in the top VnmrJ display window.
- `rad` is a return value in radians.
- `deg` is a return value in degrees.

Examples: `arcsin(.5)`  
`arcsin(-.2,'silent'):r1,d1`

See also: *VnmrJ Imaging NMR*

Related:  
- `acos` Find arc cosine of number (C)
- `arcsin` Calculate arc sine of a real number (M)
- `arctan` Calculate arc tangent of a real number (M)
- `atan` Find arc tangent of a number (C)

**arctan**  
**Calculate arc tangent of real number (M)**

Applicability: Systems with imaging capabilities.

Syntax: `arctan(x<,'silent'>)<:rad,deg>`

Related:  
- `acos` Find arc cosine of number (C)
- `arcsin` Calculate arc sine of a real number (M)
- `arctan` Calculate arc tangent of a real number (M)
- `atan` Find arc tangent of a number (C)
Description: Calculates the arc tangent value of a real number. The answer is given, in radians and degrees, in the top VnmrJ display window and is optionally returned to two destination variables. The calculation is based on a rational approximation.

Arguments: \( x \) is a real number.

'\texttt{silent}' is a keyword to suppress the display of the results in the top VnmrJ display window.

\( \text{rad} \) is a return value in radians.

\( \text{deg} \) is a return value in degrees.

Examples: \( \text{arctan}(0.5) \)
\( \text{arctan}(-0.2, '\texttt{silent}') : r1, d1 \)

See also: \textit{VnmrJ Imaging NMR}

Related:
- \texttt{arccos} Calculate arc cosine of a real number (M)
- \texttt{arcsin} Calculate arcsine of a real number (M)
- \texttt{asin} Find arc sine of number (C)
- \texttt{atan} Find arc tangent of a number (C)

\begin{description}
\item[{array}] \textbf{Easy entry of linearly spaced array values (M)}
\item[Syntax:] \texttt{array<\{parameter<,number_steps,start,step_size\}>}
\item[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 \texttt{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.
\item[Arguments:] \texttt{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.
\texttt{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.
\texttt{start} is the starting value of the parameter array. The default is an interactive mode in which you are prompted for the starting value.
\texttt{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.
\item[Examples:] \texttt{array}
\texttt{array('pw')}
\texttt{array('tof', 40, 1400, -50)}
\item[See also:] \textit{VnmrJ Liquids NMR}
\end{description}

\begin{description}
\item[{array}] \textbf{Parameter order and precedence (P)}
\item[Description:] Whenever an array of one or more parameters is set up, the string parameter \texttt{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 \texttt{array} is automatically updated when acquisition parameters are set. “Diagonal arrays” (those corresponding to using parentheses in the parameter \texttt{array}) must be entered by hand.
\item[Values:] ' ' (two single quotes with no space between) indicates no parameter is arrayed.
'x' indicates the parameter \( x \) is arrayed.
'\texttt{x,y}' indicates the parameters \( x \) and \( y \) are arrayed, with \( y \) taking precedence. That is, the order of the experiments is \( x_1y_1, x_1y_2, \ldots, x_1y_n, x_2y_1, x_2y_2, \ldots, x_2y_n, \ldots, x_my_n \), with a total of \( m \times n \) experiments being performed.
\end{description}
'\(y, x\)' indicates the parameters \(x\) and \(y\) are arrayed, with \(x\) taking precedence. That is, the order of the experiments is \(x_1y_1, x_2y_1, \ldots, x_ny_1, x_1y_2, x_2y_2, \ldots, x_ny_2, \ldots, x_ny_m\), 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 \(x_1y_1, x_2y_2, \ldots, x_ny_n\), 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: VnmrJ Liquids NMR

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

arraydim Dimension of experiment (P)

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

See also: VnmrJ Liquids NMR

Related: \texttt{calcdim} Calculate dimension of experiment (C)
\texttt{celem} Completed FID elements (P)

asin Find arc sine of number (C)

Syntax: \texttt{asin(value)<:n>}

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

Arguments:
\texttt{value} is a number in the range of \(\pm 1.0\).
\texttt{n} is a return argument giving the arc sine, in radians, of \texttt{value}. The default is to display the arc sine value in the status window.

Examples: \texttt{asin(.5)}
\texttt{asin(val):asin_val}

See also: VnmrJ User Programming

Related: \texttt{sin} Find sine value of an angle (C)

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

Syntax: \texttt{asize}

Description: Adjusts the 2D display parameters (\texttt{sc}, \texttt{wc}, \texttt{sc2}, and \texttt{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: VnmrJ Liquids NMR

Related: \texttt{sc} Start of chart (P)
\texttt{sc2} Start of chart in second direction (P)
\texttt{wc} Width of chart (P)
\texttt{wc2} Width of chart in second direction (P)

assign Assign transitions to experimental lines (M)

Syntax:
(1) \texttt{assign\('<\text{mark}'\)}
(2) \texttt{assign\(\text{(transition_number, line_number)\)}}
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: VnmrJ Liquids NMR

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)

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.

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>
atan2

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 \( \pi/2 \) and \( -\pi/2 \) 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)**

Description: `atcmd<(<macro'><,'timespec'><,'day'><,'cancel'>)>`

Syntax: The atcmd macro calls a macro at the specified time. It only functions on a spectrometer. A background VnmrJ is started to execute the command. This background VnmrJ is not started in an experiment; therefore, the macro executes a jexp or runs commands or macros that do not need experiment parameters. It will have access to global and systemglobal parameters.

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 envirnoment 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')
atcmd('echo(`What are you doing here on a weekend?`)','8:00 Sat Sun')
atcmd('startNightQueue','22:00')
```

Displays a welcome message every weekday at 8:00 am.

Questions your intentions on the weekend.

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:** `VnmrJ Liquids NMR`

**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 B₁ 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)`

### 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:
```plaintext
au
au('wait')
```

See also: VnmrJ Liquids NMR

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.

**Related:**
- `AuCALch3i1` Get autocalibration with CH3I sample (M)
- `gcal` Gradient calibration constant (P)

**AuCALch3i1** Get autocalibration with CH3I 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.

**Related:**
- `AuCALch3i` Set up autocalibration macros with CH3I sample (M)
- `gcal` Gradient calibration constant (P)

**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), \texttt{gcal} and C/H gradient ratio. The \texttt{AuCALch30h} macro is the same as the \texttt{AuCALch30h1} macro.

Related: \texttt{AuCALch3oh} Autocalibration macros with Autotest sample (M)
\texttt{gcal} Gradient calibration constant (P)

\textbf{AuCALch3oh1} \hspace{1em} \textbf{Get autocalibration with Autotest sample (M)}

Syntax: \texttt{AuCALch3oh1}

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

Related: \texttt{AuCALch3oh} Autocalibration macros with Autotest sample (M)
\texttt{gcal} Gradient calibration constant (P)

\textbf{Aucalibz0} \hspace{1em} \textbf{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 \texttt{lkhzdac} parameter.

Related: \texttt{Augmapz0} Automatic lock gradient map generation and Z0 calibration (M)
\texttt{Aufindz0} Automatic adjustment of Z0 (M)

\textbf{AuCdec} \hspace{1em} \textbf{Carbon decoupler calibration macro (M)}

Syntax: \texttt{AuCdec}

Description: Used by \texttt{AuCALch3i} and \texttt{AuCALch3oh} autocalibration routines to do carbon decoupler calibrations. Calibrates high-power pulse widths and \texttt{dmf}.

Related: \texttt{AuCALch3i} Get autocalibration with CH$_3$I sample (M)
\texttt{AuCALch3oh} Get autocalibration with Autotest sample (M)
\texttt{dmf} Decoupler modulation frequency for first decoupler (P)

\textbf{AuCgrad} \hspace{1em} \textbf{Carbon/proton gradient ratio calibration macro (M)}

Syntax: \texttt{AuCgrad}

Description: Used by \texttt{AuCALch3i1} and \texttt{AuCALch3oh1} autocalibration routines for C/H gradient ratio calibrations.

Related: \texttt{AuCALch3i1} Get autocalibration with CH$_3$I sample (M)
\texttt{AuCALch3oh1} Get autocalibration with Autotest sample (M)

\textbf{AuCobs} \hspace{1em} \textbf{Carbon observe calibration macro (M)}

Syntax: \texttt{AuCobs}

Description: Used by \texttt{AuCALch3i1} autocalibration routines for carbon observe calibrations.

Related: \texttt{AuCALch3i1} Get autocalibration with CH$_3$I sample (M)
audiofilter  Audio filter board type (P)

Applicability: All systems except MERCURYplus/Vx.

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 CONFIG 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 CONFIG window).
'e' indicates the system has a 100-kHz elliptical filter board (100 kHz Elliptical choice in the CONFIG window).
'2' indicates the system has a 200-kHz Butterworth filter board (200 kHz Butterworth choice in the CONFIG window).
'5' indicates the system has a 500-kHz elliptical filter board (500 kHz Elliptical choice in the CONFIG 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.

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.

Related: AuCALch3i1 Get autocalibration with CH3I sample (M)
AuCALch3oh1 Get autocalibration with Autotest sample (M)
gcal Gradient calibration constant (P)

gsize Number of z-axis shims used by gradient shimming (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 D2O sample. This macro is used by the Aumakegmap auto gradient map generation macro and is applicable only for a lock gradient map.

Related: Aumakegmap Auto lock gradient map generation (M)
gsize 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$_2$O sample, the augmapz0 macro automatically creates a lock gradient map, followed by Hz to DAC calibration of Z0 for the autolocking procedure.

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.

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.

Related: AuCALch3i  Get autocalibration with CH3I sample (M)  
         AuCALch3oh  Get autocalibration with Autotest sample (M)

**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$_2$O 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')

Related: auto
         auto_au
         autodir
         autogo
         autoname

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.

Related: auto
         auto_au
         autogo
         autoname

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.

Related: au
         auto
         autolist
         exec
         loc
         lookup
         solvent
         wexp
**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.

**Related:** AuCALch3i1 Get autocalibration with CH3I sample (M)
AuCALch3oh1 Get autocalibration with Autotest sample (M)

**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.

**Related:** adept Automatic DEPT analysis and spectrum editing (C)
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:** VnmrJ Liquids NMR

**Related:** auto Set up an automation directory (C)
autoname Prefix for automation data file (P)

**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 path name of the automation directory. The default is the current value of the parameter autodir.

**Examples:**

```
autogo
autogo('MySamples')
autogo('MySamples','/home/vnmr1/AutoRun_621')
```
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: VnmrJ Liquids NMR
during an automation run. If an alternate parameter is used, it will probably need to be created in the global tree as a string.

$path$ is a return argument with the path. If no return argument is present, the result is displayed on line 3.

Examples:

- autoname:$autoname_path
- autoname('curexp+/text'):$p1

See also: VnmrJ Liquids NMR

Related:

- auto: Set up an automation directory (C)
- autogo: Start automation run (C)
- autodir: Automation directory absolute path (P)
- autoname: Prefix for automation data file (P)
- enter: Enter sample information for automation run (M)
- status: Display status of sample changer (C,U)

**autoname**

**Prefix for automation data file (P)**

Applicability: Systems with an automatic sample changer.

Description: Stores a string in the global tree that determines a prefix to the file name of the FID data (e.g., 0204.fid) during an automation run. Percent signs (%) are used to delimit a string to search for in the sampleinfo file, and the word after the delimited string is used in the file name. This word can be terminated with a space, tab, or carriage return. Dollar signs ($) delimit a string to search for a parameter to be used in the filename. Text not delimited by percent or Dollar signs is copied from autoname without any changes.

If autoname does not start with a slash mark (/), the file is stored in the path given by autodir; otherwise, the name is used as is. The sample number is not automatically appended, but a revision number is appended.

Values: If autoname is a null string, the file name %SAMPLE#:%%PEAK#: is the default, resulting in the name sample_number+revision_number.fid (LC-NMR uses PEAK#: in the sampleinfo file, resulting in the name peak_number+revision_number.fid). Note that the autoname of the user doing the automation run is used for all file names and that the resulting path and file name must be accessible (with read-write permission) by that user.

autoname controls the version number attached to the name of a file and uses the value of VnmrJ parameters as part of the file’s name. For example, autoname='$seqfil$_$tn$ names a file with the current value of the parameters seqfil and tn. The resulting file name might be s2pul_H1 or dept_C13. If a numeric value is used, this value is truncated to an integer. For example, if autoname='$sfrq$', the file name would be 500, not 500.456.

%Rn%, where n is 0 to 9 (default is 2) is a special substitute string. n determines how the revision number is appended to the file name:

- If n is 0, no revision digits are appended (all names must be uniquely constructed without these revision digits).
- If n is 1 to 9, the revision number is padded with leading zeroes to form an n-digit number. If more places are needed than specified, more zeroes are used.

If n is greater than 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 autoname string; the revision digits are always appended.

You can also 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.
% keywords are allowed for time specification:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%DATE%</td>
<td>YYYYMMDD</td>
<td>4-digit year</td>
</tr>
<tr>
<td>%TIME%</td>
<td>HHMMSS</td>
<td>2-digit hour</td>
</tr>
<tr>
<td>%YR%</td>
<td>YYYY</td>
<td>4-digit year</td>
</tr>
<tr>
<td>%YR2%</td>
<td>YY</td>
<td>2-digit year</td>
</tr>
<tr>
<td>%MO%</td>
<td>MM</td>
<td>2-digit month</td>
</tr>
<tr>
<td>%DAY%</td>
<td>DD</td>
<td>2-digit day</td>
</tr>
<tr>
<td>%HR%</td>
<td>HH</td>
<td>2-digit hour</td>
</tr>
<tr>
<td>%MIN%</td>
<td>MM</td>
<td>2-digit month</td>
</tr>
<tr>
<td>%SEC%</td>
<td>SS</td>
<td>2-digit second</td>
</tr>
</tbody>
</table>

Examples: Using the `enter` program, a sample is entered with the following information (which is copied to the sampleinfo file):

```
SAMPLE#: 3
MACRO: h1
USER: John Doe
SOLVENT: CDCl3
TEXT: EthylBenzene in CDCl3
Page 01-3015
This is a text
USERDIR: ...
```

This entry creates the following file names for each autoname string:

<table>
<thead>
<tr>
<th>autoname string</th>
<th>File name created</th>
</tr>
</thead>
<tbody>
<tr>
<td>''</td>
<td>0301.fid</td>
</tr>
<tr>
<td>%USER:%</td>
<td>John01.fid</td>
</tr>
<tr>
<td>%Page%</td>
<td>01-301501.fid</td>
</tr>
<tr>
<td>%USER:%/%Page%</td>
<td>John/01-301501.fid</td>
</tr>
<tr>
<td>'/export/home/%TEXT:%'</td>
<td>/export/home/EthylBenzene01.fid</td>
</tr>
<tr>
<td>%USER:%%R0%</td>
<td>John.fid</td>
</tr>
<tr>
<td>%USER:%-%R5%</td>
<td>John-00001.fid</td>
</tr>
<tr>
<td>%USER:%-%R1%</td>
<td>John-10.fid (if tenth revision)</td>
</tr>
</tbody>
</table>

See also: *VnmrJ Liquids NMR*

Related:
- `auto` Set up an automation directory (C)
- `autogo` Start automation run (C)
- `autodir` Automation directory absolute path (P)
- `autoname` Create path for data storage (C)
- `enter` Enter sample information for automation run (C)
- `status` Display status of sample changer (C,U)

**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 bootup), 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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: `procarray` Process arrayed 1D spectra (M)  
`piarray` 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`).

Related: `explist` Display approximate time for current experiment chain (M)
Set abs. value mode in directly detected dimension (C)

Syntax: \texttt{av}

Description: Selects the absolute-value spectra display mode by setting the parameter \texttt{dmg} to the string value \texttt{'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, \texttt{av} has no effect on data prior to the second Fourier transform. If \texttt{pmode='full'}, \texttt{av} acts in concert with commands \texttt{ph1}, \texttt{av1}, or \texttt{pwr1} to yield the resultant contour display for the 2D data.

See also: VnmrJ Liquids NMR

Related: \texttt{av1} Set abs. value mode in 1st indirectly detected dimension (C) \texttt{av2} Set abs. value mode in 2nd indirectly detected dimension (C) \texttt{dmg} Display mode in directly detected dimension (C) \texttt{dmgf} Absolute-value display of FID data or spectrum in \texttt{acqi} (P) \texttt{ft} Fourier transform 1D data (C) \texttt{ft1d} Fourier transform along \texttt{f2} dimension (C) \texttt{ft2d} Fourier transform 2D data (C) \texttt{pa} Set phase angle mode in directly detected dimension (C) \texttt{pal} Set phase angle mode in 1st indirectly detected dimension (C) \texttt{ph} Set phased mode in directly detected dimension (C) \texttt{ph1} Set phased mode in 1st indirectly detected dimension (C) \texttt{pmode} Processing mode for 2D data (P) \texttt{pwr1} Set power mode in 1st indirectly detected dimension (C) \texttt{wft} Weigh and Fourier transform 1D data (C) \texttt{wft1d} Weigh and Fourier transform of 2D data (C) \texttt{wft2d} Weigh and Fourier transform 2D data (C)

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

Syntax: \texttt{av1}

Description: Selects the absolute-value spectra display mode along the first indirectly detected dimension by setting the parameter \texttt{dmg1} to the value \texttt{'av1'}. If the parameter \texttt{dmg1} does not exist, \texttt{av1} creates it and set it to \texttt{'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 \texttt{av1} command is only needed if mixed-mode display is desired. If the parameter \texttt{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 \texttt{dmg}). For the contour display of multidimensional data, the result of \texttt{av1} is the same as for traces provided that \texttt{pmode='partial'} or \texttt{pmode=''} (two single quotes with no space between).

See also: VnmrJ Liquids NMR

Related: \texttt{av} Set abs. value mode in directly detected dimension (C) \texttt{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: *VnmrJ Liquids NMR*

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 `f2` dimension in 2D data sets, the `f3` 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: *VnmrJ Liquids NMR*

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:** *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**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')
```

**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`, `scaleswl`, 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)
- `scaleswl` 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 (f2), and the second letter describing the indirectly detected axis (f1). 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 (f3), the second letter describing the first indirectly detected axis (f1), and the third letter specifying the second indirectly detected axis (f2).

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’`.

For image display, `axis` can have values ‘c’ (for centimeters), ‘m’ (for millimeters), and ‘u’ (for microns). These values rely on the parameters `lro` and `lpe` for scaling. If both f1 and f2 dimensions are spatial, the display aspect ratio is adjusted to retain the aspect ratio of the imaging.

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: *VnmrJ Liquids NMR*

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)
- `lpe` Field of view parameter for phase encode, in cm (P)
- `lro` Field of view parameter for readout, in cm (P)
- `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: *VnmrJ Liquids NMR*

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)
**B0**

**Magnet main static field (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** The field strength, in gauss, of the main magnetic field. This value is used by planning macros in their calculations.

**Values:** Number, in units of gauss. Nominal value is $234.9 \times h1freq$. For example, a 4.7T (200 MHz) system has a value of approximately 47,000.

**See also:** *VnmrJ Imaging NMR*

**Related:** *h1freq* Proton frequency of spectrometer (P)

**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.

See also: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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*

**binom**

Set up parameters for BINOM pulse sequence (M)

Applicability: Sequence is not supplied with MERCURYplus/-Vx.

Description: Sets up a binomial water suppression pulse sequence.

See also: *VnmrJ Liquids NMR*

**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*

**boresize**

Magnet bore size (P)

Applicability: Systems with imaging capabilities.

Description: Holds the internal usable diameter of the gradient set. This parameter is used by various pulse sequence setup macros to determine the validity of the field of view and slice offset input. It is defined in the system gradient table files found in `$vnmrsystem/imaging/gradtables`, and is automatically set from one of those files when a value is entered for `gcoil`.

Values: 18, 31, 33, 40 (nominal, in cm)

See also: *VnmrJ Imaging NMR*

**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' or '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.

\( x_{1\text{mm}} \) is the left edge of the box, \( x_{2\text{mm}} \) is the right edge, \( y_{1\text{mm}} \) is the bottom, and \( y_{2\text{mm}} \) 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 \( wc2. \)

\( r_{1}, r_{2} \) return the location of the upper left corner of the box.

Examples:

\[
\text{box('plotter',20,100,40,150)}
\]

\[
\text{box(25,105,45,155,'nolimit'):r1,r2}
\]

See also: VnmrJ Liquids NMR

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

\[
\text{boxes('plotter')}
\]

See also: VnmrJ Liquids NMR

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 bpa and ppa 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:  VnmrJ Liquids NMR

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. Sequence not supplied with MERCURY plus/Vx.

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)

browser  Start Image Browser application (U)

Applicability:  Systems with imaging capabilities.

Syntax:  (From UNIX) browser <macro_name> <XView_arguments> <image path> <imagelist path>

Description:  Starts up the Image Browser application. Image Browser requires the environment variable BROWSERDIR to be set to point to the user’s directory ib_initdir, which contains initialization files and directories. The environment variable and the initialization directory can be created when the makeuser command is run.

Image Browser reads in files in Flexible Data Format (FDF) for displaying and processing. To generate files in FDF format, the following macros are available to write out single or multislice images:

- For the current imaging software, which includes sequences sems, mems, and flash, use the svib macro.
- For older style SIS imaging sequences and microimaging sequences, use the macro svis.
- 3D data can be saved in the FDF format by the ft3d macro.

The FDF format is an ASCII header describing the data, followed by the data. For more information on FDF, see the User Programming manual.

After images are read into Image Browser, image data can be written in a number of other formats for use with other imaging applications. browser can be used to extract up to three Maximum Intensity Projections (MIPs).
Arguments: Arguments can appear in any order.

macro_name is the file name of a macro, which must be stored in $BROWSERDIR/macro/macro_name. The macro is executed when Image Browser starts. If no macro name is specified, the macro startup is executed.

XView_arguments are any type of standard XView arguments, which can be found by typing man xview on a UNIX command line.

-image path specifies the path of an image that should be loaded at startup.

-it is loaded after the startup macro is executed. Multiple -image arguments can be used to load multiple images.

-image list path specifies the path of a file containing a list of image files to be loaded.

See also: VnmrJ Imaging NMR; User Programming

Related: fdfgluer Make FDF file from header and data parts (C)

ft3d Perform a 3D Fourier transform on a 3D FID data set (M,U)

svib Generate and save images as Image Browser FDF files (M)

svsis Generate and save images as FDF files (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: VnmrJ Liquids NMR

Related: wbs Specify action when bs transients accumulate (C)

wbs When block size (P)

btune

Tune broadband channel on MERCURYplus/-Vx (M)

Applicability: MERCURYplus/Vx systems

Description: Turns on the broadband transmitter, directing to the probe about 0.5 watts of rf at frequency sfrq, enabling the user to tune the probe coil. Before entering btune, be sure to move the proper cable on the back of the left-hand magnet leg to the BNC connector labeled TUNE, and also to move the proper cable leading to the probe to the BNC connector labeled TUNE. Enter tuneoff to turn off the transmitter. btune cannot be executed while the console is acquiring. For the full tuning procedure, see the probe installation manual.

See also: VnmrJ Liquids NMR; Autoswitchable NMR Probes Installation

Related: acqi Interactive acquisition display process (C)

sethw Set values for hardware in acquisition system (C)

sfrq Transmitter frequency of observe nucleus (P)

su Submit a setup experiment to acquisition (M)

tuneoff Turn off probe tuning mode, MERCURYplus/-Vx (M)
c13  Automated carbon acquisition (M)
c13p  Process 1D carbon spectra (M)
calcdim  Calculate dimension of experiment (C)
calpha  Recalculate alpha so that first-order phase is zero (M)
calibflag  Correct systematic errors in DOSY experiments (P)
calibrate  Start a dialog for autocalibration routines (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)
celelem  Completed FID elements (P)
center  Set display limits for center of screen (C)
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)
cfpmult  Calculate first-point multiplier for 2D experiments (M)
change  Submit a change sample experiment to acquisition (M)
Cigar2j3j  Convert the parameter to a CIGAR2j3j experiment (M)
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 subexperiment (M)
clearStacks()  Clear stack (C)
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)
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)
continueMovie  Continue movie in either forward or backward direction (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 \texttt{c13} macro on the MACRO line by following it with additional commands and parameters. For example, \texttt{c13 nt=1} uses the standard \texttt{c13} setup but with only one transient.

Arguments: \texttt{solvent} is the name of the solvent. In automation mode the solvent is supplied by the \texttt{enter} program. The default is \texttt{'CDC13'}.

Examples: \texttt{c13} \
\hspace{1em} \texttt{c13('DMSO')}

See also: \textit{VnmrJ Liquids NMR}

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

\texttt{c13p} \hspace{1em} \textbf{Process 1D carbon spectra (M)}

Syntax: \texttt{c13p}

Description: Processes non-arrayed 1D carbon spectra using a set of standard macros. \texttt{c13p} is called by the \texttt{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 (\texttt{aphx} macro), automatic integration (\texttt{integrate} macro if required only), vertical scale adjustment (\texttt{vsadjc} macro), avoiding excessive noise (\texttt{noislm} macro), threshold adjustment (\texttt{thadj} macro), and referencing to the TMS signal if present (\texttt{setref} macro then \texttt{tmsref} macro).

See also: \textit{VnmrJ Liquids NMR, VnmrJ Liquids NMR}

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

\texttt{calcdim} \hspace{1em} \textbf{Calculate dimension of experiment (C)}

Syntax: \texttt{calcdim}

Description: Calculates the dimension of an experiment and puts the result into the parameter \texttt{arraydim}. If an experiment is arrayed, \texttt{arraydim} is the product of the size of the arrays.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{arraydim} Dimension of experiment (P)

\texttt{calfa} \hspace{1em} \textbf{Recalculate alfa so that first-order phase is zero (M)}

Syntax: \texttt{calfa}

Description: Based upon the current \texttt{alfa} and \texttt{lp} values, \texttt{calfa} calculates a new value for \texttt{alfa} so that the first-order phase parameter \texttt{lp} is rendered approximately 0. When digital filtering is active (\texttt{dsp='r'} or \texttt{dsp='i'}), \texttt{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: VnmrJ Liquids NMR

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 2p 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: VnmrJ Liquids NMR

Related: dosy Process DOSY experiments (M)

calibrate Start a dialog for autocalibration routines (M)  

Syntax: calibrate  
Description: Starts a dialog for autocalibration routines.

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'.

Examples: capt au  
capt('dmso')

See also: VnmrJ Liquids NMR

Related: apt Prepare parameters for APT experiment (M)  
cl3 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 13C 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: VnmrJ Liquids NMR
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 CONFIG 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 CONFIG window).
79 for standard UNITY INOVA (79 dB choice in CONFIG window).
127 for imaging attenuator (63.5 dB SIS choice in CONFIG window).
63 for UNITY INOVA 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: VnmrJ Liquids NMR
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.
**cdept**

**Automated carbon and DEPT acquisition (M)**

**Syntax:** cdept<(solvent)>

**Description:** Prepares parameters for automatically acquiring a standard $^{13}\text{C}$ 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:** *VnmrJ Liquids NMR*

**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 it 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: `ps` for PostScript formatted output, `jpeg` for Joint Photographic Experts Group JFIF formatted output, `png` 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:** *VnmrJ Liquids NMR*

**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)
**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: *VnmrJ Liquids NMR*

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)

**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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`xxx macro so that the newly created experiment can be joined.
cexp

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: VnmrJ Liquids NMR

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: VnmrJ Imaging NMR

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 “t2 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 t2 FID and the t1 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: VnmrJ Liquids NMR

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: *VnmrJ Liquids NMR*

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)

Cigar2j3j Convert the parameter to a CIGAR2j3j experiment (M)

Syntax: Convert the parameter to a CIGAR2j3j experiment.

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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 subexperiment (M)**

**Syntax:** `cleardosy`

**Description:** Deletes any copies of DOSY data temporarily saved in the current subexperiment.

See also: *VnmrJ Liquids NMR*

**Related:**
- **dosy**: Process DOSY experiments (M)

### clearStacks()

**Clear stack (C)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** `clearStacks()`

**Description:** Deletes all stacks.

---

112  VnmrJ 1.1D Command and Parameter Reference  01-999252-00  A0604
See also: *VnmrJ Imaging NMR*

Related: *gplan*  Start interactive image planning (C)

**clf**
*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 *st*. Enter *dla* to display *clf*.

See also: *VnmrJ Liquids NMR*

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)
- *st*  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: *VnmrJ Liquids NMR*

Related:
- *assign*  Assign transitions to experimental lines (M)
- *cla*  Clear line assignments (M)
- *cla*  Calculated transition number (P)
- *dla*  Display line assignments (M)

**clr**
*Clear add/subtract experiment (C)*

**Description:** Deletes the add/subtract experiment (*exp5*).

See also: *VnmrJ Liquids NMR*

Related:
- *add*  Add current FID to add/subtract experiment (C)
- *sub*  Subtract current FID from add/subtract experiment (C)

**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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 `procpar`. `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: *VnmrJ Liquids NMR*

Related: `dp` Double precision (P)

**config** Display current configuration and possibly change it (M)

Syntax: `config <('display')>`

Description: Displays the current system configuration parameters in a window (called the CONFIG window). The values of the configuration parameters can be changed if `config` is entered from the console without any arguments and the user has write access to the directories `/vnmr` and `/vnmr/conpar`. If so, the user can interactively make changes to the choices in the window.

If the user does not meet the conditions above, or if the VnmrJ administrator enters the command `config('display')`, instead of the interactive mode, the user is restricted to the display mode, where system information is listed in the Process tab -> Text page. On MERCURYplus/Vx, the mode is always interactive.
If `config` is entered without any arguments, or if Utilities->System Settings is selected, the program checks if the user is logged in as the administrator. If so, it runs in interactive mode; if not, it runs in display mode. By entering `config('display')`, vnmr1 can run in the display mode instead of interactively.

In the interactive mode, a separate panel displays the options with the current choice appearing to the right. Position the mouse over the choice to be modified, then use the left button to cycle through each choice or use the right button to display a menu of all possible choices.

The Use Console Data button sets parameter values in the CONFIG window using information captured during console startup.

- On **UNITY** INOVA, this button makes `config` capture from the system all values shown in the CONFIG window except Sample Changer, Sample Changer Serial Port, Rotor Synchronization, Frequency Overrange, and Upper Limit of decoupler power. For the Gradients entry, `config` recognizes the Performa I and Performa II modules but not other gradients. For the VT Controller entry, if VT is found, `config` does not change the value set, and if VT is not found, `config` changes the value to Not Present.

- On **MERCURY**plus/-Vx systems, this button captures all the values except Sample Changer and Sample Changer Serial Port. The VT Controller entry is set the same way as **UNITY** INOVA systems (see above).

The EXIT, and SAVE button writes a new `conpar` configuration file before leaving. The QUIT, no SAVE button terminates the session with no modifications to the `conpar` file, but remember that the parameters are always set. These two buttons require use of the left button on the mouse. In the display mode, the current choices are displayed in the text window. To send output to the printer, enter the sequence of commands `printon config('display') printoff`.

Commands for working with parameters (such as `create`, `destroy`, `exists` and `setvalue`) have an option to select which parameter tree the parameter is in. The `systemglobal` tree is the internal name for `/vnmr/conpar`, and it can be used to search for, modify, or create a parameter in `conpar`. But note that any changes made, either directly (e.g., by typing `vttype=0`) or by using `create` and similar commands, only affect parameters in memory. To permanently change parameters:

- For parameters in `config`, enter the change in the CONFIG window and then quit using the Exit & Save button.

- For other parameters, after creating or changing the parameter, enter `fsave('/vnmr/conpar','systemglobal')`.

Both methods, usually restricted to `vnmr1` only, overwrite `conpar`.

The CONFIG labels listed below can be changed in the interactive mode. For each label, the choices available and a short description of the label is provided. Shown in parentheses is the associated parameter, which you should refer to for further information.

**CONFIG window for **UNITY**INOV4 and Imaging systems:**

- **System Type:** Spectrometer or Data Station. Sets the basic type of system (**system**).

- **Console:** **UNITY** INOVA, **MERCURY**plus/-Vx, or Imager. Sets the type of system console (**Console**). When `go`, `au`, or `ga` is entered, the value set is copied to the current experiment as the `console` parameter (lowercase c).
Proton Frequency: 085, 100, 200, 300, 400, 500, 600, 700, 750, 800, 900, 3T, and 4T. Sets the resonant frequency, in MHz or tesla, of \(^1\text{H}\) as determined by magnet field strength (\text{hfreq}).

Sample Changer: For \textit{UNITY/INOVA} – None, Carousel, SMS 50 Sample, SMS 100 Sample, VAST, NMS, LC-NMR, 768 AS. Sets the type of sample changer. Set to none if a sample changer is not present or is to be disabled (\text{traymax}).

Sample Changer Comm Port: Not Used, Port A, Port B, Ethernet. Sets the serial port used to connect the sample changer. Select Not Used if no sample changer is present (\text{smsport}).


Audio Filter Type: 100 kHz Elliptical, 100 kHz Butterworth 200 kHz Butterworth, 500 kHz Elliptical. If the spectral width (sw) is less than 100 kHz, sets type of audio filters used (\text{audiofilter}).

VT Controller: Not Present, Present. Sets whether a variable temperature controller is present or not on the system (\text{vttype}).

Maximum DMF: 9900, 32700, 2.0e6. Sets maximum frequency, in Hz, for decoupler modulation (\text{parmax}[11]).

Max. Spectral Width: 100 kHz, 200 kHz, 500 kHz, 2 MHz, 5 MHz. Sets maximum spectral width available to a system (\text{parmax}[5]).

Max. Narrowband Width: 100 kHz, 200 kHz, 500 kHz. Defines the maximum spectral width of the Input board (\text{maxsw_loband}).

AP Interface Type: Type 1, Type 2, Type 3, N/A. Sets type of AP bus interface board in the system.

Fifo Loop Size: 63, 1024, 2048. Sets size of FIFO loop, which depends on the type of controller board in the system.

Rotor Synchronization: Not Present, Present. Sets whether system supports the solids rotor synchronization module (\text{rotorsync}).

Lock Frequency: (frequency entered directly). Sets lock frequency of the system. To observe NMR signals, the lock frequency value must be set correctly (\text{lockfreq}).

IF Frequency: 10.5 MHz, 20.0 MHz.

Number of RF Channels: 1, 2, 3, 4, 5. Selects which rf channel is listed in the Configure panel that appears in the lower section of the CONFIG window (\text{numrfch}).

Gradients: Not Present, Present. Sets whether system has optional gradients for the X, Y, or Z axis. If present, the gradients are listed in the Configure panel in lower section of CONFIG window (Gradients is not associated with any parameter).

Configure: RF Channel 1 (Obs), RF Channel 2 (Dec), RF Channel 3 (Dec2), RF Channel 4 (Dec 3), RF Channel 5 (Dec4), Gradients. Sets which labels appear in the Configure panel in lower section of CONFIG window (Configure is not associated with any parameter).

Type of RF: U+ Direct Synthesis, U+ H1 Only, Direct Synthesis, Broadband, Fixed Frequency, Deuterium Decoupler (\textit{UNITY/INOVA} only),
SIS Modulator. Sets type of frequency generation on the current rf channel (\texttt{rfctype} and \texttt{rfchtype}).

- Synthesizer: Not Present, PTS 160, PTS 200, PTS 250, PTS, 320, PTS 500, PTS 620, PTS 1000. Sets type of PTS frequency synthesizer on the current rf channel (\texttt{ptsval}).

- Latching: Not Present, Present. On systems equipped with a special version of the PTS frequency synthesizer, sets how frequency values are sent on the current rf channel (\texttt{latch}).

- Frequency Overrange: Not Present, 10000 Hz, 100000 Hz. On systems equipped with a special version of the PTS frequency synthesizer, sets the presence of a signal phase stability option on the current rf channel (\texttt{overrange}).

- Step Size: 0.1 Hz, 0.2 Hz, 1 Hz, 100 Hz. Sets frequency step size on current rf channel. (\texttt{parstep[7]}, \texttt{parstep[8]}, \texttt{parstep[16]}, \texttt{parastep[20]}).

- Coarse Attenuator: Not Present, 63 dB, 79 dB, 63.5 dB (SIS). Sets range of coarse attenuator if this attenuator is present on the current rf channel (\texttt{cattn}).

- Upper Limit: (number entered directly). Sets upper limit of the coarse attenuator if this attenuator is present on the current rf channel (\texttt{parmax[17]}, \texttt{parmax[9]}, \texttt{parmax[18]}, \texttt{parmax[21]}).

- Fine Attenuator: Not Present, Present. Sets whether a fine attenuator is present or not on the current rf channel (\texttt{fattn}).

- Waveform Generator: Not Present, Present. Sets whether a waveform generator board is present or not on current rf channel (\texttt{rfwg}).

- Type of Amplifier: Class C, Linear Full Band, Linear Low Band, Shared, Linear Broadband. (Shared is fourth channel only.) Sets type of amplifier on the current rf channel (\texttt{amptype}).

- X Axis, Y Axis, Z Axis: None, WFG + GCU, Performa I, Performa II/III, Performa II/III+WFG, Performa XYZ, Performa XYZ+WFG, SIS (12 bit), Homospoil. On systems with gradients, sets type of gradient for each axis. The value is set separately for each axis (\texttt{gradtype}).

- Imaging Gradient Coil. Detects the gradient coil configuration file that defines the current installed gradient coil (\texttt{sysgcoil}).

**CONFIG window for MERCURYplus/-Vx systems:**

Several parameters, other than those listed below, are set automatically because they have only one choice (e.g., \texttt{Console} is set to 'mercury').

- System Type: 4-Nucleus, Broadband. Sets the basic type of system (\texttt{rftype}).

  The MERCURY-Vx 300-MHz 4-Nucleus system uses the Hi/Lo Reference Generator board. For this system, in CONFIG window set System Type to Broadband (\texttt{rftype='fe'}).

  If the board type is unknown, look at the rf card cage in the back of the console. The third rf board from the left is the reference generator. If the top of the board is labeled Hi/Lo, select Broadband, but if it is labeled 4-Nucleus or 5-Nucleus select 4-Nucleus as the system type

- Proton Frequency: 200, 300, 400. Sets the resonant frequency, in MHz, of \textsuperscript{1}H, as determined by magnet field strength (\texttt{hifreq}).

- VT Controller: Not Present, Present. Sets whether a variable temperature controller is present or not on the system (\texttt{vttype}).
Type of Amplifier: 4-Nucleus (35W/35W), Broadband (75W/125W), CP/MAS(100W/300W). Sets type of amplifier in the system (amptype: aa on 4-Nucleus, bb on Broadband, cc on CP/MAS).

Sample Changer: – None, Carousel, SMS 50 Sample, SMS 100 Sample, VAST, NMS. Sets the type of sample changer. Set to None if a sample changer is not present or is to be disabled (traymax).

Sample Changer Comm Port: Not Used, Port A, Port B, Com1. Sets the serial port used to connect the sample changer. Select Not Used if no sample changer is present (smsport).


Pulsed Field Gradient: Not Present, Homospoil, Performa I, Performa II. Sets whether the PFG hardware is present or not on the system (gradtype). Homospoil can be used for gradient shimming, but not for experiments like gHMOC.

Lock Frequency: (number entered directly). Sets the lock frequency of the system. This value must be set correctly to observe NMR signals (lockfreq).

Homodecoupler: Not Present, Present. Sets whether a homonuclear decoupler board is present or not (homdec). Standard on MERCURY-Vx.

Max. Decoupler: (number entered directly). On broadband systems, sets maximum power level for CW decoupling (parmax [9]).

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)
hlfreq Proton frequency of spectrometer (P)
latch Frequency synthesizer latching (P)
lockfreq Lock frequency (P)
maxsw_loband Maximum spectral width of Input board (P)
umrfch 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)
vtype 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: `UNITY`, `INOVA`, `MERCURYplus/Vx`, SISCO Imager. The value is usually set using the Console label in the CONFIG window (opened from `config`); however, on `MERCURYplus/Vx` systems, the value is automatically set.

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.

Values:
- `'inova'` is a `UNITY` `INOVA` console (UnityInova choice in CONFIG window).
- `'mercury'` is a `MERCURYplus/Vx` console.
- `'sisco'` is a SISCO imager console (sisco choice in CONFIG window).

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_time **MAS 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

\[ I = \left[ S0 - (S0 - S_{inf}) \times \exp(-T/Tch)) \times \exp(-T/T1rho) \right] + S_{inf} \]

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)
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 backward, 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.

conv2ta Convert imaging 3D transform to absolute value (U)

Applicability: Systems with imaging capabilities.

Syntax: (From UNIX) conv2ta in_file out_file scaling_factor

Description: Converts a complex 3D transformed data file into a 3D 8-bit absolute value data file suitable for viewing by using disp3d. The conv2ta command reads the header in the transformed file, typically named filename.transform, to determine the dimensions of the data, takes the magnitude of the complex data, scales the data, and writes out only the data (with no header) in 8-bit pixels. It also prints out the dimensions of the file that will be needed by disp3d.

Arguments: in_file is a valid UNIX file name of the 3D transformed data file.
           out_file is a valid UNIX file name of the output file in 8-bit bytes.
           scaling_factor is a value to scale the data so that it is in a range for viewing by disp3d. Reasonable values generally range from 1 to 4000. A value of 1000 is typical.

Examples: (From UNIX) conv2ta kiwi3d.transform kiwi3d.av 1000

See also: VnmrJ Imaging NMR

Related: acqmeter Open Acqmeter window (M)
          acqstat Open Acquisition status window (U)
          disp3d Convert 3D data (U)
          sa Stop acquisition (C)

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: VnmrJ Liquids NMR

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>
<thead>
<tr>
<th><strong>Bruker</strong></th>
<th><strong>Varian</strong></th>
<th><strong>Bruker</strong></th>
<th><strong>Varian</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>sweeps completed</td>
<td>ct</td>
<td>sp</td>
<td>satdly</td>
</tr>
<tr>
<td>td</td>
<td>np</td>
<td>dp</td>
<td>dpwr</td>
</tr>
<tr>
<td>dw</td>
<td>dw</td>
<td>te</td>
<td>temp=te-273</td>
</tr>
<tr>
<td>fw</td>
<td>fb=1.1*sw/2</td>
<td>id</td>
<td>sw1=l/id</td>
</tr>
<tr>
<td>ds</td>
<td>ss</td>
<td>sfo1</td>
<td>sfrq=sfo1+o1</td>
</tr>
<tr>
<td>sw</td>
<td>sw</td>
<td>sfo2</td>
<td>dfreq=sfo2+o2</td>
</tr>
<tr>
<td>experiments done</td>
<td>ni</td>
<td>p#</td>
<td>p#</td>
</tr>
<tr>
<td>o1</td>
<td>tof</td>
<td>d#</td>
<td>d#</td>
</tr>
<tr>
<td>o2</td>
<td>dof</td>
<td>s#</td>
<td>s#</td>
</tr>
<tr>
<td>rd (or d1 if rd=0)</td>
<td>rd</td>
<td>ro</td>
<td>spin</td>
</tr>
<tr>
<td>pw (or p0 if pw=0)</td>
<td>pw</td>
<td>rg</td>
<td>gain</td>
</tr>
<tr>
<td>pl</td>
<td>pw90</td>
<td>date</td>
<td>date</td>
</tr>
<tr>
<td>de</td>
<td>de</td>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>ns</td>
<td>nt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th><strong>Bruker</strong></th>
<th><strong>Varian</strong></th>
<th><strong>Bruker</strong></th>
<th><strong>Varian</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ns (from acqu)</td>
<td>nt</td>
<td>te</td>
<td>temp=te-273</td>
</tr>
<tr>
<td>ns (from acqus)</td>
<td>ct</td>
<td>sfo1</td>
<td>sfrq=sfo1</td>
</tr>
<tr>
<td>td (from acqus)</td>
<td>np</td>
<td>sfo2</td>
<td>dfreq=sfo2</td>
</tr>
<tr>
<td>td (from acqu2s)</td>
<td>ni</td>
<td>o1</td>
<td>tof</td>
</tr>
<tr>
<td>sw_h</td>
<td>sw</td>
<td>o2</td>
<td>dof</td>
</tr>
<tr>
<td>sw_h</td>
<td>dw=1.0e6/sw</td>
<td>ro</td>
<td>spin</td>
</tr>
</tbody>
</table>
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: `VnmrJ Liquids NMR`

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)`

**Description:** Makes a copy of a file using the UNIX `cp` command. All arguments are passed.

**copy** operates the same as the `cp` command.

**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.

**Examples:**

- `copy('-r', '/home/vnmr1/vnmrsys/seqlib', '/vnmr/seqlib')`
- `copy('/home/vnmr1/vnmrsys/seqlib/d2pul', '/vnmr/seqlib/d2pul')`

See also: `VnmrJ Liquids NMR`

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**  
Set up parameters to a COSY pulse sequence (M)

Description: Sets up for a COSY (correlated spectroscopy) experiment.

See also:  
*VnmrJ Liquids NMR*

Related:  
\( \text{cosyps} \)  
Set up parameters for phase-sensitive COSY pulse sequence (M)  
\( \text{dqcosy} \)  
Set up parameters for double-quantum filtered COSY (M)  
\( \text{relayh} \)  
Set up parameters for RELAYH pulse sequence (M)

**Cosy**  
Convert the parameter to a COSY experiment (M)

Description: Convert the parameter to a COSY experiment.

**COSY**  
Change parameters for COSY experiment (M)

Description: Converts the current parameter set to a COSY experiment.

**cosyps**  
Set up parameters for phase-sensitive COSY pulse sequence (M)

Description: Sets up a phase-sensitive COSY (homonuclear correlation) experiment.

See also:  
*VnmrJ Liquids NMR*

Related:  
\( \text{cosy} \)  
Set up parameters for COSY pulse sequence (M)  
\( \text{dqcosy} \)  
Set up parameters for double-quantum filtered COSY (M)  
\( \text{relayh} \)  
Set up parameters for RELAYH pulse sequence (M)

**cp**  
Copy a file (C)

Syntax:  
\( \text{cp}(<'-r',>\text{from\_file},\text{to\_file}) \)

Description: Makes a copy of a file using the UNIX \text{cp} command. All arguments are passed. \text{cp} operates the same as the \text{copy} command.

Arguments:  
\( '-r' \) is a keyword requesting a recursive copy (i.e., copy a directory).  
\text{from\_file} is the name of the file (or directory if \( '-r' \) used) to be copied.  
\text{to\_file} is the name of the copy of the file (or directory). If the \text{from\_file} argument has an extension (e.g., \text{.fid}), be sure the \text{to\_file} argument has the same extension.

Examples:  
\( \text{cp}('/\text{home}/\text{vnmr1}/\text{vnmrsys/seqlib/d2pul}', \ ' '/\text{vnmr/seqlib/d2pul}') \)
\( \text{cp}('-r','/\text{home}/\text{vnmr1}/\text{vnmrsys/seqlib}', '/\text{vnmr/seqlib}') \)

See also:  
*VnmrJ Liquids NMR*

Related:  
\( \text{copy} \)  
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 (\( \text{cp}='y' \)) or 0 (\( \text{cp}='n' \)). The only circumstance where setting \( \text{cp}='n' \).
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)

**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: *VnmrJ Liquids NMR*

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`

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)

**cpx**

*Create pbox shape file (M)*

Syntax: `cpx<(ref_pw90,ref_pwr)>` or `cpx<(g)>`
C

Description: Calls UNIX command \texttt{Pbox}, which generates the specified pulse shape or decoupling/spin locking pattern, as defined by the \texttt{shape\texttt{lib/Pbox.inp}} file.

Arguments: \texttt{ref\_pw90} is the reference 90° pulse width
\texttt{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: \texttt{cpx}
\texttt{cpx('g')}
\texttt{cpx(pw90*compH,tpwr)}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{Pbox} Pulse shaping software (U)

\textbf{cr} \hspace{1cm} \textbf{Cursor position in directly detected dimension (P)}

Description: Contains the current cursor position. The \texttt{rl} macro uses \texttt{cr} to set the reference line.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{centersw} Move cursor to center of spectrum (M)
\texttt{crf} Current time-domain cursor position (P)
\texttt{cr\_l} Clear ref. line in directly detected dimension (M)
\texttt{delta} Difference of two frequency cursors (P)
\texttt{rl} Set reference line in directly detected dimension (M)

\textbf{cr\_l} \hspace{1cm} \textbf{Cursor position in 1st indirectly detected dimension (P)}

Description: Contains the current cursor position along the first indirectly detected dimension. Analogous to the \texttt{cr} parameter except that \texttt{cr\_l} applies to the first indirectly detected dimension of a multidimensional data set. The \texttt{rl\_l} macro uses \texttt{cr\_l} to set the reference line along this dimension.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{centersw\_l} Move cursor to center of spectrum in 1st indirect dimension (M)
\texttt{cr} Cursor position in directly detected dimension (P)
\texttt{cr\_2} Cursor position in 2nd indirectly detected dimension (P)
\texttt{rl\_l} Set ref. line in 1st indirectly detected dimension (M)

\textbf{cr\_2} \hspace{1cm} \textbf{Cursor position in 2nd indirectly detected dimension (P)}

Description: Contains the current cursor position along the second indirectly detected dimension. Analogous to the \texttt{cr} parameter except that \texttt{cr\_2} applies to the second indirectly detected dimension of a multidimensional data set. The \texttt{rl\_2} macro uses \texttt{cr\_2} to set the reference line along this dimension.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{centersw\_2} Move cursor to center of spectrum in 2nd indirect dimension (M)
\texttt{cr} Cursor position in directly detected dimension (P)
\texttt{cr\_l} Cursor position in 1st indirectly detected dimension (P)
\texttt{rl\_2} Set ref. line in 2nd indirectly detected dimension (M)

\textbf{crcom} \hspace{1cm} \textbf{Create user macro without using text editor (M)}

\texttt{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', 'x', '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)

**creategttable** Generate system gradient table (M)

Applicability: Systems with imaging capabilities.

Description: Generates a gradient table in the $vnmr/system/imaging/gradtables directory (/vnmr/imaging/gradtables) needed to run an imaging experiment. The system prompts the user for the boresize of the magnet, the maximum gradient strength (gmax), and the gradient rise time. The directory /vnmr/imaging/gradtables is set up to have group write permission mode for all users; however, the administrator, vnmr1, may want to set the write permission mode for vnmr1 only.

Systems with three-axis pulse field gradients (PFGs) or microimaging gradients might not have the same gradient strength on each axis. If the gradient strength varies, creategttable prompts for the maximum gradient strength for each axis (gxmax, gymax, and gzmax). Additionally, three-axis PFG amplifiers may be limited in their total current output, and hence the gradient strength, when gradients are simultaneously applied on all three axes. If this limitation exists, the user can enter the maximum combined gradient strength, which will be the combination of x+y+z, in gauss/cm.

The macro expects gradient strength entered in gauss/cm, risetime in μs (it is converted to seconds when it is put in the table), and boresize in cm.

Gradient tables are needed when using the obliquing, phase encode, or magic-angle gradient PSG statements.

See also: *VnmrJ Imaging NMR*

Related: gmax Maximum gradient strength (P)
gxmax, gymax, gzmax Maximum gradient strengths for each axis (P)

**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 ('fidi').

Values: Number, in seconds.

See also: *VnmrJ Liquids NMR*

Related: addpar Add selected parameters to the current experiment (M)
crl1 Clear ref. line in 1st indirectly detected dimension (C)
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 andreffrq.
See also: VnmrJ Liquids NMR
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 andreffrq1.
See also: VnmrJ Liquids NMR
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 andreffrq2.
See also: VnmrJ Liquids NMR
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.
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 `alpha` 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)

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

Completed transients (P)

Description: Stores a non-user-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: VnmrJ Liquids NMR

Related: `dg` Display parameters of acquisition/processing group (C)

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: VnmrJ Liquids NMR

Related: `atext` Append string to the current experiment text (M)
`text` Display text or set new text for current experiment (C)
curecc  **Name of eddy current compensation file (P)**

Applicability: Systems with the imaging capabilities.

Description: A global string parameter containing the name of the file containing the last eddy current compensation file set. `eddyend` updates this parameter from ECC Tool window or from the keyboard.

See also: *VnmrJ Imaging NMR*

Related: `eccTool` Pop-up ECC Tool window (M)
`eddyend` Update acquisition eddy current settings (M)

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: `nscans` Number of scout/real scan repetitions (P)
`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: *VnmrJ Liquids NMR*

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.
cyclenoef 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. Sequence is supplied with MERCURYplus/Vx as noedif.
Description: Sets up a difference NOE experiment.

cyclbr24 Set up parameters for cycled BR24 pulse sequence (M)
Applicability: Systems with solids module. Sequence is not supplied with MERCURY.
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)

cyclmrev 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: VnmrJ Liquids NMR
Related dli Display listed integral values (C)
dlini Display listed normalized integral values (C)
nli Find normalized integral values (C)
z Add integral reset point at the cursor position (C)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d0</td>
<td>Overhead delay between FIDs (P)</td>
</tr>
<tr>
<td>d1</td>
<td>First delay (P)</td>
</tr>
<tr>
<td>d2</td>
<td>Incremented delay in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>d2pul</td>
<td>Set up parameters for D2PUL pulse sequence (M)</td>
</tr>
<tr>
<td>d3</td>
<td>Incremented delay for 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>d4</td>
<td>Incremented delay for 3rd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>DAC_to_G</td>
<td>Store gradient calibration value in DOSY sequences (P)</td>
</tr>
<tr>
<td>da</td>
<td>Display acquisition parameter arrays (C)</td>
</tr>
<tr>
<td>daslp</td>
<td>Increment for t1 dependent first-order phase correction (P)</td>
</tr>
<tr>
<td>date</td>
<td>Date (P)</td>
</tr>
<tr>
<td>daxis</td>
<td>Display horizontal LC axis (M)</td>
</tr>
<tr>
<td>Dbppste</td>
<td>Set up parameters for Dbppste pulse sequence (M)</td>
</tr>
<tr>
<td>Dbppsteinept</td>
<td>Set up parameters for Dbppsteinept pulse sequence (M)</td>
</tr>
<tr>
<td>dbsetup</td>
<td>Set up VnmrJ database (U)</td>
</tr>
<tr>
<td>dbupdate</td>
<td>Update the VnmrJ database (U)</td>
</tr>
<tr>
<td>dc</td>
<td>Calculate spectral drift correction (C)</td>
</tr>
<tr>
<td>dc2d</td>
<td>Apply drift correction to 2D spectra (C)</td>
</tr>
<tr>
<td>dcg</td>
<td>Drift correction group (P)</td>
</tr>
<tr>
<td>dcon</td>
<td>Display noninteractive color intensity map (C)</td>
</tr>
<tr>
<td>dconi</td>
<td>Interactive 2D data display (C)</td>
</tr>
<tr>
<td>dconi</td>
<td>Control display selection for the dconi program (P)</td>
</tr>
<tr>
<td>dconn</td>
<td>Display color intensity map without screen erase (C)</td>
</tr>
<tr>
<td>dcrmv</td>
<td>Remove dc offsets from FIDs in special cases (P)</td>
</tr>
<tr>
<td>ddf</td>
<td>Display data file in current experiment (C)</td>
</tr>
<tr>
<td>ddfp</td>
<td>Display phase file in current experiment (C)</td>
</tr>
<tr>
<td>ddif</td>
<td>Synthesize and show DOSY plot (C)</td>
</tr>
<tr>
<td>dds</td>
<td>Default display (M)</td>
</tr>
<tr>
<td>dds_seqfil</td>
<td>Sequence-specific default display (M)</td>
</tr>
<tr>
<td>debug</td>
<td>Trace order of macro and command execution (C)</td>
</tr>
<tr>
<td>deccwarnings</td>
<td>Control reporting of DECC warnings from PSG (P)</td>
</tr>
<tr>
<td>decomp</td>
<td>Decompose a VXR-style directory (M)</td>
</tr>
<tr>
<td>def_osfilt</td>
<td>Default value of osfilt parameter (P)</td>
</tr>
<tr>
<td>defaultdir</td>
<td>Default directory for Files menu system (P)</td>
</tr>
<tr>
<td>delcom</td>
<td>Delete a user macro (M)</td>
</tr>
<tr>
<td>delete</td>
<td>Delete a file, parameter directory, or FID directory (C)</td>
</tr>
<tr>
<td>deleteSelected</td>
<td>Delete selected stack or slice (C)</td>
</tr>
<tr>
<td>deleteSlice</td>
<td>Delete selected slice (C)</td>
</tr>
<tr>
<td>delexp</td>
<td>Delete an experiment (M)</td>
</tr>
<tr>
<td>dels</td>
<td>Delete spectra from $T_1$ or $T_2$ analysis (C)</td>
</tr>
<tr>
<td>delta</td>
<td>Cursor difference in directly detected dimension (P)</td>
</tr>
<tr>
<td>delta1</td>
<td>Cursor difference in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>delta2</td>
<td>Cursor difference in 2nd indirectly detected dimension (P)</td>
</tr>
</tbody>
</table>
delaf
  Difference of two time-domain cursors (P)
dep
  Set up parameters for DEPT pulse sequence (M)
Dept
  Set up parameters for DEPT experiment (M)
DEPT
  Change parameters for DEPT experiment (M)
deptgl
  Set up parameters for DEPTGL pulse sequence (M)
deptrproc
  Process array of DEPT spectra (M)
destroy
  Destroy a parameter (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)
dfrq
  Transmitter frequency of first decoupler (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)
dgm
  Display menu to view parameter screens (C)
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)
dialog
  Display a dialog box from a macro (C)
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)
disCenterLines
  Show overlay as center lines (C)
disp3d
  Display 3D data (U)
display
  Display parameters and their attributes (C)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disStripes</td>
<td>Show overlay as stripes (C)</td>
</tr>
<tr>
<td>dla</td>
<td>Display spin simulation parameter arrays (M)</td>
</tr>
<tr>
<td>dlong</td>
<td>Long display of spin simulation parameter arrays (C)</td>
</tr>
<tr>
<td>dl1</td>
<td>Display list of integrals (C)</td>
</tr>
<tr>
<td>dlivast</td>
<td>Produce text file and process wells (M)</td>
</tr>
<tr>
<td>dll</td>
<td>Display listed line frequencies and intensities (C)</td>
</tr>
<tr>
<td>dlni</td>
<td>Display list of normalized integrals (M)</td>
</tr>
<tr>
<td>dlp</td>
<td>Decoupler low-power control with class C amplifier (P)</td>
</tr>
<tr>
<td>dm</td>
<td>Decoupler mode for first decoupler (P)</td>
</tr>
<tr>
<td>dm2</td>
<td>Decoupler mode for second decoupler (P)</td>
</tr>
<tr>
<td>dm3</td>
<td>Decoupler mode for third decoupler (P)</td>
</tr>
<tr>
<td>dm4</td>
<td>Decoupler mode for fourth decoupler (P)</td>
</tr>
<tr>
<td>dmf</td>
<td>Decoupler modulation frequency for first decoupler (P)</td>
</tr>
<tr>
<td>dmf2</td>
<td>Decoupler modulation frequency for second decoupler (P)</td>
</tr>
<tr>
<td>dmf3</td>
<td>Decoupler modulation frequency for third decoupler (P)</td>
</tr>
<tr>
<td>dmf4</td>
<td>Decoupler modulation frequency for fourth decoupler (P)</td>
</tr>
<tr>
<td>dmfadj</td>
<td>Adjust tip-angle resolution time for first decoupler (M)</td>
</tr>
<tr>
<td>dmf2adj</td>
<td>Adjust tip-angle resolution time for second decoupler (M)</td>
</tr>
<tr>
<td>dmf3adj</td>
<td>Adjust tip-angle resolution time for third decoupler (M)</td>
</tr>
<tr>
<td>dmf4adj</td>
<td>Adjust tip-angle resolution time for fourth decoupler (M)</td>
</tr>
<tr>
<td>dmg</td>
<td>Data display mode in directly detected dimension (P)</td>
</tr>
<tr>
<td>dmg1</td>
<td>Data display mode in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>dmg2</td>
<td>Data display mode in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>dmgf</td>
<td>Absolute-value display of FID data or spectrum in acqi (P)</td>
</tr>
<tr>
<td>dmi</td>
<td>Display multiple images (M)</td>
</tr>
<tr>
<td>dmm</td>
<td>Decoupler modulation mode for first decoupler (P)</td>
</tr>
<tr>
<td>dmm2</td>
<td>Decoupler modulation mode for second decoupler (P)</td>
</tr>
<tr>
<td>dmm3</td>
<td>Decoupler modulation mode for third decoupler (P)</td>
</tr>
<tr>
<td>dmm4</td>
<td>Decoupler modulation mode for fourth decoupler (P)</td>
</tr>
<tr>
<td>dn</td>
<td>Nucleus for first decoupler (P)</td>
</tr>
<tr>
<td>dn2</td>
<td>Nucleus for second decoupler (P)</td>
</tr>
<tr>
<td>dn3</td>
<td>Nucleus for third decoupler (P)</td>
</tr>
<tr>
<td>dn4</td>
<td>Nucleus for fourth decoupler (P)</td>
</tr>
<tr>
<td>dnode</td>
<td>Display list of valid limNET nodes (M,U)</td>
</tr>
<tr>
<td>doautodial</td>
<td>Start a dialog window using def file (M)</td>
</tr>
<tr>
<td>dodi</td>
<td>Start a dialog window with dialoglib file (M)</td>
</tr>
<tr>
<td>dof</td>
<td>Frequency offset for first decoupler (P)</td>
</tr>
<tr>
<td>dof2</td>
<td>Frequency offset for second decoupler (P)</td>
</tr>
<tr>
<td>dof3</td>
<td>Frequency offset for third decoupler (P)</td>
</tr>
<tr>
<td>dof4</td>
<td>Frequency offset for fourth decoupler (P)</td>
</tr>
<tr>
<td>Doneshot</td>
<td>Set up parameters for Doneshot pulse sequence (M)</td>
</tr>
<tr>
<td>dopardialog</td>
<td>Start a dialog with dialoglib/experiment def file (M)</td>
</tr>
<tr>
<td>dopcss</td>
<td>Calculate proton chemical shifts spectrum (C)</td>
</tr>
<tr>
<td>dosy</td>
<td>Process DOSY experiments (M)</td>
</tr>
<tr>
<td>dosyfrq</td>
<td>Larmor frequency of phase encoded nucleus in DOSY (P)</td>
</tr>
<tr>
<td>dosygamma</td>
<td>Gyromagnetic constant of phase encoded nucleus in DOSY (P)</td>
</tr>
<tr>
<td>dosytimecubed</td>
<td>Gyromagnetic constant of phase encoded nucleus in DOSY (P)</td>
</tr>
</tbody>
</table>
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)
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)
dpwr2f  First decoupler fine power (P)
dpwr2f2  Second decoupler fine power (P)
dpwr3f  Third decoupler fine power (P)
dpwr3f2  Second decoupler linear modulator power (P)
dpwr4f  Third decoupler linear modulator power (P)
dqcosy  Set up parameters for double-quantum filtered COSY (M)
Dqcosy  Convert the parameter to a DQCOSY experiment (M)
DQCOSY  Change parameters for DQCOSY experiment (M)
draw  Draw line from current location to another location (C)
drawslice  Display target slices (M)
drawvox  Display target voxels (M)
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)
dscoef  Display scale below spectrum or FID (C)
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)
**d0**  
**Overhead delay between FIDs (P)**

**Applicability:**  
UNITY/INOVA systems

**Description:**  
Defines the extra overhead delay at the start of each FID or array element. Overhead times between increments and transients on the UNITY/INOVA 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 \( d_0 = 'n' \), 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 \( d_0 \) can be viewed by entering \( d_0 = 'y' \) in the input window.

If \( d_0 \) 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 \( d_0 \).

See also: User Programming

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

\textbf{d1} \\
**First delay (P)**

\textbf{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.

\textbf{Values:} On \textit{MERCURYplus/Vx}: 0, 0.2 \( \mu \text{s} \) to 150,000 sec.  
On \textit{INOV A}: 0.1 \( \mu \text{s} \) to 8190 sec, smallest value possible is 0.1 \( \mu \text{s} \), finest increment possible is 12.5 ns.

See also: \textit{VnmrJ Liquids NMR}

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)  

d2pul \\
**Set up parameters for D2PUL pulse sequence (M)**

\textbf{Applicability:} D2PUL is not available on \textit{MERCURYplus/Vx} systems.

\textbf{Description:} Sets up a standard two-pulse sequence using the decoupler as transmitter.

See also: \textit{VnmrJ Liquids NMR}

Related: dhp Decoupler high power with class C amplifier (P)  
dx Nucleus for the first decoupler (P)  
dof Frequency offset for first decoupler (P)  
dpwr Power level for first decoupler with linear amplifiers (P)  
homo Homodecoupling control for first decoupler (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)  

Related:

\textbf{Related:} create Create new parameter in parameter tree (C)
d3 **Incremented delay for 2nd indirectly detected dimension (P)**

**Description:** Length of a delay controlled by the parameters \( n_2 \) and \( s_2 \) in a 3D experiment. The \( d_2 \) delay, which is controlled by \( n_1 \) and \( s_1 \), is incremented through its entire implicit array first before \( d_3 \) is incremented. To create parameters \( d_3, n_2, p_2, \) and \( s_2 \) to acquire a 3D data set in the current experiment, enter `addpar ('3d')`.

**Values:** On MERCURYplus/Vx: 0, 0.2 \( \mu \)s to 150,000 sec. On INOVA: 0.1 \( \mu \)s to 8190 sec, smallest value possible is 0.1 \( \mu \)s, finest increment possible is 12.5 ns.

See also: *VnmrJ Liquids NMR*

**Related:**
- `addpar`: Add selected parameters to the current experiment (M)
- `d1`: First delay (P)
- `n2`: Number of increments in 2nd indirectly detected dimension (P)
- `par3d`: Create 3D acquisition, processing, display parameters (C)
- `p2`: Phase selection for 3D acquisition (P)
- `s2`: 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 \( n_3 \) and \( s_3 \) in a 4D experiment. The \( d_3 \) delay, which is controlled by \( n_2 \) and \( s_2 \), is incremented through its entire implicit array first before \( d_4 \) is incremented. To create parameters \( d_4, n_3, p_3, \) and \( s_3 \) to acquire a 4D data set in the current experiment, enter `addpar ('4d')`.

**Values:** On MERCURYplus/Vx: 0, 0.2 \( \mu \)s to 150,000 sec. On INOVA: 0.1 \( \mu \)s to 8190 sec, smallest value possible is 0.1 \( \mu \)s, finest increment possible is 12.5 ns.

See also: *VnmrJ Liquids NMR*

**Related:**
- `addpar`: Add selected parameters to the current experiment (M)
- `d1`: First delay (P)
- `n3`: Number of increments in 3rd indirectly detected dimension (P)
- `par4d`: Create 4D acquisition parameters (C)
- `p3`: Phase selection for 4D acquisition (P)
- `s3`: Spectral width in 3rd indirectly detected dimension (P)

---

**DAC_to_G** **Store gradient calibration value in DOSY sequences (P)**

**Description:** `DAC_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: *VnmrJ Liquids NMR.*

**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:** \( \text{par1}, \text{par2}, \text{par3}, \ldots \) are names of parameters to be displayed. The default is to display all such parameters.
Examples:

```
da
nda(‘d2’)
```

See also: *VnmrJ Liquids NMR*

Related: *dg*  Display parameters of acquisition/processing group (C)

**daslp**

**Increment for t1 dependent first-order phase correction (P)**

Applicability: 

UNITY/INOVA systems.

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: *VnmrJ Liquids NMR; User Guide: Solid-State NMR*

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 $f_2$ 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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

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:**  
*VnmrJ Liquids NMR*

**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:**  
*VnmrJ Liquids NMR*

**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:**  
*VnmrJ Liquids NMR*  
*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: VnmrJ Liquids NMR

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 f1 axis direction. 'f2' is a keyword to apply drift correction in the f2 axis direction.

Examples:
dc2d('f1')
dc2d('f2')

See also: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related:
cdc  Cancel drift correction (C)
dc  Calculate spectral drift correction (C)
**dcon**

**Display noninteractive color intensity map (C)**

**Syntax:**

```plaintext
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:**

```plaintext
dcon

dcon('gray')

dcon('linear','phcolor','plot')
```

**See also:**

*VnmrJ Liquids NMR*

**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)
- `image` Display noninteractive gray scale image (M)
- `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)
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: *VnmrJ Liquids NMR*

**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)
deltal Cursor difference in 1st indirectly detected dimension (P)
df2d Display FIDs of 2D experiment (C)
dpcon Display plotted contours (C)
da2d 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)
image Display noninteractive gray scale image (M)
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)
sp2 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, dconn, da2d, or da2d. 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: VnmrJ Liquids NMR; 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)

donnn

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR; 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: VnmrJ Liquids NMR

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)

**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 debug(‘c’) output with a particular terminal, enter tty. The system responds with `/dev/pts/yyyy`, where `yyyy` is a numerical value. On the VnmrJ command line, enter jFunc(55, `' /dev/pts/yyyy'`), substituting the numerical value for the `yyyy`.

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
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 grandien 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.

decom 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: VnmrJ Liquids NMR

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, AnalogPlus™ 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 AnalogPlus 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: VnmrJ Liquids NMR

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: *VnmrJ Liquids NMR*

**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 or a macro in a macro directory specified by the maclibpath parameter.

**Arguments:** file is the file name of the user’s macro to be deleted.

**Examples:** delcom('lds')

See also: *User Programming*

**Related:**
- *crcm* Create user macro without using a text editor (C)
- *maclibpath* Path to user’s macro directory (P)
- *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: *VnmrJ Liquids NMR*

**Related:**
- *delexp* Delete an experiment (M)
- *rm* Delete file (C)
- *rmdir* Remove directory (C)

**deleteSelected**

**Delete selected stack or slice (C)**

**Applicability:** Systems with imaging capabilities.

**Description:** Deletes selected stack or slice (only one is selected at a time).

**Related:** *gplan* Start interactive image planning (C)
**deleteSlice**  Delete selected slice (C)

Applicability: Systems with imaging capabilities.

Description: Deletes selected slice.

Related: gplan  Start interactive image planning (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 jexpXXX macro if necessary.

Examples: delexp(321)

See also: VnmrJ Liquids NMR

Related: cexp  Create an experiment (M)

jexp  Join existing experiment (C)

**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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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.
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: *VnmrJ Liquids NMR*

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, vpf, and crf (if the parameter set is older and lacks these parameters), enter addpar('fid').

Values: Number, in seconds.

See also: *VnmrJ Liquids NMR*

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 pulse sequence (M)**

Description: Macro for the DEPT (Distortionless Enhancement by Polarization Transfer) experiment.

See also: *VnmrJ Liquids NMR*

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)
deptrc proc Process array of DEPT spectra (M)
padept Plot automatic DEPT analysis (C)
ppcal Proton decoupler pulse calibration (M)

**Dept** **Set up parameters for DEPT experiment (M)**

Description: Set up parameters for DEPT experiment

**DEPT** **Change parameters for DEPT experiment (M)**

Description: Converts the current parameter set to a DEPT experiment.

**deptgl** **Set up parameters for DEPTGL pulse sequence (M)**

Applicability: Sequence is not supplied with *MERCUryplus/Vx*.

Description: Macro for the DEPTGL pulse sequence for spectral editing and polarization transfer experiments.
See also: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: `adept` Automatically edit DEPT spectra (C)

### destroy

**Destroy a parameter (C)**

**Syntax:** `destroy(parameter<,tree>)`

**Description:** Removes a parameter from one of the parameter trees. If the destroyed parameter was an array, the `array` parameter 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:**
- `parameter` is the name of the parameter to be destroyed.
- `tree` is a keyword for the type of parameter tree: 'global', 'current', 'processed', or 'systemglobal'. The default is 'current'. Refer to the `create` command for more information on types of trees.

**Examples:**
- `destroy('a')`
- `destroy('c','global')`

See also: *User Programming*

**Related:**
- `array` Parameter order and precedence (P)
- `create` Create new parameter in a parameter tree (C)
- `display` Display parameters and their attributes (C)
- `paramvi` Edit a variable and its attributes using vi text editor (C)
- `prune` Prune extra parameters 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)
**df**

**Display a single FID (C)**

**Syntax:**
1. `df<(index)>`
2. `df(options)`

**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.

**Examples:**
- `df`
- `df(4)`
- `df('restart')`

**See also:**
*VnmrJ Liquids NMR*

**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)`
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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: dn4  Nucleus for fourth decoupler (P)  

dof4  Frequency offset for fourth decoupler (P)  

cspfrq  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: *VnmrJ Liquids NMR*

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)
**dfs**  
**Display stacked FIDs automatically (C)**

Syntax: `dfs{<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: *VnmrJ Liquids NMR*

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 **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: *VnmrJ Liquids NMR*

Related:  
- **dfs** Display stacked FIDs (C)
- **dfsan** Display stacked FIDs automatically without screen erase (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: 
VnmrJ Liquids NMR

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 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: 
VnmrJ Liquids NMR
Related: 
dfs 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: 
VnmrJ Liquids NMR
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: 
VnmrJ Liquids NMR
Related: 
dfs Display stacked FIDs (C)
pfww Plot FIDs in whitewash mode (C)

dg Display group of acquisition/processing parameters (C)
Syntax: dg({template})>
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. Commands such as dg1, dg2, and dgs (but not da) are macros that activate dg with the appropriate template argument ('dg1', 'dg2', 'dgs', etc.).

Examples:  
dg  
dg('dgexp')

See also:  VnmrJ Liquids NMR; User Programming

Related:
- ? Display individual parameter value (C)  
- da Display acquisition parameter arrays (C)  
- dg Control dg parameter group display (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)  
- da Display acquisition parameter arrays (C)

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:  VnmrJ Liquids NMR

Related:
- ? Display individual parameter value (C)  
- da Display acquisition parameter arrays (C)  
- dg Control dg parameter group display (P)  
- dg1 Display group of display parameters (M)  
- dgs Display group of special/automation parameters (M)  
- da Display acquisition parameter arrays (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:  VnmrJ Liquids NMR

Related:
- ? Display individual parameter value (C)  
- da Display acquisition parameter arrays (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:  VnmrJ Liquids NMR

Related:
- ? Display individual parameter value (C)  
- da Display acquisition parameter arrays (C)  
- dg1 Control dg1 parameter group display (P)  
- dg Display group of acquisition/processing parameters (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.
D

See also: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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:** *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**Related:**  
dglc  
Display LC-NMR parameters (M)
parlc  
Create LC-NMR parameters (M)

**dgm**  
**Display menu to view parameter screens (C)**

**Applicability:** Systems with imaging capabilities.

**Description:** Displays a menu for selecting and viewing a list of parameter screens.

**See also:** *VnmrJ Imaging NMR*

**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:** *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**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: VnmrJ Liquids NMR

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)

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)

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/changelto3
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 sv command.
See also:  VnmrJ Liquids NMR
Related:  sv  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:  VnmrJ Liquids NMR
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:  VnmrJ Liquids NMR
Related:  lf  List files in directory (C)
          ls  List files in directory (C)

disCenterLines  Show overlay as center lines (C)
Applicability:  Systems with imaging capabilities.
Description:  Shows intersection overlay of stack as center lines or stripes.
See also:  VnmrJ Liquids NMR
Related:  gplan  Start interactive image planning (C)
D

**disp3d**  
**Display 3D data (U)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** (From UNIX) `disp3d <fdf_file>`

**Description:** Displays a 3D FDF (Flexible Data Format) file or a raw 8-bit 3D data file with no header. Compatible FDF files are produced by `ft3d` with the 'fdf' option (or by default if `appmode='imaging'`).

FDF data can also be loaded either by entering the file name as an argument to `disp3d` or by typing the file name into the File field in the `disp3d` control panel and clicking the Load button. If the FDF data word size is larger than 8 bits, the data are scaled and truncated to 8 bits for display. Raw data files can only be loaded from the control panel.

Besides the file name, the user must enter the size of the data matrix in the fast, medium, and slow dimensions in the Data size field. Typically, these would be the values \(fn/2\), \(fn1/2\), and \(fn2/2\), respectively.

Furthermore, the desired size of the image in screen pixels—also in the fast, medium, and slow dimensions—must be entered in the Display size fields. Typically, these values would be near 100 and the relative ratio of the parameters \(lro\), \(lpe\), and \(lpe2\), respectively.

After loading the data, a 3D volume appears in the display panel.

**Arguments:** `fdf_file` is the name of a file containing FDF data.

**See also:** *VnmrJ Imaging NMR*

**Related:**  
- `appmode` Application mode (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 on a 3D FID data set (M,U)  
- `lpe` Field of view size for phase encode axis (P)  
- `lpe2` Field of view size for 2nd phase-encode axis (P)  
- `lro` Field of view size for readout axis (P)

**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)

**disStripes**  
**Show overlay as stripes (C)**

Applicability: Systems with imaging capabilities.

Description: Shows intersection overlay of stack as stripes.

See also: *VnmrJ Liquids NMR*

Related: **gplan** Start interactive image planning (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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: **dla** Display spin simulation parameter arrays (M)

**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 \texttt{ins} or by \texttt{insref}. When the integral blanking mode is used (i.e., \texttt{intmod='partial' }), only the integrals corresponding to the displayed integral regions are listed.

The displayed integral value can be scaled with the \texttt{setint} macro. The integral is scaled by the parameters \texttt{ins} and \texttt{insref}.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \texttt{axis}  
  Axis label for displays and plots (P)
- \texttt{cz}  
  Clear integral reset points (C)
- \texttt{dlni}  
  Display list of normalized integrals (M)
- \texttt{ins}  
  Integral normalization scale (P)
- \texttt{insref}  
  Fourier number scaled value of an integral (P)
- \texttt{liamp}  
  Amplitudes of integral reset points (P)
- \texttt{lifrq}  
  Frequencies of integral reset points (P)
- \texttt{nli}  
  Find integral values (C)
- \texttt{rfl}  
  Reference peak position in directly detected dimension (P)
- \texttt{rfp}  
  Reference peak frequency in directly detected dimension (P)
- \texttt{setint}  
  Set value of an integral (M)
- \texttt{z}  
  Add integral reset point at cursor position (C)

\texttt{dlivast}  
Produce text file and process wells (M)

Applicability:  
VAST accessory.

Syntax:  
\texttt{dlivast\langle\texttt{last}\rangle}\)

Description:  
Produces a text file containing the integral of the partial regions and processes the wells.

Arguments:  
\texttt{last} is the number of the last well. The default is 96.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \texttt{combiplate}  
  View a color map for visual analysis of VAST microtiter plate (U)
- \texttt{combishow}  
  Display regions as red, green, and blue in CombiPlate window (M)

\texttt{dll}  
Display listed line frequencies and intensities (C)

Syntax:  
\texttt{dll\langle\texttt{,'pos'\langle,noise\_mult\rangle}\rangle\langle:\texttt{number\_lines},\texttt{scale}\rangle}\)

Description:  
Displays a list of line frequencies and amplitudes that are above a threshold defined by \texttt{th}. Frequency units are defined by the parameter \texttt{axis}. The results of this calculation are stored in \texttt{lifrq} and \texttt{liamp}. The frequencies are stored as Hz and are not referenced to \texttt{rfl} and \texttt{rfp}. Amplitudes are stored as the actual data point value; they are not scaled by \texttt{vs}.

Arguments:  
\texttt{,'pos'} is a keyword to list only positive lines.

\texttt{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 \texttt{th}. Negative values of \texttt{noise\_mult} are changed to 3.

\texttt{number\_lines} is a return argument with the number of lines above the threshold.

\texttt{scale} is a return argument with a scaling factor for line amplitudes. This scaling factor accounts for \texttt{vs} and whether the lines are listed in absolute intensity mode or normalized mode.

Examples:  
\texttt{dll}\)
\texttt{dll\langle\texttt{,'pos'}\rangle}
dl1 (2.5)
dll: r1, sc

See also: VnmrJ Liquids NMR

Related: axis  Axis label for displays and plots (P)
dels  Delete spectra from T1 or T2 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)
rfi  Reference peak position in directly detected dimension (P)
rfp  Reference peak frequency in directly detected dimension (P)
ths  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: VnmrJ Liquids NMR

cz  Clear integral reset points (C)
dli  Display list of integrals (C)
is  Integral normalization scale (P)
lli  Find integral values (C)
z  Add integral reset point at cursor position (C)

dl1p  Decoupler low-power control with class C amplifier (P)

Applicability: Systems with a class C amplifier.

Description: dl1p controls the decoupler power level for systems with a class C decoupler amplifier in the low-power mode, generally used for homonuclear decoupling. dl1p specifies dB of attenuation of the decoupler, below a nominal 1 watt value. dl1p is active only if dhp = 'n'. On systems with a decoupler linear amplifier, dl1p is nonfunctional and dpwr controls decoupler power.

Values: 0 to 39 (in dB of attenuation, 0 is maximum power).

See also: VnmrJ Liquids NMR

Related: dhp  Decoupler high-power control with class C amplifier (P)
dm  Decoupler mode for first decoupler (P)
dnf  Decoupler modulation frequency for first decoupler (P)
dpwr  Power level for first decoupler with linear amplifier (P)

dm  Decoupler mode for first decoupler (P)

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 = 'ynyr' 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.

'a' specifies the asynchronous mode, the same as 'y'. The 'a' value is not available on MERCURYplus/Vx systems.

's' specifies the synchronous mode in which the decoupler rf is gated on and modulation is started at the beginning of the modulation sequence. This value has meaning only on UNITY INOVA systems.

See also: VnmrJ Liquids NMR

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)

**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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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)

**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 \texttt{dm}.

Values: Same as \texttt{dm}, except that if \texttt{dn4=''} (two single quotes with no space in between) and a fourth decoupler is present in the console, \texttt{dm4} assumes a default value of 'n' when \texttt{go} is executed.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{dm} Decoupler mode of first decoupler (P)  
\texttt{dm4} Decoupler modulation frequency for fourth decoupler (P)  
\texttt{dm4} Decoupler modulation mode for fourth decoupler (P)  
\texttt{dn4} Nucleus for fourth decoupler (P)

\texttt{dmf} \hfill \textbf{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 \(\gamma H_2\) (expressed in units of Hz), \texttt{dmf} should be set equal to \(4*\gamma H_2\) for WALTZ, MLEV16, GARP, and XY32 (when available).

\texttt{dmf} is inactive for CW mode decoupling (\texttt{dmm='c'}).  
\texttt{dmf} is also active for square wave mode decoupling (\texttt{dmm='s'}) and fm-fm mode (\texttt{dmm='f'}) decoupling. For \texttt{dmm='f'}, the modulation frequency is swept back and forth between about 0.5% and 5% of the \texttt{dmf} frequency (e.g., if \texttt{dmf} is 100 kHz, the modulation is swept between approximately 500 Hz and 5 kHz). A reasonable optimum value for \texttt{dmf} when \texttt{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 \texttt{dmf} value is internally multiplied by 45, making the limit of possible \texttt{dmf} values to 5 Hz to 44.4 kHz when \texttt{dmm='g'}.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{dmf2} Decoupler modulation frequency for second decoupler (P)  
\texttt{dmf3} Decoupler modulation frequency for third decoupler (P)  
\texttt{dmf4} Decoupler modulation frequency for fourth decoupler (P)  
\texttt{dmm} Decoupler modulation mode for first decoupler (P)  
\texttt{pw90} 90° pulse width (P)

\texttt{dmf2} \hfill \textbf{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 \texttt{dmf}.

Values: Same as \texttt{dmf} except that if \texttt{dn2=''} (two single quotes with no space in between) and a second decoupler is present in the console (\texttt{numrfch} greater than 2), \texttt{dmf2} assumes a default value of 1000 Hz when \texttt{go} is executed.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{dm2} Decoupler mode for second channel (P)  
\texttt{dmf} Decoupler modulation frequency for first decoupler (P)  
\texttt{dmm2} Decoupler modulation mode for second decoupler (P)  
\texttt{dn2} Nucleus for second decoupler (P)  
\texttt{numrfch} Number of rf channels (P)

\texttt{dmf3} \hfill \textbf{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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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**

Adjust tip-angle resolution time for first decoupler (M)

Applicability: All systems except MERCURYplus/Vx.

Syntax: dmfadj<(tipangle_resolution)>

Description: Adjusts the parameter dmf so that time associated with the first decoupler tip-angle resolution is an integral multiple of 50 ns. This eliminates time truncation error in execution of programmable decoupling or spin-locking sequence by the waveform generator. For example, the tip-angle resolution for an MLEV-16 decoupling sequence should be 90.0° since every pulse in that sequence can be represented as an integral multiple of 90.0°; however, the tip-angle resolution for a GARP decoupling sequence should be 1.0°.

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 dres.

Examples:
```
dmfadj
```
```
dmfadj (90.0)
```

See also: VnmrJ Liquids NMR

Related: dmf Decoupler modulation frequency for first decoupler (P)
dmf2adj Adjust tip-angle resolution time for second decoupler (M)
dmf3adj Adjust tip-angle resolution time third decoupler (M)
dmf4adj Adjust tip-angle resolution time fourth decoupler (M)
dres Tip angle resolution for programmable decoupling (P)
**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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 (`UNITY/INOVA`). `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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: 
- **aig** Absolute intensity group (P)
- **av** Set absolute-value mode in directly detected dimension (C)
- **dcg** Drift correction group (P)
- **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)
- **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 `phl`, `avl`, `pwr1`, or `pal` for the values 'phl', 'avl', 'pwr1', or 'pal', 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: 'phl' sets phased mode.

'avl' sets absolute-value mode.

'pwr1' sets power mode.

'pal' sets phase angle mode.

See also: *VnmrJ Liquids NMR*

Related: 
- **avl** Set absolute-value mode in 1st indirectly det. dim. (C)
- **dmg** Data display mode in directly detected dimension (P)
- **pal** Set phase angle mode in 1st indirectly detected dimension (C)
- **phl** 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)**

**Applicability:** All systems except MERCURYplus/Vx.

**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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**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)

dmi  **Display multiple images (M)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** dmi

**Description:** Displays a series of multiple images from a single arrayed and/or multislice/multiecho experiment in the graphics window. The resulting display is noninteractive. The layout and size of the images are optimized to maximize the image display size.

**See also:** VnmrJ Imaging NMR

**Related:**
- svib  Generate and save images as ImageBrowser FDF files (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: On UNITY/INOVA, 'c', 'f', 'g', 'm', 'p', 'r', 'u', 'w', and 'x' are available; on MERCURYplus/Vx 'c', 'f', 'g', 'm', 'r', '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.
- 'p' sets noise modulation.
- 'r' sets programmable pulse modulation using the dseq parameter to specify the decoupling sequence.
- 'u' sets square-wave modulation.
- 'u' sets user-supplied modulation using external hardware.
- 'w' sets WALTZ-16 modulation.
- 'x' sets XY32 modulation.

See also: VnmrJ Liquids NMR

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: For UNITY/INOVA '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: VnmrJ Liquids NMR

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: For \textit{UNITY INOVA}, \textit{c'}, \textit{f'}, \textit{g'}, \textit{m'}, \textit{p'}, \textit{r'}, \textit{u'}, \textit{w'}, and \textit{x'} are available. Refer to \texttt{dmm} for the definition of these values (note that if the mode \textit{p'} is selected, \texttt{dseq3} specifies the decoupling sequence). If \texttt{dn3} = '' (two single quotes) and a third decoupler is present in the console (\texttt{numrfch} equal to 4), \texttt{dmm3} is internally set to \textit{c'} when \texttt{go} is executed.

See also: \textit{VnmrJ Liquids NMR}

Related: 
\begin{itemize}
  \item \texttt{dm3} Decoupler modulation for third decoupler (P)
  \item \texttt{dmf3} Decoupler modulation frequency for third decoupler (P)
  \item \texttt{dmm} Decoupler modulation mode for first decoupler (P)
  \item \texttt{dn3} Nucleus for the third decoupler (P)
  \item \texttt{dseq3} Decoupler sequence for the third decoupler (P)
  \item \texttt{numrfch} Number of rf channels (P)
\end{itemize}

\textbf{dmm4}

\textbf{Decoupler modulation mode for fourth decoupler (P)}

\textbf{Applicability}: Systems with a deuterium decoupler channel as the fourth decoupler.

\textbf{Description}: Sets type of decoupler modulation for the fourth decoupler during different status periods within a pulse sequence. It functions analogously to \texttt{dmm}.

Values: For \textit{UNITY INOVA}, \textit{c'}, \textit{f'}, \textit{g'}, \textit{m'}, \textit{r'}, \textit{u'}, \textit{w'}, and \textit{x'} are available. Refer to \texttt{dmm} for the definition of these values. If \texttt{dn4} = '' (two single quotes) and a fourth decoupler is present in the console (\texttt{numrfch} greater than 4), \texttt{dmm4} is internally set to \textit{c'} when \texttt{go} is executed.

See also: \textit{VnmrJ Liquids NMR}

Related: 
\begin{itemize}
  \item \texttt{dm4} Decoupler modulation for fourth decoupler (P)
  \item \texttt{dmf4} Decoupler modulation frequency for fourth decoupler (P)
  \item \texttt{dmm} Decoupler modulation mode for first decoupler (P)
  \item \texttt{dn4} Nucleus for the fourth decoupler (P)
  \item \texttt{dseq4} Decoupler sequence for the fourth decoupler (P)
  \item \texttt{numrfch} Number of rf channels (P)
\end{itemize}

\textbf{dn}

\textbf{Nucleus for first decoupler (P)}

\textbf{Description}: Changing the value of \texttt{dn} causes a macro (named \_\texttt{dn}) to be executed that extracts values for \texttt{dfrq} and \texttt{dof} from lookup tables. The tables, stored in the directory /vnmr/nuctables, are coded by atomic weights.

Values: In the lookup tables, typically \textit{'H1'}, \textit{'C13'}, \textit{'P31'}, etc.

See also: \textit{VnmrJ Liquids NMR}

Related: 
\begin{itemize}
  \item \texttt{dfrq} Transmitter frequency of first decoupler (P)
  \item \texttt{dn2} Nucleus for second decoupler (P)
  \item \texttt{dn3} Nucleus for third decoupler (P)
  \item \texttt{dn4} Nucleus for fourth decoupler (P)
  \item \texttt{dof} Frequency offset for first decoupler (C)
  \item \texttt{tn} Nucleus for observe transmitter (P)
\end{itemize}

\textbf{dn2}

\textbf{Nucleus for second decoupler (P)}

\textbf{Applicability}: Systems with a second decoupler.

\textbf{Description}: Changing the value of \texttt{dn2} causes a macro (named \_\texttt{dn2}) to be executed that extracts values for \texttt{dfrq2} and \texttt{dof2} from lookup tables. Otherwise, \texttt{dn2} functions analogously to the parameters \texttt{tn} and \texttt{dn}. If an experiment does not use the second decoupler channel, the channel can be disabled by setting \texttt{dn2} = '' (two single quotes with no space in between). This sets \texttt{dm2} = \textit{n'},
### D

\[
\begin{aligned}
dmm2 &= 'c', 
\text{dmf2} &= 1000 \text{ (in Hz)}, 
\text{dfrq2} &= 1 \text{ (in MHz)}, 
\text{dof2} &= 0, 
\text{dpwr2} &= 0, 
\text{homo2} &= 'n', 
\text{dseq2} &= '', \text{ and } 
\text{dres2} &= 1.
\end{aligned}
\]

See also: *VnmrJ Liquids NMR*

**Related:**
- \text{dfrq2}  
  Transmitter frequency of second decoupler (P)
- \text{dn}  
  Nucleus for first decoupler (P)
- \text{dof2}  
  Frequency offset for second decoupler (C)
- \text{numrfch}  
  Number of rf channels (P)
- \text{tn}  
  Nucleus for observe transmitter (P)

#### \text{dn3}

**Nucleus for third decoupler (P)**

**Applicability:** Systems with a third decoupler.

**Description:** Changing the value of \text{dn3} causes a macro (named \_\text{dn3}) to be executed that extracts values for \text{dfrq3} and \text{dof3} from lookup tables. Otherwise, \text{dn3} functions analogously to the parameters \text{tn} and \text{dn}. If an experiment does not use the third decoupler channel, the channel can be disabled by setting \text{dn3} = '' (two single quotes with no space in between). This sets \text{dm3} = 'n', \text{dmm3} = 'c', 
\text{dmf3} = 1000 \text{ (in Hz)}, 
\text{dfrq3} = 1 \text{ (in MHz)}, 
\text{dof3} = 0, 
\text{dpwr3} = 0, 
\text{homo3} = 'n', 
\text{dseq3} = '', \text{ and } 
\text{dres3} = 1.

See also: *VnmrJ Liquids NMR*

**Related:**
- \text{dn}  
  Nucleus for first decoupler (P)
- \text{dfrq3}  
  Transmitter frequency of third decoupler (P)
- \text{dof3}  
  Frequency offset for third decoupler (C)
- \text{numrfch}  
  Number of rf channels (P)
- \text{tn}  
  Nucleus for observe transmitter (P)

#### \text{dn4}

**Nucleus for fourth decoupler (P)**

**Applicability:** Systems with a deuterium decoupler channel as the fourth decoupler.

**Description:** Changing the value of \text{dn4} causes a macro (named \_\text{dn4}) to be executed that extracts values for \text{dfrq4} and \text{dof4} from lookup tables. Otherwise, \text{dn4} functions analogously to the parameters \text{tn} and \text{dn} except that the only valid value for \text{dn4} is 'H2'. If an experiment does not use the fourth decoupler channel, the channel can be disabled by setting \text{dn4} = '' (two single quotes with no space in between). This sets \text{dm4} = 'n', \text{dmm4} = 'c', 
\text{dmf4} = 1000 \text{ (in Hz)}, 
\text{dfrq4} = 1 \text{ (in MHz)}, 
\text{dof4} = 0, 
\text{dpwr4} = 0, 
\text{homo4} = 'n', 
\text{dseq4} = '', \text{ and } 
\text{dres4} = 1.

See also: *VnmrJ Liquids NMR*

**Related:**
- \text{dfrq4}  
  Transmitter frequency of fourth decoupler (P)
- \text{dn}  
  Nucleus for first decoupler (P)
- \text{dof4}  
  Frequency offset for fourth decoupler (C)
- \text{numrfch}  
  Number of rf channels (P)
- \text{tn}  
  Nucleus for observe transmitter (P)

#### 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: *VnmrJ Liquids NMR*

**Related:**
- \text{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 generally 0.1 Hz. The value is specified in the `config` program.
Description: $-100000$ to $100000$ Hz (approximate, depends on frequency), in steps of 0.1 Hz.
See also: *VnmrJ Liquids NMR*
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: *VnmrJ Liquids NMR*
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.
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: *VnmrJ Liquids NMR*

Related: ddif Synthesize and display DOSY plot (C)
fiddle Perform reference deconvolution (M)
setup_dosy Set up gradient levels for DOSY experiments (M)

**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: dosy Process DOSY experiments (M)

**dosytimecubed**  
Gyromagnetic constant of phase encoded nucleus in DOSY (P)

Description: Timecubed factor in the expression for diffusional attenuation. It is automatically set by the DOSY sequences and used by the dosy macro.

See also: *VnmrJ Liquids NMR*

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 $90^\circ$ 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:  
*VnmrJ Liquids NMR*

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)
- $pw90$: $90^\circ$ 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:  
*VnmrJ Liquids NMR*

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)

**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 $downsamp$, $dscoef$, $dsfb$, $dsisfreq$, and $filtfile$. 


**Values**: Number for the downsampling factor. 1 sets digital filtering with a filter bandwidth specified by \( \text{dsfb} \) without downsampling. 

'\( n \)' sets normal data processing without digital filtering.

**See also**: *VnmrJ Liquids NMR*

**Related**:
- \( \text{addpar} \): Add selected parameters to current experiment (M)
- \( \text{digfilt} \): Write digitally filtered FID to another experiment (M)
- \( \text{dscoef} \): Digital filter coefficients for downsampling (P)
- \( \text{dsfb} \): Digital filter bandwidth for downsampling (P)
- \( \text{dslafrq} \): Bandpass filter offset for downsampling (P)
- \( \text{filtfile} \): File of FIR digital filter coefficients (P)
- \( \text{pards} \): Create additional parameters used by downsampling (M)
- \( \text{sw} \): 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 CONFIG window), \( \text{dp} \) is forced to 'n' if \( 120000 < \text{sw} <= 200000 \). If \( \text{sw} > 200000 \), \( \text{dp} \) is forced to 'y'. On wideline systems, \( \text{dp} = 'y' \) is required when \( \text{sw} > 100000 \). On *MERCURYplus/Vx* \( \text{dp} = 'y' \) only.

**See also**: *VnmrJ Liquids NMR*

**Related**: \( \text{sw} \): Spectral width in directly detected dimension (P)

**dpcon**

**Display plotted contours (C)**

**Syntax**: \( \text{dpcon(<options,>><levels,spacing>)} \)

**Description**: Produces a true contour plot display.

**Arguments**: \( \text{options} \) must precede \( \text{levels} \) and \( \text{spacing} \) 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.

\( \text{levels} \) is the maximum number of contours to be shown. The default is 4.

\( \text{spacing} \) is the spacing by relative intensity of successive contour levels. The default is 2.

**Examples**: 
- \( \text{dpcon} \)
- \( \text{dpcon('pos',6)} \)
- \( \text{dpcon(15,1.4)} \)

**See also**: *VnmrJ Liquids NMR*

**Related**: 
- \( \text{dcon} \): Display noninteractive color intensity map (C)
- \( \text{dconi} \): Control display selection for the \( \text{dconi} \) program (P)
- \( \text{dpconn} \): Display plotted contours without screen erase (C)
- \( \text{pcon} \): 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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
Related: axis Axis label for displays and plots (P)  
dpir Display integral amplitudes below spectrum (C)  
dpirm Display normalized integral amplitudes below spectrum (M)  
pir Plot integral amplitudes below spectrum (C)  
pirm 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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related:

- **dpir**  Display integral amplitudes below spectrum (C)
- **ins**  Integral normalization scale (P)
- **pirn**  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)**

Applicability: All systems; however, although **dplane** is available on MERCURYplus/Vx, such systems can only process 3D data and cannot acquire 3D data.

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 f1f3, f2f3, and f1f2 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 f1f3, the range of **plane_number** is 1 to fn2/2
  - For plane f2f3, the range of **plane_number** is 1 to fn1/2
  - For plane f1f2, the range of **plane_number** is 1 to fn/2
Examples:

dplane(3)
dplane('f1f2',2)

See also: *VnmrJ Liquids NMR*

Related:

dasplanes 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: *VnmrJ Liquids NMR*

Related:

pprofile Plot pulse excitation profile (M)
Pbox Pulse shaping software (U)
**dproj**

**Display a 3D plane projection (M)**

**Applicability:** All systems; however, although dproj is available on MERCURYplus/Vx, such systems can only process 3D data and cannot acquire 3D data.

**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 f1f3, f2f3, and f1f2 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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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 in the CONFIG window (opened by `config`). The Upper Limit entry in CONFIG 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:** On INOVA, 79 dB, -16 to +63, in steps of 1 dB. On MERCURYplus/Vx, 63 dB, 0 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 CONFIG window (opened by `config`).

**Values:** On INOVA, 79 dB, -16 to +63, in steps of 1 dB. On MERCURYplus/Vx, 63 dB, 0 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:** *VnmrJ Liquids NMR*

**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 CONFIG 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: VnmrJ Liquids NMR
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 CONFIG 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 \( dpwr4 = 52 \) 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: VnmrJ Liquids NMR
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 on UNITY/INOVA, or on solids systems. Systems with this attenuator are designated within the CONFIG...
window (opened by \texttt{config}) by the status of the Fine Attenuator entry. The fine attenuator is linear and spans 6 dB.

On \textit{MERCURY}plus/\textit{Vx} systems, \texttt{dpwr} controls the decoupler by simulating a fine attenuator. The fine power control is linear and spans 0 to \texttt{dpwr}.

Values: 0 to 4095 (where 4095 is maximum power). If \texttt{dpwr} does not exist in the parameter table, a value of 4095 is assumed.

On \textit{MERCURY}plus/\textit{Vx} systems, 0 to 255 (where 255 is maximum power). If \texttt{dpwr} or \texttt{dpwrm} does not exist in the parameter table, a value of 255 is assumed. If both exist, \texttt{dpwrm} is used.

See also: \textit{User Programming};\textit{User Guide: Solids}; \textit{MERCURY}plus and \textit{MERCURY-\textit{Vx} CP/MAS Installation, Testing, and Operation}

\texttt{dpwr}2 \textbf{Second decoupler fine power (P)}

\textbf{Applicability:} Systems with an optional fine attenuator on the second decoupler channel.

\textbf{Description:} Controls the second decoupler fine attenuator, functioning analogously to \texttt{dpwr}.

Values: 0 to 4095 (where 4095 is maximum power). If \texttt{dpwr}2 does not exist in the
parameter table, a value of 4095 is assumed.

See also: \textit{User Programming}

Related: \texttt{dpwr} \textit{First decoupler fine power (P)}

\texttt{dpwr}3 \textbf{Third decoupler fine power (P)}

\textbf{Applicability:} Systems with an optional fine attenuator on the third decoupler channel.

\textbf{Description:} Controls the third decoupler fine attenuator, functioning analogously to \texttt{dpwr}.

Values: 0 to 4095 (where 4095 is maximum power). If \texttt{dpwr}3 does not exist in the
parameter table, a value of 4095 is assumed.

See also: \textit{User Programming}

Related: \texttt{dpwr} \textit{First decoupler fine power (P)}

\texttt{dpwrm} \textbf{First decoupler linear modulator power (P)}

\textbf{Applicability:} \textit{UNITY}, \textit{INOVA}, and \textit{MERCURY}plus/\textit{Vx} systems with a first decoupler linear modulator.

On \textit{MERCURY} systems, \texttt{dpwrm} controls the decoupler by simulating a fine attenuator. The fine power control is linear and spans 0 to \texttt{dpwr}.

Values: 0 to 4095 (where 4095 is maximum power). If \texttt{dpwrm} does not exist in the
parameter table, a value of 4095 is assumed.

On \textit{MERCURY}plus/\textit{Vx} systems, 0 to 255 (where 255 is maximum power). If \texttt{dpwrm} does not exist in the parameter table, a value of 255 is assumed.
See also: *User Programming; User Guide: Solids; MERCURYplus/Vx CP/MAS Installation, Testing, and Operation*

**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:** UNITY/INOVA 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:** UNITY/INOVA 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**  
**Set up parameters for double-quantum filtered COSY (M)**

**Description:** Macro to set up a double-quantum filtered COSY (homonuclear correlation) experiment.

**See also:** *VnmrJ Liquids NMR*

**Related:**
- `cosyps` Set up parameters for phase-sensitive COSY (M)
- `relayh` Set up parameters for COSY pulse sequence (M)

---

**Dqcosy**  
**Convert the parameter to a DQCOSY experiment (M)**

**Description:** Convert the parameter to a DQCOSY experiment

**DQCOSY**  
**Change parameters for DQCOSY experiment (M)**

**Description:** Converts the current parameter set to a DQCOSY experiment.

---

**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:
VnmrJ Liquids NMR

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)
wcmx Maximum width of chart (P)
wcm2x Maximum width of chart in second direction (P))

drawslice Display target slices (M)

Applicability: Systems with imaging capabilities.

Description: Displays target slices defined by the file curexp+'/mark2d.out'. The program shows graphically the position and orientation of the selected target slices on a scout image. This macro is also called by the Show Target button in the slice planner menu. See the plan macro for more details.

See also:
VnmrJ Imaging NMR

Related:
curexp Current experiment directory (P)
drawvox Display target voxels (M)
plan Display menu for planning a target scan (M)
ssplan Set slice parameters for target slice (M)
voxplan Set voxel parameters for voxel defined by 2D box cursor (M)

drawvox Display target voxels (M)

Applicability: Systems with imaging capabilities.

Description: Displays target voxels defined by the file curexp+'/mark2d.out'. This program shows graphically the position of the selected target voxels on the scout image. The user can plan and then display more than one voxel with this macro. This macro is also called by the Show Target button in the voxel planner menu. See the plan macro for more details.

See also:
VnmrJ Imaging NMR

Related:
curexp Current experiment directory (P)
drawslice Display target slices (M)
**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:** VnmrJ Liquids NMR; 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:** VnmrJ Liquids NMR

**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.
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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 \texttt{trace} equal to \texttt{"f1"} or \texttt{"f2"}, respectively. After entering \texttt{ft1d}, interferograms can be viewed by setting \texttt{trace=\texttt{"f1"}} and then typing \texttt{ds}.

Spectra are scaled according to the number of completed transients \texttt{ct}. If \texttt{nt} is arrayed (\texttt{nt=1, 2, 4, 8}), each spectrum is scaled by its own \texttt{ct}.

Arguments: \texttt{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).

\texttt{options} (used with syntax 2) is any of the following keywords:
- \texttt{"toggle"} switches between the box and the cursor modes.
- \texttt{"restart"} redraws the cursor if it has been turned off.
- \texttt{"expand"} toggles between expanded and full view of the spectrum.
- \texttt{"spwp"} interactively adjusts start and width of the spectrum display.
- \texttt{"phase"} enters an interactive phasing mode.
- \texttt{"thresh"} interactively adjusts the threshold.
- \texttt{"z"} interactively sets integral resets.
- \texttt{"dscale"} toggles the scale below the spectrum on and off.
- \texttt{"lvl\_ltlt"} interactively adjusts the \texttt{lvl} and \texttt{tlt} parameters.
- \texttt{"scwc"} interactively adjusts the start and width of chart.

Examples:
- \texttt{ds}
- \texttt{ds(7)}
- \texttt{ds('restart')}

See also: \textit{VnmrJ Liquids NMR}

Related:
- \texttt{crmode} Current state of cursors in \texttt{dfid}, \texttt{ds}, or \texttt{dconi} (P)
- \texttt{ct} Completed transients (P)
- \texttt{ft1d} Fourier transform along \( f_2 \) dimension (C)
- \texttt{intmod} Integral display mode (P)
- \texttt{lp} First-order phase in directly detected dimension (P)
- \texttt{lvl} Zero-order baseline correction (P)
- \texttt{lvl\_ltlt} Control sensitivity of \texttt{lvl} and \texttt{tlt} adjustments (P)
- \texttt{nt} Number of transients (P)
- \texttt{phasing} Control update region during \texttt{ds} phasing (P)
- \texttt{rp} Zero-order phase in directly detected dimension (P)
- \texttt{select} Select a spectrum without displaying it (C)
- \texttt{tlt} First-order baseline correction (P)
- \texttt{trace} Mode for n-dimensional data display (P)
- \texttt{wft1d} Weight and Fourier transform \( f_2 \) for 2D data (C)

\textbf{ds2d} \hspace{1cm} \textbf{Display 2D spectra in whitewash mode (C)}

**Syntax:** \texttt{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 \texttt{dcon}), because intensity can be seen visually, but instead successive traces are displayed in different colors so that color represents frequency.

**Arguments:** \texttt{options} can be any of the following keywords:
- \texttt{"nobase"} is a keyword to activate the \texttt{th} parameter to suppress all intensity below the \texttt{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: VnmrJ Liquids NMR

Related: dcon Display noninteractive color intensity map (C)
dconi Control display selection for the dconi program (P)
da2dn Display 2D spectra in whitewash mode without screen erase (C)
pl2d Plot 2D spectra in whitewash mode (C)

Display 2D spectra in whitewash mode without screen erase (C)

ds2dn 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: VnmrJ Liquids NMR

Related: ds2d Display 2D spectra in whitewash mode (C)

dscale Display scale below spectrum or FID (C)

dscale 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: VnmrJ Liquids NMR

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)

dseq

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: VnmrJ Liquids NMR

Related:
addpar Add selected parameters to current experiment (M)
downsamp Downsampling factor applied after digital filtering (P)
dsfb Digital filter bandwidth for downsampling (P)
dalsfrq Bandpass filter offset for downsampling (P)
filtfile File of FIR digital filter coefficients (P)
pards Create additional parameters used for downsampling (M)

dseq2

Decoupler sequence for second decoupler (P)

Applicability: Systems with waveform generators.

See also: VnmrJ Liquids NMR

Related:
dmm Decoupler modulation mode for first decoupler (P)
dseq2 Decoupler sequence for second decoupler (P)
dseq3 Decoupler sequence for third decoupler (P)
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., \texttt{dmm2='p'}). The decoupling sequence must be located in the user's \texttt{shapelib} directory or in the VnmrJ system \texttt{shapelib} directory.

See also: \textit{VnmrJ Liquids NMR}

Related:

\begin{itemize}
  \item \texttt{dseq2} Decoupler sequence for second decoupler (P)
  \item \texttt{dseq} Decoupler sequence for first decoupler (P)
\end{itemize}

\textbf{dseq3}\quad\textbf{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., \texttt{dmm3='p'}). The decoupling sequence must be located in the user's \texttt{shapelib} directory or in the system's \texttt{shapelib} directory.

See also: \textit{VnmrJ Liquids NMR}

Related:

\begin{itemize}
  \item \texttt{dmm3} Decoupler modulation mode for third decoupler (P)
  \item \texttt{dseq} Decoupler sequence for first decoupler (P)
\end{itemize}

\textbf{dseq4}\quad\textbf{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 fourth decoupler under status control (i.e., \texttt{dmm4='p'}). The decoupling sequence must be located in the user's \texttt{shapelib} directory or in the system's \texttt{shapelib} directory.

See also: \textit{VnmrJ Liquids NMR}

Related:

\begin{itemize}
  \item \texttt{dmm4} Decoupler modulation mode for third decoupler (P)
  \item \texttt{dseq} Decoupler sequence for first decoupler (P)
\end{itemize}

\textbf{dsfb}\quad\textbf{Digital filter bandwidth for downsampling (P)}

Description: Specifies the bandwidth of the digital filter used for downsampling. If \texttt{dsfb} does not exist in the current experiment, enter \texttt{addpar('downsamp')} to add it. \texttt{addpar('downsamp')} creates the digital filtering and downsampling parameters \texttt{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 $\pm \text{sw}/2$.

'n' makes \texttt{dsfb} default to the final $\text{sw}/2$.

See also: \textit{VnmrJ Liquids NMR}

Related:

\begin{itemize}
  \item \texttt{addpar} Add selected parameters to current experiment (M)
  \item \texttt{downsamp} Downsampling factor applied after digital filtering (P)
  \item \texttt{dscoef} Digital filter coefficients for downsampling (P)
  \item \texttt{dslsfrq} Bandpass filter offset for downsampling (P)
  \item \texttt{filtfile} File of FIR digital filter coefficients (P)
  \item \texttt{pards} Create additional parameters used for downsampling (M)
  \item \texttt{sw} Spectral width in directly detected dimension (P)
\end{itemize}

\textbf{dshape}\quad\textbf{Display pulse shape or modulation pattern (M)}

Syntax: \texttt{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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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)

**dlsfrq**

**Bandpass filter offset for downsampling (P)**

Description: For downsampling, selects a bandpass filter that is not centered about the transmitter frequency. In this way, dlsfrq works much like lsfrq. If dlsfrq 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, dlsfrq, 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: VnmrJ Liquids NMR

Related:

addpar Add selected parameters to current experiment (M)
downsamp Dowsampling 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)
movedsw 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

\[
    noise = \left( \sum_{i=1}^{np} Y_i^2 / np \right)^{1/2}
\]

where \( Y_i^2 \) 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.
'nodsp' 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: VnmrJ Liquids NMR

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:

- **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 \((sw/2) \times oversamp\) as closely as possible.

Another type of DSP is available that allows post-processing of data. See the description of the pard macro for details.

Values:
- 'i' selects inline DSP and calls addpar('oversamp') to create the DSP parameters def_osfilt, filtfile, osccoef, 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. On systems other than UNITY/INOVA, inline DSP is not possible if interleaving is active (il='y'). Also, the command sa can be used to stop acquisition, but ra cannot be used to resume it. On UNITY/INOVA, inline DSP is completely compatible with interleaving and with stopping and restarting on acquisition with sa and ra. Set fsq='y' to use frequency-shifted quadrature detection on UNITY/INOVA.

- 'r' selects real-time DSP and calls the macro addpar('oversamp') to create the DSP parameters def_osfilt, filtfile, osccoef, 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. On UNITY/INOVA systems, 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'.
dsplanes

Display a series of 3D planes (M)

Applicability: All systems; however, although dsplanes is available on MERCURYplus/Vx systems, such systems can only process 3D data and cannot acquire 3D data.

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: VnmrJ Liquids NMR

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)
extpl 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: \texttt{dsptype?=0} \texttt{dsptype?=1}

See also: \textit{VnmrJ Liquids NMR}

Related: \textit{dsp} Type of DSP for data acquisition (P)

dss \textbf{Display stacked spectra (C)}

Syntax: \texttt{dss\langle\langle\text{start,finish<,step>},options\rangle\rangle}

Description: Displays one or more spectra on the screen, but not interactively like the command \texttt{ds}. When a single spectrum is displayed, integral display is controlled by the parameter \texttt{intmod}, which has the following values:

\begin{itemize}
  \item \texttt{intmod='off'} turns off the integral display.
  \item \texttt{intmod='full'} displays the entire integral.
  \item \texttt{intmod='partial'} displays every other integral region.
\end{itemize}

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 \texttt{f1} or \texttt{f2} domain by setting the parameter \texttt{trace} equal to \texttt{'f1'} or \texttt{'f2'}, respectively. After entering \texttt{ft1d}, interferograms can be viewed by setting \texttt{trace='f1'} and then entering \texttt{dss}. Multiple spectra can be displayed by supplying indexes of the first and last spectra.

The position of the first spectrum is governed by the parameters \texttt{wc}, \texttt{sc}, and \texttt{vp}. For 1D data, subsequent spectra are positioned relative to the preceding spectrum by the parameters \texttt{vo} (vertical offset) and \texttt{ho} (horizontal offset). For 2D data, \texttt{ho} defines the total horizontal offset between the first and last spectrum. Also for 2D data, \texttt{vo} is inactive while the parameter \texttt{wc2} defines the total vertical offset between the first and last spectrum.

The parameter \texttt{cutoff}, if it exists and is active, defines the distance above and below the current vertical position \texttt{vp} at which peaks are truncated. By arraying \texttt{cutoff} to have two different values, the truncation limits above and below the current vertical position can be controlled independently. For example, \texttt{cutoff=50} truncates peaks at \texttt{vp+50 mm} and \texttt{vp–50 mm}. \texttt{cutoff=50,10} truncates peaks at \texttt{vp+50 mm} and \texttt{vp–10 mm}.

Arguments:

\begin{itemize}
  \item \texttt{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.
  \item \texttt{finish} is the index of the last spectra when displaying multiple spectra. Since the parameter \texttt{arraydim} is automatically set to the total number of spectra, it can be used to set \texttt{finish} to include all spectra (e.g., \texttt{dss(1,arraydim,3)}).
  \item \texttt{step} is the increment for the spectral index when displaying multiple spectra. The default is 1.
  \item \texttt{options} can be any of the following:
    \begin{itemize}
      \item \texttt{'all'} is a keyword to display all of the spectra.
      \item \texttt{'int'} is a keyword to only display the integral, independently of the value of the parameter \texttt{intmod}
      \item \texttt{'top'} or \texttt{'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 \texttt{sc}, \texttt{wc}, \texttt{sc2}, and \texttt{wc2} are those used to position the contour plot.
      \item \texttt{'dodc'} is a keyword for all spectra to be drift corrected independently.
      \item \texttt{'red'}, \texttt{'green'}, \texttt{'blue'}, \texttt{'cyan'}, \texttt{'magenta'}, \texttt{'yellow'}, \texttt{'black'}, and \texttt{'white'} are keywords that select a color.
    \end{itemize}
\end{itemize}
Examples:
```
dss(1,3)
dss(1,12,3,'green')
```

See also: *VnmrJ Liquids NMR*

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)
- `ftld`  Fourier transform along f2 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)
- `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. When a single spectrum is displayed, integral display is controlled by the parameter `intmod`, which has the following values:

- `intmod='off'` turns off the integral display.
- `intmod='full'` displays the entire integral.
- `intmod='partial'` displays every other integral region.

For arrayed 1D spectra or for 2D spectra, a particular trace can be viewed by supplying the index number. For 2D data sets, spectra can be displayed from either the f1 or f2 domain by setting the parameter `trace` equal to `'f1'` or `'f2'`, respectively. Following the command `ft1d`, interferograms may be viewed by setting `trace='f1'` and then entering `dssa`. Multiple spectra can be 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.

Examples:  
- `dssa(1,3)`

See also: *VnmrJ Liquids NMR*

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)
- `dssan` Display stacked spectra automatically without erasing (C)
- `dssh` 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 f2 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)
- `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: *VnmrJ Liquids NMR*

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. When a single spectrum is displayed, integral display is controlled by the parameter `intmod`, which can have the following values:
- `intmod='off'` turns off the integral display.
- `intmod='full'` displays the entire integral.
- `intmod='partial'` displays every other integral region.

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 f1 or f2 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 entering `dss`. Multiple spectra can be 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.

**Examples:**
```
dssh(1,3)
```

**See also:** *VnmrJ Liquids NMR*

**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)
- `dssn` Display spectra in whitewash mode (C)
- `ft1d` Fourier transform along f2 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)
- `trace` Mode for 2D data display (P)
- `vo` Vertical offset (P)
- `vp` Vertical position of spectrum (P)
D

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: VnmrJ Liquids NMR
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. Note that if wysiwyg='n', labels can appear at incorrect positions. The positions were 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.
- 'value' is a keyword that produces a display of the values of each array element, instead of an integer index.
- '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.
Examples: dssl
dssl('top','left')
dssl('value','format=%3.1f')
See also: VnmrJ Liquids NMR
Related: dss Display stacked spectra (C)
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: VnmrJ Liquids NMR
Related: dss Display stacked spectra (C)
dsvast

**Display VAST data in a stacked 1D-NMR matrix format (M)**

**Applicability:** Systems with the VAST accessory.

**Syntax:** `dsvast<(display order,number of columns displayed)>`

**Description:** `dsvast` will arrange and display the traces from a reconstructed 2D data set (see `vastglue`) as an array of 1D spectra in a matrix of 1D spectra. If no arguments are provided, the number of rows and columns will be determined by the periodicity of the display order based on the `doneQ`. For example, if a block of 96 spectra (typical for a microtiter-plate) have been acquired using VAST automation, the spectra will be displayed in a matrix 8 rows and 12 columns with the well label using the format [A->H][1->12]. The spectra can be plotted using the macro `plvast`.

**Arguments:**
- **display order** is optional and its default value is the glue order as listed in `glueorderarray`. A display order can be defined using the `plate_glue` program.
- **number of columns displayed**. The default value is deduced by examining the periodicity of the requested display order. The number of columns displayed can entered as the second argument or as the first argument if the default display order is used.

**Examples:**
- `dsvast`
- `dsvast(12)`
- `dsvast('glue_file', 4)`

**Related:**
- `dsast2d` 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)
- `plate_glue` Define a display order (U)

**dsvast2d**

**Display VAST data in a pseudo-2D format (M)**

**Applicability:** Systems with the VAST accessory.

**Syntax:** `dsvast2d(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`), this macro will arrange and display them (on the screen) 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 8 spectra per row.

**Arguments:**
- The default is to display all the spectra (from 1 through `arraydim`) with 8 columns (spectra) and 12 rows. An optional argument `dsvast2d(number)` allows specifying that only spectra from 1 through `number` should be plotted. The number of spectra displayed is rounded up to the nearest multiple of 8.

**Related:**
- `dsast` Display VAST data in a 1D-NMR matrix format (M)
- `plvast` Plot VAST data in a stacked 1D-NMR matrix (M)
- `plvast2d` Plot VAST data in a pseudo-2D format (M)

**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: dww(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)
eject sample (M)
display ethernet address (M,U)
set up parameters to get eddy current compensation data (M)
put gcal value and ecc file into table (M)
open eccTool window (M)
display strings and parameter values in text window (C)
current echo index for transformed image (P)
data analysis of eddy current compensation (M)
update acquisition eddy current settings (M)
display directory on remote VXR-style system (M,U)
current array index for transformed image (P)
Enter sample information for automation run (M,U)
start a dialog window using enterexp file (M)
process and display image in EPI experiments (M)
generate phasemap file in EPI experiments (M)
reverse spectral data in EPI experiments (C)
collect, process, and display EPI data (M)
set up parameters for EPI experiments (M)
save EPI images in FDF for ImageBrowser (M)
transfer file from remote source (M,U)
calculate the Ernst angle pulse (C)
display recent error messages (C)
number of lines in error message display (P)
transfer file to remote destination (M,U)
execute a command (C)
set up the exec parameters (M)
execute plotting macro (P)
execute prepare macro (P)
execute prescan macro (P)
execute processing macro (P)
execute setup macro (P)
checks if parameter, file, or macro exists and file type (C)
call the vnmrexit command (M)
find exponential value of a number (C)
determine if experiment has active acquisition (C)
make least-squares fit to polynomial or exponential curve (U)
display exponential or polynomial curves (C)
add another diffusion analysis to current display (M)
display experiment library (M)
display current experiment chain and approx. time for each (M)
**E**

**explog**  
Display log file for experiment (M)

**exptime**  
Display experiment time (C)

**e**  
**Eject sample (M)**

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 eject macro.

See also: **VnmrJ Liquids NMR**

Related:  
eject  Eject sample (M)  
i  Insert sample (M)  
insert  Insert sample (M)

**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: **VnmrJ Liquids NMR**

Related:  
dnode  Display list of valid limNET nodes (M,U)

**ecc**  
**Set up parameters to get eddy current compensation data (M)**

Applicability: Systems with the imaging module.

Description: Loads parameter sets during imaging installation for a pulse sequence to obtain eddy current compensation data using balance gradients.

See also: **Imaging Module Installation Manual**

Related:  
eddyout  Data analysis of eddy current compensation (M)

**ecctabl**  
**Put gcal value and ecc file into table (M)**

Applicability: Systems with the imaging module.

Syntax: ecctabl<(ecc_file><,gcal>)>

Description: Moves the gcal value and ecc file into the reference table ecctabl in $vnmrsystem/imaging/eddylib. If the gcal value or file name would overwrite data already in the table, the monitor displays a prompt to confirm the overwrite.

Arguments:  
ecc_file specifies the name of the ecc file to be placed in the ecctabl reference table. The default value is the file name 'curecc'.

gcal specifies the gcal value to be placed in the ecctabl reference table. The default is the current gcal value.

Examples:  
ecctabl  
ecctabl('test1',0.001)

See also: **VnmrJ Imaging NMR**

Related:  
ecc  Set up parameters to obtain compensation data (M)  
gcal  Gradient calibration constant (P)  
getgcal  Get gcal value from table (M)

**ecctool**  
**Open eccTool window (M)**

Applicability: Systems with imaging capabilities.
Description: Opens the eccTool window to adjust eddy current compensation parameters.

See also: *VnmrJ Imaging NMR*

echo

**Display strings and parameter values in text window (C)**

**Syntax:** echo<(<'–n',>string1,string2, ...)> 

**Description:** Displays strings and parameter values in the text window similar to the UNIX echo command.

**Arguments:** '–n' is a keyword that suppresses advancing to the next line. The default is to advance to the next line.

string1,string2,... are one or more strings (surrounded with single quote marks) or parameters. The format used for numbers is identical to the %g format described for the write command.

**Examples:**
echo
  echo('This is a string')
echo('Pulse Width is: ',pwr)
echo('–n','No new line')

See also: *User Programming*

**Related:**

write Write formatted text to a device (C)

**echo**

**Current echo index for transformed image (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Stores the current echo index for the transformed image.

**See also:** *VnmrJ Imaging NMR*

**Related:**

element Current array index for transformed image (P)

eddyout

**Data analysis of eddy current compensation (M)**

**Applicability:** Systems with the imaging module.

**Syntax:** eddyout(start,stop)

**Description:** Analyzes the data obtained with the pulse sequence set up by ecc for a series of acquisitions obtained after varying delays following shut off of a gradient. eddyout calculates the time constants and amplitudes of the eddy currents and recommends new time constants and amplitudes to be set into the compensation networks.

**Arguments:**

start specifies the number of starting array of spectra acquired by ecc.

stop specifies the number of the ending array of spectra acquired by ecc.

**Examples:** eddyout(1,16)

**See also:** *VnmrJ Imaging NMR*

**Related:**

ecc Set up parameters to obtain compensation data (M)

eddysend

**Update acquisition eddy current settings (M)**

**Applicability:** Systems with the imaging module.

**Syntax:** eddysend<(file)>

**Description:** Assigns the compensation data from eccTool to the current eddy current compensation file specified by curecc, then sets the compensation data into
the acquisition system. eccTool uses eddysend to automatically track the file(s) in use by eccTool.

Arguments: file is the file name of data from eccTool. If that file exists, that data is assigned to the current compensation file and becomes curecc. The default is the data in the current compensation file is loaded from curecc.

Examples: eddysend
eddysend('data04')

See also: VnmrJ Imaging NMR
Related: curecc Name of eddy current compensation file (P)
        eccTool Pop up eccTool window (M)

edit Edit a file with user-selectable editor (M)
Syntax: edit(file)
Description: Opens a file for editing using a text editor. The default editor is vi. To select another editor, set the UNIX environmental variable vnmreditor to the name of the editor (change the line setenv vnmreditor old_editor in .login 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 (e.g., vnmr_emacs) is placed in the bin subdirectory of the system directory. The script file makes adjustments for the type of graphic interface in use.

Scripts provided with VnmrJ include vnmr_vi and vnmr_textedit. To create other scripts, see the vnmr_vi script for non-window editor interfaces and the vnmr_textedit script for window-based editor interfaces.

Arguments: file is the name of the file you wish to edit.
Examples: edit('myfile')
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)
        macroedit Edit a user macro with user-selectable editor (C)
        macrovi Edit a user macro with vi editor (C)
        menuvi Edit a menu with the vi editor (M)
        textvi Edit text file of current experiment with vi editor (M)

eff_echo Effective echo position in EPI experiments (P)
Applicability: Systems with echo planar imaging (EPI) capabilities.
Description: Refers to the echo showing the highest signal in an EPI echo-train. The readout gradient dephaser is adjusted so that the maximum signal occurs at eff_echo.

Values: Usually set to nv/2.
See also: VnmrJ Imaging NMR
Related: nv Number of phase encode steps for 1st indirectly detected dim. (P)

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: VnmrJ Liquids NMR

 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: VnmrJ Liquids NMR
Related: dnode  Display list of valid limNET nodes (M,U)

element  Current array index for transformed image (P)
Applicability: Systems with imaging capabilities.
Description: Stores the current array index for the transformed image.
See also: VnmrJ Imaging NMR
Related: echo  Current echo index for transformed image (P)

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: VnmrJ Liquids NMR; User Programming, VnmrJ Walkup NMR

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.

**Related:** enter Enter sample information for automation run (M,U)

**epift**  
*Process and display image in EPI experiments (M)*

**Applicability:** Systems with echo planar imaging (EPI) capabilities.

**Syntax:** epift(index)

**Description:** Processes and displays an image in array number index. The first data array must contain the reference scan. The phase correction information saved in the file phasemap is used to correct phase errors in EPI data. phasemap must be present in the current experiment directory. Use dconi to view the data.

**Arguments:** index is the array number of the image.

**See also:** VnmrJ Imaging NMR

**Related:** dconi Interactive 2D data display (C)
        epiph Generate phase correction map in EPI experiments (M)
        pcmapapply Apply phase correction map to data in EPI experiments (C)

**epiph**  
*Generate phasemap file in EPI experiments (M)*

**Applicability:** Systems with echo planar imaging (EPI) capabilities.

**Description:** Generates the phasemap file from the EPI reference scan. The file is generated in the current experiment directory for EPI processing. The first data array must correspond to the reference scan, which is collected with the phase-encode gradient turned off (image=0).

**See also:** VnmrJ Imaging NMR

**Related:** episet Set up parameters for EPI experiments (M)
        image Control phase encoding gradient in EPI experiments (P)
        pcmapgen Generate phase correction map in EPI experiments (M)

**epirs**  
*Reverse spectral data in EPI experiments (C)*

**Applicability:** Systems with echo planar imaging (EPI) capabilities.

**Description:** Reverses spectral data. It is used by **epift.**
See also: *VnmrJ Imaging NMR*
Related: `epif` Process and display images in EPI experiments (M)

### `epirun`
**Collect, process, and display EPI data (M)**

**Applicability:** Systems with echo planar imaging (EPI) capabilities.

**Description:** Collects, process, and displays EPI data. It is used to obtain a single EPI image. The phasemap file must be present in the current experiment directory.

See also: *VnmrJ Imaging NMR*
Related: `epiph` Generate phasemap file in EPI experiments (M)
`episet` Set up parameters for EPI experiments (M)

### `episet`
**Set up parameters for EPI experiments (M)**

**Applicability:** Systems with echo planar imaging (EPI) capabilities.

**Description:** Collects an EPI dataset with the phase-encode gradient turned off ($image=0$). It optimizes parameters for EPI, collects a reference scan, and allows you to adjust the gradient parameters `groa` and `grora` and the timing parameter `tep`. The phasemap file is generated in the current experiment directory.

See also: *VnmrJ Imaging NMR*
Related: `epiph` Generate phasemap file in EPI experiments (M)
`groa` Readout gradient adjuster in EPI experiments (P)
`grora` Readout dephasing gradient adjuster in EPI experiments (P)
`image` Control phase encoding gradient in EPI experiments (P)
`tep` Post-acquisition delay in EPI experiment (P)

### `episvib`
**Save EPI images in FDF for ImageBrowser (M)**

**Applicability:** Systems with echo planar imaging (EPI) capabilities.

**Description:** Saves images in Flexible Data Format (FDF) for viewing with ImageBrowser. The first image in an arrayed dataset must contain the reference scan. This scan must be acquired with the phase encode gradient turned off.

See also: *VnmrJ Imaging NMR*
Related: `browser` Start ImageBrowser application (U)

### `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:  *VnmrJ Liquids NMR*

Related:  `dnode` Display list of valid limNET nodes (M,U)
`ewrite` Transfer file to remote destination (M,U)

### ernst

**Calculate the Ernst angle pulse (C)**

Syntax:  `ernst(t1_estimate<,90_pulse_width>)`

Description: Calculates the optimum (“Ernst”) pulse width according to the formula

\[
pw = \cos^{-1}\left(\exp\left(-\frac{t_1 + d}{t_1 \text{ estimate}}\right)\right) \left(\frac{90}{360}\right)
\]

The new `pw` value is entered in the parameter table.

Arguments:  
- `t1_estimate` is an estimate of the $T_1$ for a peak of interest.
- `90_pulse_width` is a 90° pulse width determined by the parameter `pw90`. The default is the current value of parameter `pw90` if `pw90` exists.

Examples:  `ernst(5)`
`ernst(3,12.6)`

See also:  *VnmrJ Liquids NMR*

Related:  `pw` Pulse width (P)
`pw90` 90° pulse width (P)

### errlog

**Display recent error messages (C)**

Description: Displays in the text window the most recent error messages. The global parameter `errloglen` controls the number of lines displayed. If `errloglen` is not defined, `errlog` displays 10 lines by default.

See also:  *VnmrJ Liquids NMR*

Related:  `acqstatus` Acquisition status (P)
`errloglen` Number of lines in error message display (P)

### errloglen

**Number of lines in error message display (P)**

Description: Sets the number of lines in the display of error messages by the `errlog` command.

Values:  Integer, default is 10.

See also:  *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: `dnode` Display list of valid limNET nodes (M,U)
`eread` Transfer file from remote source (M,U)

**exec**  
Execute a command (C)

Syntax: exec(command_string)

Description: Executes the command given by the string argument.

Arguments: `command_string` is a character string constructed from a macro.

Examples: `exec($cmdstr)`
`exec(parstyle)`

See also: *User Programming*

**execpars**  
Set up the exec parameters (M)

Description: Set up the exec parameters as listed in `/vnmr/execpars`.

See also: *User Programming*

**execplot**  
Execute plotting macro (P)

Description: Defines which plotting macro to use to plot this experiment.

See also: *User Programming*

**execprep**  
Execute prepare macro (P)

Description: Defines which prepare macro to use to prescan this experiment.

See also: *User Programming*

**execprescan**  
Execute prescan macro (P)

Description: Defines which prescan macro to use to prescan this experiment.

See also: *User Programming*

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

Syntax:
1. exists (name, 'parameter'<,tree>):$exists
2. exists (name, 'file'<,permission>):$exists
3. exists (name, 'maclib'): $exists
4. exists (name, 'command'): $exists
5. exists (name, 'ascii'): $exists
6. exists (name, 'directory'): $exists

Description: Checks for the existence of a parameter, file, command, or a macro from within a macro. It also checks if a file is an ASCII text file or a directory.

Arguments:
- name is the name of a parameter, file, command, or macro.
- 'parameter' checks if the parameter specified by name exists.
- 'tree' is 'global', 'current', 'processed', or 'systemglobal'. The default is 'current'. Refer to the create command for more information on the types of parameter trees.
- 'file' checks if the file specified by name exists.
- 'permission' is a string to be used with an access permission test on the file specified by name. The default is to check only the simple existence of the file. Access permission can be identified 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, the command exists('/vnmr/conpar','file','rw') checks not only that the file /vnmr/conpar exists, but also whether the current user has read and write access to that file.
- 'maclib' checks if the macro specified by name exists.
- 'command' checks if the command or macro specified by name exists.
- 'ascii' checks if the file specified by name is an ASCII text file.
- 'directory' checks if the file specified by name is a directory.

$exists is the return variable that changes according to the second argument:

- For 'parameter', exists returns 1 if the parameter specified by name exists in the tree specified by tree; otherwise, it returns 0.
- For 'file', exists returns 1 if the file specified by name exists with the file permission specified by permission; otherwise, it returns 0.
- For 'maclib', exists searches the macro libraries in the following order for the macro specified in name and returns 1 if the macro is in the user's maclib directory, returns 2 if in a directory defined by maclibpath, returns 3 if in a directory defined by sysmaclibpath, returns 4 if in the system maclib directory, or returns 0 if not found in any of these libraries. Only the value of the first location found is returned.
- For 'command', exists searches the command list and macro libraries in the following order and returns 1 if name is a command, returns 2 if it is in the user's maclib directory, returns 3 if in a directory defined by maclibpath, returns 4 if in a directory defined by sysmaclibpath, returns 5 if in the system maclib directory, or returns 0 if not found in any of these libraries. Only the value of the first location found is returned.
- For 'ascii', exists returns 1 if the file specified in name is an ASCII text file; otherwise it returns 0.
- For 'directory', exists returns 1 if the file specified in name is a directory; otherwise it returns 0.

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.
**exit**

**Call the vnmrexit command (M)**

**Description:** Calls the `vnmrexit` command to exit from VnmrJ. As a macro, `exit` provides a user some flexibility in defining other things to do when exiting.

**CAUTION:** When you exit from the VnmrJ 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 VnmrJ running on your system. Failure to do this can cause current parameter values and even current data to be lost.

**exp**

**Find exponential value of a number (C)**

**Syntax:** `exp(value):n`

**Description:** Finds the exponential value (base $e$) of a number.

**Arguments:**
- `value` is a number.
- `n` is the return value giving the exponential value of `value`. The default is to display the exponential value in the status window.

**Examples:**
- `exp(.5)`
- `exp(val):exp_val`

**See also:** User Programming

**Related:**
- `arccos` Calculate arc cosine of real number (M)
- `arcsin` Calculate arc sine of real number (M)
- `arctan` Calculate arc tangent of real number (M)
- `atan` Find arc tangent of a number (C)
- `cos` Find cosine value of an angle (C)
- `ln` Find natural logarithm of a number (C)
- `sin` Find sine value of an angle (C)
- `tan` Find tangent value of an angle (C)

**expactive**

**Determine if experiment has active acquisition (C)**

**Syntax:**
1. `expactive<(exp_number)>:$answer`
2. `expactive('auto'):$mode`
3. `expactive('current'):<$exp>,<$user>`

**Description:** Determines whether an acquisition is active or pending in an experiment.

**Arguments:**
- `exp_number` is the number, from 1 to 9999, of the experiment to be checked. The default is the current experiment.
- `$answer` is a return value: -1 if an acquisition is not possible (e.g., the system is a data station), 0 if no acquisition active in the requested experiment, 1 if an acquisition active in that experiment, and 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). With no return argument, the result displays on line 3.
- `'auto'` is a keyword to check if the system is in automation mode.
E

$mode$ is a return value: 1 if the system is in automation mode, or 0 if otherwise. With no return argument, the result is displayed on line 3.

'current' is a keyword that determines whether an active experiment has an active acquisition command running. An experiment is still considered active if it holds up additional acquisitions during its wexp processing by the 'wait' flag. If expactive('current') does not have a return argument, results are displayed on line 3.

$exp$ is a return value indicating the current active experiment number: −1 if no acquisition is possible, or 0 if no acquisition is active.

$user$ is a return value indicating the user who started the acquisition. If the system is running in automation mode, $user$ is set to “auto.” If no acquisition is running, $user$ is set to “nobody.”

Examples:

expactive
expactive(3)
expactive(2):$active
expactive('auto'): $automode

expfit
Make least-squares fit to polynomial or exponential curve (U)

Syntax: (From UNIX) expfit options <analyze.inp >analyze.list

Description: Makes a least-squares curve fitting to the data supplied in the file analyze.inp. For the specialized uses of analyze, VnmrJ macros (e.g., t1, t2, kind) are available that provide the correct file format and avoid the need for the user to select options.

In the regression mode, the type of curve fitting, ('poly1',...) must be selected. For regression (generalized curve fitting), the regression section in the manual VnmrJ Liquids NMR shows the input file format and describes the menus that permit option choices indirectly through menu buttons.

The following text file is an example of the file analyze.inp (for options T1, T2, kinetics, contact_time, and regression). (1), (2), etc. do not actually appear in the file but are used to identify lines in the description presented below the file.

(1) time
(2) <amp>
(3) 2 4 linear linear

(4) NEXT 4
(5) 1
(6) 1 1
  2 4
  3 9
  4 16
(4) NEXT 3
(5) 2
(6) 2 5
  3 10
  4 17

This file contains the following information:

(1) Optional x-axis title.
(2) Optional y-axis title, for regression only.
(3) Line containing an integer for the number of peaks, followed by another integer for the number of pairs per peak. 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, followed by an integer for the number of pairs for the peak.

(5) An integer that indexes the peaks.

(6) Data pairs, one to a line, listed by peak.

For options 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, the analyze.inp file is slightly different:

1. List of n x-y data pairs
2. <text line>
3. <x-values> <y-values>
4. x y ...

(1) Title line.
(2) Descriptive text line.
(3) Number of x values and y values.
(4) 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)
   ...

expfit also makes a file analyze.out that 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.

Arguments: options can be any of the following:

T1 sets T1 analysis. This value is the default.
T2 sets T2 analysis.
kinetics sets kinetics analysis with decreasing peak height.
increment sets kinetics analysis with increasing peak height.
list sets 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.
regression sets regression mode, providing generalized curve fitting with choices poly1, poly2, poly3, and exp:
   • poly0 calculates the mean.
   • poly1 sets a linear fitting.
   • poly2 sets a quadratic fitting.
   • poly3 sets a cubic curve fitting.
   • exp sets an exponential curve fitting.

Examples: (From UNIX) expfit d2 T1 list <analyze.inp >analyze.out
(From UNIX) expfit regression exp list <analyze.inp >analyze.out
See also: \textit{VnmrJ Liquids NMR}

Related: 
\begin{itemize}
\item \texttt{analyze} \hspace{1em} Generalized curve fitting (C)
\item \texttt{expl} \hspace{1em} Display exponential or polynomial curves (C)
\item \texttt{fp} \hspace{1em} Find peak heights (C)
\item \texttt{kind} \hspace{1em} Kinetics analysis, decreasing intensity (M)
\item \texttt{t1} \hspace{1em} $T_1$ exponential analysis (M)
\item \texttt{t2} \hspace{1em} $T_2$ exponential analysis (M)
\end{itemize}

\textbf{expl}

\textbf{Display exponential or polynomial curves (C)}

Syntax: \texttt{expl(<options,line1,line2,...>)}

Description: Displays exponential curves resulting from $T_1$, $T_2$, or kinetic analyses. Also displays polynomial curves from diffusion or other types of analysis. The parameters $sc$, $wc$, $sc2$, and $wc2$ control the size of the display.

In general, the first time \texttt{expl} is displayed, it calculates appropriate limits for the two axes. A subsequent call to \texttt{expl}, while a previous \texttt{expl} is displayed on the graphics screen, uses the axis scaling that displayed \texttt{expl}. To have the new \texttt{expl} recalculate its own axis limits and not use those currently displayed, call the \texttt{autoscale} macro before executing \texttt{expl}. Alternately, the axis limit for the \texttt{expl} display can be specified using the \texttt{scalelimits} macro.

Arguments: \texttt{options} can be any of the following:

\begin{itemize}
\item \texttt{'regression'} is a keyword signifying the beginning of generalized curve fitting. \texttt{expl} displays the data in the file \texttt{regression.inp} as unconnected points and also uses \texttt{regression.inp} to create the file \texttt{analyze.inp}, which serves as input to \texttt{analyze} for curve fitting.
\item \texttt{'linear'}, \texttt{'square'}, and \texttt{'log'} are keywords for display of the data points against a square or logarithmic axis scale, with the exception of the results from regression. The first keyword controls the x-axis scale, the second the y-axis. The default is \texttt{'linear'}.
\item \texttt{'link'} is a keyword to link the data points rather than a display of the theoretical curve.
\item \texttt{'nocurve'} is a keyword to produce a plot of data points only.
\item \texttt{'tinysymbol'} is a keyword to display small-scale data point symbols.
\item \texttt{'nosymbol'} is a keyword to produce a plot of the curve only.
\item \texttt{'noclear'} is a keyword to not erase the graphics screen before drawing the plot. This prevents the graphics screen from being cleared of data.
\item \texttt{'oldbox'} is a keyword to plot an additional curve on an existing plot. Only the first data set in the file \texttt{analyze.out} is plotted. The box and scale description is derived from the file \texttt{expl.out} in the current experiment. When the \texttt{'oldbox'} option is used, a second argument is necessary to identify the curve number and data point symbol to represent the data. This second argument is a number from 1 to 6.
\item \texttt{'file'} is a keyword that, when followed by a file name, makes that file replace the file \texttt{analyze.out} as the input to \texttt{expl}.
\end{itemize}

\texttt{line1, line2,...} specify the curves to be displayed. The default is to display the first eight curves (if that many exist) along with data points.

Examples: \texttt{expl}
\begin{itemize}
\item \texttt{expl(1,3,6)}
\item \texttt{expl('oldbox',5)}
\item \texttt{expl('regression')}
\item \texttt{expl('regression',4,5)}
\end{itemize}
See also: *VnmrJ Liquids NMR*

**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)`

**Related:**  
`analyze` Generalized curve fitting (C)  
`autoscale` Resume autoscaling after limits set by `scalelimits` (M)  
`expfit` Make least squares fit to polynomial or exponential curve (C)  
`pexpl` Plot exponential or polynomial curves (C)  
`sc` Start of chart (P)  
`sc2` Start of chart in second direction (P)  
`scalelimits` Set limits for scales in regression (M)  
`wc` Width of chart (P)  
`wc2` Width of chart in second direction (P)

**explib**  
**Display experiment library (M)**

**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:** *VnmrJ Liquids NMR; VnmrJ Walkup NMR*

**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 log file for experiment (M)**

**Description:** Displays the log file for an experiment. This file includes when the experiment started, any acquisition errors that may have occurred, and when the experiment finished. Each acquisition generates this information, which is stored in the experiment's `acqfil` directory in a text file named `log`.

**See also:** *VnmrJ Liquids NMR*

**exptime**  
**Display experiment time (C)**

**Syntax:**  
`exptime<(sequence)><:$seconds>`

**Related:**  
`analyze` Generalized curve fitting (C)  
`autoscale` Resume autoscaling after limits set by `scalelimits` (M)  
`expfit` Make least squares fit to polynomial or exponential curve (C)  
`pexpl` Plot exponential or polynomial curves (C)  
`pexpladd` Add another diffusion analysis to current plot (M)
Description: Estimates the acquisition time for an experiment, based on the parameters used in the current experiment, and displays the time in the format hh:mm:ss. The time macro uses exptime to determine the time of an experiment.

Arguments: sequence is a pulse sequence that exists in the seqlib directory. If this argument is used, exptime estimates the acquisition time for the specified sequence. The default is the current value of seqlib.

$seconds is a return argument with the number of seconds estimated for the experiment. If this argument is used, the time display is suppressed.

Examples:
- `exptime`
- `exptime('apt')`
- `exptime:$etime`
- `exptime('noesy'):$est_time`

See also: VnmrJ Liquids NMR

Related: time Display experiment time or recalculate number of transients (M)
F

f  Set display parameters to full spectrum (C)
fl9  Automated fluorine acquisition (M)
fl9p  Process 1D fluorine spectra (M)
fl1coef  Coefficient to construct F1 interferogram (P)
fl2coef  Coefficient to construct F2 interferogram (P)
fattn  Fine attenuator (P)
fb  Filter bandwidth (P)
fbc  Apply baseline correction for each spectrum in an array (M)
fdfgluer  Make FDF file from header and data parts (U)
fdfsplt  Divide FDF file into header and data parts (U)
fdml  Set, write 1D FDM parameters, run FDM (M)
fddc3d  3D time-domain dc correction (P)
fiddle  Perform reference deconvolution (M)
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)
fidpar  Add parameters for FID display in current experiment (M)
fidsave  Save data (M)
fifolpsize  FIFO loop size (P)
fixgrd  Convert gauss/cm value to DAC (M)
file  File name of parameter set (P)
files  Interactively handle files (C)
filesinfo  Return file information for files display (C)
filter  Gaussian low-pass filter for image processing (M)
filtfile  File of FIR digital filter coefficients (P)
fitplot  Adjust plot parameters (M)
fitspec  Perform spectrum deconvolution (C, U)
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)
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)
fliplist  Standard flip angle list (P)
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)
**f**

**Set display parameters to full spectrum (C)**

**Syntax:**

```
f <solvent>
```

**Description:**
Sets up the `sp` and `wp` display parameters for a full display of a 1D spectrum. If an FID is displayed, the parameters `sf` and `wf` are set for a full display. In multidimensional data sets, the parameters for both displayed dimensions are 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 are set.

**Related:**
- `sf`: Start of FID (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)
- `wf`: Width of FID (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)

**See also:**
*VnmrJ Liquids NMR*

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>f</code></td>
<td>Set display parameters to full spectrum (C)</td>
</tr>
<tr>
<td><code>fn2</code></td>
<td>Fourier number in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td><code>fn2D</code></td>
<td>Fourier number to build up 2D DOSY display in freq. domain (P)</td>
</tr>
<tr>
<td><code>focus</code></td>
<td>Send keyboard focus to input window (C)</td>
</tr>
<tr>
<td><code>foldcc</code></td>
<td>Fold INADEQUATE data about two-quantum axis (C)</td>
</tr>
<tr>
<td><code>foldj</code></td>
<td>Fold J-resolved 2D spectrum about $f_1=0$ axis (C)</td>
</tr>
<tr>
<td><code>foldt</code></td>
<td>Fold COSY-like spectrum along diagonal axis (C)</td>
</tr>
<tr>
<td><code>fontselect</code></td>
<td>Open FontSelect window (C)</td>
</tr>
<tr>
<td><code>format</code></td>
<td>Format a real number or convert a string for output (C)</td>
</tr>
<tr>
<td><code>fp</code></td>
<td>Find peak heights or phases (C)</td>
</tr>
<tr>
<td><code>fpmult</code></td>
<td>First point multiplier for np FID data (P)</td>
</tr>
<tr>
<td><code>fpmult1</code></td>
<td>First point multiplier for ni interferogram data (P)</td>
</tr>
<tr>
<td><code>fpmult2</code></td>
<td>First point multiplier for ni2 interferogram data (P)</td>
</tr>
<tr>
<td><code>fr</code></td>
<td>Full recall of a display parameter set (M)</td>
</tr>
<tr>
<td><code>fread</code></td>
<td>Read parameters from file and load them into a tree (C)</td>
</tr>
<tr>
<td><code>fsave</code></td>
<td>Save parameters from a tree to a file (C)</td>
</tr>
<tr>
<td><code>fsq</code></td>
<td>Frequency-shifted quadrature detection (P)</td>
</tr>
<tr>
<td><code>ft</code></td>
<td>Fourier transform 1D data (C)</td>
</tr>
<tr>
<td><code>ft1d</code></td>
<td>Fourier transform along $f_2$ dimension (C)</td>
</tr>
<tr>
<td><code>ft1da</code></td>
<td>Fourier transform phase-sensitive data (M)</td>
</tr>
<tr>
<td><code>ft1dac</code></td>
<td>Combine arrayed 2D FID matrices (M)</td>
</tr>
<tr>
<td><code>ft2d</code></td>
<td>Fourier transform 2D data (C)</td>
</tr>
<tr>
<td><code>ft2da</code></td>
<td>Fourier transform phase-sensitive data (M)</td>
</tr>
<tr>
<td><code>ft2dac</code></td>
<td>Combine arrayed 2D FID matrices (M)</td>
</tr>
<tr>
<td><code>ft3d</code></td>
<td>Perform a 3D Fourier transform on a 3D FID data set (M,U)</td>
</tr>
<tr>
<td><code>full</code></td>
<td>Set display limits for a full screen (C)</td>
</tr>
<tr>
<td><code>fullsq</code></td>
<td>Display largest square 2D display (M)</td>
</tr>
<tr>
<td><code>fullt</code></td>
<td>Set display limits for a full screen with room for traces (C)</td>
</tr>
<tr>
<td><code>f19</code></td>
<td>Automated fluorine acquisition (M)</td>
</tr>
</tbody>
</table>

**Syntax:**

```
f19 <solvent>
```
### f19

**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:** VnmrJ Liquids NMR

### Related:

- `au` Submit experiment to acquisition and process data (M)
- `enter` Enter sample information for automation run (C)
- `f19` Automated fluorine acquisition (M)
- `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:** VnmrJ Liquids NMR

**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'`.

---

**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:** VnmrJ Liquids NMR

### Related:

- `au` Submit experiment to acquisition and process data (M)
- `enter` Enter sample information for automation run (C)
- `f19` Automated fluorine acquisition (M)
- `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:** VnmrJ Liquids NMR

**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: *VnmrJ Liquids NMR*

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 -1 0'`.

---

**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 CONFIG window (opened from `config`).

On *MERCURYplus/Vx* systems, `fattn` indicates if a fine attenuator is present. It is implicitly set by `config`.

**Values:**
- 0 specifies the fine attenuator is not present on the channel (Not Present choice in CONFIG window).
- 4095 specifies the fine attenuator is present on the channel (Present choice in CONFIG window).

On *MERCURYplus/-Vx* systems, `fattn` should be set to an array value of 0,0.

See also: *VnmrJ Installation and Administration; User Guide: Solids; MERCURYplus/-Vx CP/MAS Installation, Testing, and Operation*

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:**
- On *UNITY/INOVA*, if `sw` is 500,000 or less: 1000 to 256000 Hz, 1000-Hz steps.
- On *UNITY/INOVA*, if `sw` is greater than 500,000: 256 kHz, 1 MHz.
On MERCURYplus/Vx: 1 to 25 kHz and 55 kHz. Actual values are a non-linear set, entered in steps of 200, and rounded to the larger available value.

See also: VnmrJ Liquids NMR

Related: sw spectral width in directly detected dimension (P)
mrfb set the filter bandwidths for multiple receivers (P)

**fbc**

**Apply baseline correction for each spectrum in an array (M)**

**Description:** 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 blank as large an area of baseline as is possible.

See also: VnmrJ Liquids NMR

Related: dosy Process DOSY experiments (M)

**fdfgluer**

**Make FDF file from header and data parts (U)**

**Applicability:** Systems with imaging capabilities.

**Syntax:**

1. `fdfgluer <-align> header_file <data_file <output_file>>`
2. `fdfgluer -infiles template <-offset n> <-align> header_file`
3. `fdfgluer -vnmrfile fname -outfiles template <-traces n> <-align> header_file`

**Description:** Takes an FDF (flexible data format) header file defining a set of data and data from a file, files, or standard input, and combines them to form an FDF data file.

Using syntax 1 attaches a header to a raw data file. If the `data_file` argument is given (rather than being taken from standard input), a checksum is calculated and appended to the header.

Using syntax 2 takes the data from a group of raw data files whose names are `template1`, `template2`, etc. These data files can have fixed length headers, which will be ignored.

Using syntax 3 takes data from a data file, such as a FID file.

**Arguments:**

- `header_file` is the name of the header file created or edited by the user.
- `data_file` is the name of file containing data for a FDF file. If this argument is not present, `fdfgluer` takes the data from the standard input.
- `output_file` is the name of the FDF file created. If this argument is not present, `fdfgluer` puts the FDF file to the standard output.
- `-align` is a numerical argument giving the size of words that the data should be aligned on. For example, `-8` ensures that the length of the header is a multiple of 8 bytes.
- `-infiles template` gives the base name of the group of files from which to take data. `template` can be a path. `fdfgluer` will read data from files named `template1`, `template2`, `template3`, etc. in numerical order until the next sequential file name is not found.
- `-offset n` gives the number of bytes of header in the data files. The first `n` bytes of each data file are ignored.
- `-vnmrfile fname` specifies the name of a data file to use for the input data.
- `-outfiles template` specifies the base name of output files to be written using syntax 3. The template should have a “#” somewhere in it. The output files will substitute a serial number (0001, 0002,...) for the #. For example, `-outfiles myrat#.fdf` writes data to output files `myrat0001.fdf`, etc.
-traces n gives the number of traces to put in each output file in syntax 3.

See also: *VnmrJ Imaging NMR, User Programming*

Related: **fdfsplit**  
Divide FDF file into header and data parts (U)

### fdfsplit

**Divide FDF file into header and data parts (U)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** fdfsplit output_file data_file header_file

**Description:** Takes an FDF (Flexible Data Format) file and splits it into its data and header parts. Note that the header may still have a checksum value—that value should be removed after the split has completed.

**Arguments:**
- **output_file** is the name of the FDF file to be split.
- **data_file** is the file name to be given to the data part.
- **header_file** is the file name to be given to the header part.

See also: *User Programming, VnmrJ Liquids NMR*

Related: **fdfgluer**  
Make FDF file from header and data parts (C)

### 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/datdir/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.
Examples:  
fdm1('cheat', 0.8)  
fdm1('Nsig', 3000, 'Nb', 20, 'Gamm', 0.5)

See also: 
*VnmrJ Liquids NMR*

### fiddc3d

**3D time-domain dc correction (P)**

**Applicability:** All systems; however, although *fiddc3d* is available on *MERCURYplus/Vx* systems, such systems can only process 3D data and cannot acquire 3D data.

**Description:** Sets whether a 3D time-domain dc correction occurs. If *fiddc3d* does not exist, it is created by the macro *par3d*. The time-domain dc correction occurs immediately after any linear prediction operations and before all other operations on time-domain data.

**Values:** A three-character string. The default value is `'nnn'`.

- The first character refers to the f3 dimension (sw, np, fn), the second character refers to the f1 dimension (sw1, ni, fn1), and the third character refers 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.

See also: 
*VnmrJ Liquids NMR*

**Related:**
- 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)
- ptspc3d: Region-selective 3D processing (P)
- specdc3d: 3D spectral dc 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)

### fiddle

**Perform reference deconvolution (M)**

**Syntax:**
```
fiddle(option<,file><,option<,file>><,start> <,finish><,increment>)
```
Description: Performs reference deconvolution using a reference signal with known characteristics to correct instrumental errors in experimental 1D or 2D spectra.

Arguments: option can be any of the following:
- 'alternate' is a keyword specifying the alternate reference phase (for phase sensitive gradient 2D data).
- 'autophase' is a keyword specifying to automatically adjust the phase of the reference signal.
- 'displaycf' is a keyword specifying to stop at the display of the correction function.
- 'fittedbaseline' is a keyword specifying to use cubic spline baseline correction defined by the choice of integral regions.
- 'invert' is a keyword specifying to invert the corrected difference spectrum/spectra.
- 'noaph' is a keyword specifying not to automatically adjust zero order phase of the reference region.
- 'nodc' is a keyword specifying not to use dc correction of reference region.
- 'noextrap' is a keyword specifying not to use extrapolated dispersion mode.
- 'nohilbert' is a keyword specifying not to use Hilbert transform algorithm and to use extrapolated dispersion mode reference signal unless 'noextrap' is also used as an option.
- 'normalise' is a keyword specifying to keep corrected spectrum integrals equal to that of the first spectrum.
- 'satellites' is a keyword specifying to use satellites defined in file in ideal reference region; file should be in /vnmr/satellites, and should immediately follow 'satellites' in the argument list.
- 'stop1' is a keyword specifying to stop at display of experimental reference FID.
- 'stop2' is a keyword specifying to stop at display of correction function.
- 'stop3' is a keyword specifying to stop at display of corrected FID.
- 'stop4' is a keyword specifying to stop at display of first corrected FID.
- 'verbose' is a specifying keyword to display information about processing in the main window.
- 'writecf' is a keyword specifying to write the correction function to file; the argument file must immediately follow 'writecf'.
- 'writefid' is a keyword specifying to write out corrected FID to file; if file does not begin with /, it is assumed to be in the current working directory. In the argument list, file should immediately follow 'writefid'.

file is the name of the file used with the 'satellites' and 'writefid' options.

start and finish are the indices of the first and last array elements to be processed; increment specifies the steps in which the index is to be incremented. The default is to process all the transformed spectra in an array.

See also: VnmrJ Liquids NMR

Related: fiddled
- Perform reference deconvolution subtracting alternate FIDs

fiddleu
- Perform reference deconvolution subtracting successive FIDs
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
Related: fiddle (Perform reference deconvolution)
**fidpar**

**Add parameters for FID display in current experiment (M)**

**Description:** Creates the FID display parameters `axisf`, `crf`, `deltaf`, `dotflag`, `vpf`, and `vpfi` in the current experiment. Use `fidpar` to define these parameters in old parameter sets (they are already defined in new parameter sets).

**See also:** *VnmrJ Liquids NMR*

**Related:**
- `addpar` Add selected parameters to current experiment (M)
- `axisf` Axis label for FID displays and plots (P)
- `crf` Current time domain cursor position (P)
- `deltaf` Difference of two time cursors (P)
- `dotflag` Display FID as connected dots (P)
- `vpf` Current vertical position of FID (P)
- `vpfi` Current vertical position of imaginary FID (P)

**fidsave**

**Save data (M)**

**Description:** Macro to save data. It uses `svfdir` and `svfname` to construct the data filename.

**fifolpsize**

**FIFO loop size (P)**

**Applicability:** All systems except *MERCURYplus/-VX*.

**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 CONFIG window (opened by `config`).

**Values:** 2048

**See also:** *VnmrJ Installation and Administration*

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

**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)

**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:** *VnmrJ Liquids NMR*

**Related:**
- `auto` Automation mode active (P)
- `go` Submit experiment to acquisition (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 VnmrJ Liquids NMR 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 file information for files display (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: files  Interactively handle files (C)

filter  Gaussian low-pass filter for image processing (M)

Applicability: Systems with imaging capabilities.

Syntax:  filter(strength)
Description: Sets the processing parameters gf, gfs, gf1, and gfs1 to create a low-pass filter for improving the signal-to-noise (S/N) ratio in images. S/N improvement is achieved at the expense of resolution. The results of the parameter setting performed by filter can be applied to the image using the wft2d command. The parameters gf, gfs, gf1, and gfs1 are calculated to center the filter in both the t1 and t2 dimensions. The filter setting can be bypassed with the ft2d command. A side effect of filter is to reset the maximum, minimum and step values for all of the Gaussian processing parameters. This is to allow precise setting of the filter.

Arguments: strength is a number from 0 to 100 that represents the attenuation, in dB, applied to the signal at the edges of the sampling windows in the t1 and t2 dimensions. For example, strength set to 6 produces a Gaussian filter for t1 and t2 that reduces the signal at the edge of the sampling window by half (i.e., a 6-dB attenuation). If strength is set to 0, gf, gfs, gf1, and gfs1 are set to 'n', effective turning the parameters off.

Examples: filter(10)
See also: VnmrJ Imaging NMR
Related: ft2d Fourier transform 2D data (C)
gf Gaussian function on directly detected dimension (P)
gf1 Gaussian function on 1st indirectly detected dimension (P)
gfs Gaussian shift constant on directly detected dimension (P)
gfs1 Gaussian shift constant on 1st indirectly detected dimension (P)
wft2d Weight and Fourier transform 2D data (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)
oversamp Oversampling factor for acquisition (P)
pards Create additional parameters used for downsampling (M)
paros Create additional parameters used for oversampling (M)
**fitplot**

**Adjust plot parameters (M)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** fitplot

**Description:** If the parameter `axis` is set to 'cc', fitplot uses an algorithm that adjusts the display and subsequent plot to present the image in the largest possible format for the current conditions specified by the `wcmax`, `wc2max`, and `trace` parameters. For example, fitplot could be entered as `fitplot imageprint page` for plotting. This algorithm leaves a column of 50 mm for plotting parameters down the left-hand edge of the paper. fitplot also has other algorithms for different settings of the `axis` and `ni` parameters.

**See also:** VnmrJ Imaging NMR

**Related:**
- axis: Axis labels for displays and plots (P)
- imageprint: Plot noninteractive gray scale image (M)
- ni: Number of increments in 1st indirectly detected dimension (P)
- page: Submit plot and change plotter page (C)
- trace: Mode for n-dimensional data display (P)
- wcmax: Maximum width of chart (P)
- wc2max: Maximum width of chart in second direction (P)

**fitspec**

**Perform spectrum deconvolution (C, U)**

**Syntax:** (From VnmrJ) `fitspec<(<'usell'><,><'setsfreq'>)>` (From UNIX) `fitspec`

**Description:** Fits experimental data to Lorentzian and/or Gaussian lineshapes. `fitspec` uses as a starting point data in a file `fitspec.inpar`, which must be prepared prior to performing the calculation. This file contains the frequency, intensity, linewidth, and (optionally) the Gaussian fraction of the lineshape. Any number followed by an asterisk (*) is held fixed during the calculation; all other parameters are varied to obtain the best fit. `fitspec` creates a file `fitspec.data`, which is a text representation of the spectral data (that part of the spectrum between `sp` and `sp+wp`). After the calculation is finished, the results of the fit are contained in a file `fitspec.outpar`, with a format identical to `fitspec.inpar`.

It is often useful to use the output from a deconvolution as the input to a spin simulation to ensure the most accurate possible frequencies for the spin simulation calculation. For this reason, the frequencies and amplitudes of the calculated lines in a deconvolution are automatically stored in the parameters `slfreq` and `llamp`, respectively, from where they can serve as input to an iterative spin simulation. If the spin system is defined after a deconvolution is performed, this information is lost (`slfreq` is reset). In this case, `fitspec('setslfreq')` can be used to copy the information from `fitspec.outpar` back into `slfreq`. This is not necessary if you define the spin system before performing the deconvolution (you need not perform the entire spin simulation, only define the spin system).

**Arguments:**
- `usell` is a keyword to prepare the file `fitspec.inpar` from the last line listing (stored in `llfrq` and `llamp`). All lines are set to have a linewidth of `slw` and a fixed Gaussian fraction of 0. If another starting point is desired, this file can be edited with a text editor. Alternatively, the macro `usemark` may be used.

- `setslfreq` is a keyword to copy the information from the file `fitspec.outpar` back into the parameters `slfreq`. 

- `setsfreq` is a keyword to copy the information from the file `fitspec.outpar` back into the parameters `slfreq`. 

Examples: fitspec
fitspec('usell')
fitspec('setslfreq')

See also: *VnmrJ Liquids NMR*

**fixpar**  
**Correct parameter characteristics in experiment (M)**

**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.

**fixpar**
**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).

**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)

**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 by the macro. `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).

**Related:** `fixpar` Correct parameter characteristics in experiment (M)
`fixpar3rf` Create parameters for third rf channel (M)
`numrfch` Number of rf channels (P)
**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).

**Related:**  
- `fixpar`: Correct parameter characteristics in experiment (M)  
- `fixpar4rf`: Create parameters for fourth rf channel (M)  
- `numrfch`: Number of rf channels (P)

**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 (C)  

**Syntax:** `flashc(<'nf'>,'ms'|'mi'|'rare',ns,traces,echoes)`  

**Description:** Converts 2D FID data files from compressed formats (`seqcon='nncsn'`, `seqcon='nccnn'`, `seqcon='nnccn'`) to standard format (`seqcon='ncsnn'`) or from standard format to compressed format. Compressed data is taken by using the `nf` parameter; that is, compressed data is acquired as one large uninterrupted “multiFID” acquisition. `flashc` reads the file `fid` in the `acqfil` subdirectory of the current experiment. `flashc` can convert a compressed-compressed multislice, multi-echo, or multi-image sequence. It can also convert a “rare” type sequence with a compressed phase-encode echo train. `flashc` changes the values of the following parameters:  

**Compressed-compressed or standard format to compressed format**  
- **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.  
- **arrayelements** is set to 1 if no parameters were arrayed during data acquisition or to 2 if any parameter was arrayed during data acquisition.  

**Compressed format to standard format**  
- **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.
**f**

- `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.

Arguments: `nf` is the number of FIDs in the second dimension of a 2D experiment. When converting data in the standard format to a compressed format, `nf` must always be the first argument.

When converting compressed-compressed or “rare” type sequences, the first argument must be a string defining the type of compression:

- 'mi' is a keyword for the multi-image type of compression.
- 'ms' is a keyword for the multislice type of compression.
- 'rare' is a keyword for the “rare” multiecho, rare type, fast-imaging data sets.

(Standard to compressed) `ns` is the number of images slices or array elements to be retained.

(Compressed-compressed or rare to standard) `traces` is the number of compressed traces to retain for each `ni`. The parameter `nf` is set to this number after `flashc` has run.

(Compressed-compressed or rare to standard) `echoes` is the number of compressed echoes, used with “rare” type formatting.

Examples:

Compressed-compressed or standard format to compressed format

```
flashc('nf') (standard to compressed)
```

```
flashc('nf','ms',ns) (compressed phase-encode and multislice)
```

```
flashc('nf','mi',ns) (compressed multi-image and phase-encode)
```

Compressed-compressed format or rare format to standard format

```
flashc (simple compressed phase-encode)
```

```
flashc('ms',ns) (compressed phase-encode and multislice)
```

```
flashc('mi',ns) (compressed multi-image and phase-encode)
```

```
flashc('rare',ns,etl)
```

See also: *VnmrJ Imaging NMR*

**flipflop**

Set up parameters for FLIPFLOP pulse sequence (M)

Applicability: Systems with solids module. Sequence is not supplied on *MERCURYplus/Vx*.

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*

**fliplist**

Standard flip angle list (P)

Applicability: Systems with imaging capabilities.

Description: Contains an array of real values defining values of the standard flip angles used for the pulses in the `plist` array (e.g., `fliplist=180,90,180`). The `nD`, `seqcon`, `plist`, `patlist`, `pwrlist`, `fliplist`, and `sslist` parameters configure a particular parameter set for an application sequence.
defined by the value of the seqfil parameter. The plist, patlist, pwrlist, fliplist, and sslist parameters provide information concerning the rf pulse and conjugate gradients used by the sequence.

See also: *VnmrJ Imaging NMR*

Related: nD Application dimension (P)
patlist Active pulse template parameter list (P)
plist Active pulse length parameter list (P)
pwrlist Active pulse power level parameter list (P)
seqcon Acquisition loop control (P)
seqfil Application object code name (P)
sslist Conjugate gradient list (P)

**Fluorine**

**Set up parameters for 19F experiment (M)**

Description: Set Up parameters for 19F experiment.

**flush**

**Write out data in memory (C)**

Description: Writes out the current data and parameters in memory buffers. Normally, this information is not written to disk until exiting VnmrJ or joining another experiment. One reason to use flush is to be able to access experimental data from a program separate from the VnmrJ program.

See also: *User Programming*

**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 f2 dimension in 2D data sets, the f3 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 f1 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: *VnmrJ Liquids NMR*

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)
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 \( \text{ni2} \). \( \text{fn2} \) is set in a manner analogous to the parameter \( \text{fn} \), with \( \text{np} \) being substituted by \( 2 \times \text{ni2} \).

**See also:** *VnmrJ Liquids NMR*

**Related:**
- \( \text{fn} \) Fourier number in directly detected dimension (P)
- \( \text{fn1} \) Fourier number in 1st indirectly detected dimension (P)
- \( \text{ni2} \) Number of increments in 2nd indirectly detected dimension (P)
- \( \text{np} \) Number of data points (P)

Fourier number to build up 2D DOSY display in freq. domain (P)

**Description:** In 2D DOSY sequences (Dbppste, DgcsteSL, Doneshot, Dbppsteinet), replaces \( \text{fn} \) when setting up the 2D display.

**See also:** *VnmrJ Liquids NMR*

**Related:**
- \( \text{ddif} \) Synthesize and display DOSY plot (C)
- \( \text{dosy} \) Process DOSY experiments (M)

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*

Fold INADEQUATE data about two-quantum axis (C)

**Syntax:** foldcc

**Description:** Symmetrizes 2D INADEQUATE data along the P-type double-quantum axis and applies an automatic \( \text{dc} \) baseline correction. \( \text{foldcc} \) functions for both hypercomplex and complex 2D data.

**See also:** *VnmrJ Liquids NMR*

**Related:**
- \( \text{dc} \) Calculate spectral drift correction (C)
- \( \text{foldj} \) Fold J-resolved 2D spectrum about \( f_1=0 \) axis (C)
- \( \text{foldt} \) Fold COSY-like spectrum along diagonal axis (C)
- \( \text{rotate} \) Rotate 2D data (C)

Fold J-resolved 2D spectrum about \( f_1=0 \) axis (C)

**Description:** Symmetrizes heteronuclear 2D-J, or rotated homonuclear 2D-J, experiments about the \( f_1=0 \) axis. The \( \text{foldj} \) command functions with both complex and hypercomplex 2D data.

**Related:**
- \( \text{foldcc} \) Fold INADEQUATE data about 2-quantum axis (C)
- \( \text{foldt} \) Fold COSY-like spectrum along diagonal axis (C)
- \( \text{rotate} \) Rotate 2D data (C)

Fold COSY-like spectrum along diagonal axis (C)

**Syntax:** foldt<('symm'|'triang')>

**Description:** Folds COSY-like correlation spectra about the diagonal. The 2D spectrum must exhibit a *P-type diagonal* for \( \text{foldt} \) to work properly (a P-type diagonal goes
foldt functions for both hypercomplex and complex 2D data but requires that \( fn = fn1 \) and \( sw = sw1 \).

Arguments:

- `symm` is a keyword for the folding process to perform a symmetrization of the data by replacing every two symmetry-related points with the one point therein that has the least magnitude. This value is the default.
- `triang` is a keyword for the folding process to perform a triangularization of the data by replacing every two symmetry-related points with their geometric mean.

Related:

- \( fn \): Fourier number in directly detected dimension (P)
- \( fn1 \): Fourier number in 1st indirectly detected dimension (P)
- \( foldcc \): Fold INADEQUATE data about 2-quantum axis (C)
- \( foldj \): Fold J-resolved 2D spectrum about \( f_j = 0 \) axis (C)
- \( rotate \): Rotate 2D data (C)
- \( sw \): Spectral width in directly detected dimension (P)
- \( sw1 \): Spectral width in 1st indirectly detected dimension (P)

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: VnmrJ Liquids NMR

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  
**Format a real number or convert a string for output (C)**

Syntax:

1. `format(real_number,length,precision):return`  
2. `format(string,'upper'|'lower'|'isreal'):return`

Description: Using syntax 1, `format` takes a real number or real type variable and formats it into a string with given length and precision and rounds it off if necessary (see examples 1 to 4 below). `format` can also be used to format a real type variable as a real number (see example 5).

Using syntax 2, `format` converts a string variable into a new string of characters either all upper case or all lowercase (see examples 6 and 7) or tests the string to determine if it represents a real number (see example 8).

Arguments:

- `real_number` is the real type variable containing the value to be formatted.
- `length` is the length of for formatted real number. If `length` is set to 0, just enough places are used to hold the number.
- `precision` is the precision (i.e., the number of places to the right of the decimal point) of the formatted real number. If `precision` is set to 0, output is an integer.
- `string` is the string variable to be converted into upper or lower case.
- `upper` is a keyword to convert the string variable given by `string` into all upper case characters.
- `lower` is a keyword to convert `string` into all lower case characters.

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)
'isreal' is a keyword that tests the first argument to verify that the argument satisfies the rules for a real number. When given, format returns a 1 in the first argument and can represent a real number and a zero otherwise.

return is the return string variable, real number, or integer.

Examples:
1. format(a,5,2):n1 If a=24.1264 then n1='24.13'
2. format(a,9,4):n2 If a=24.1264 then n2='24.126'
3. format(a,0,3):n3 If a=24.1264 then n3='24.126'
4. format(a,2,0):n1 If a=24.1264 then n1='24'
5. format(a,2,0):r1 If a=24.1264 then r1=24
6. format(solvent,'upper'):n2 If solvent='CDCl3' then n2='cdcl3'
7. format(solvent,'lower'):n3 If solvent='CDCl3' then n3='CDCDL3'
8. format($1,'isreal'):$a If $1=1 then $a=1

See also: User Programming

Related: n1,n2,n3 Name storage for macros (P)
r1–r7 Real-value storage for macros (P)

**fp**  
Find peak heights or phases (C)

**Syntax:** fp(<'phase'>,<index1,index2,...>)

**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)
fpphase'

See also: VnmrJ Liquids NMR

Related: dll Display listed line frequencies and intensities (C)
fn Fourier number in directly detected dimension (P)
gett 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)

**fpmult**  
First point multiplier for np FID data (P)

**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 “f2 ridges.” In 3D experiments, the ridges appear as “f3 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 \texttt{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 \texttt{fpmult} is set to 'n', \texttt{fpmult} takes on its default value.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{fpmult1} \textit{First point multiplier for ni interferogram data (P)} \texttt{fpmult2} \textit{First point multiplier for ni2 interferogram data (P)} \texttt{np} \textit{Number of data points (P)} \texttt{trace} \textit{Mode for n-dimensional data display (P)} \texttt{wft2da} \textit{Weight and Fourier transform phase-sensitive data (M)}

\textbf{fpmult1} \textit{First point multiplier for ni interferogram data (P)}

\textbf{Description:} Operates on \textit{ni} hypercomplex or complex interferogram data in a manner analogous to \texttt{fpmult}. In many 2D experiments, the \textit{t1} values are adjusted so there is no first-order phasing in the \textit{f1} and \textit{f2} dimensions. In this case, \texttt{fpmult1} should be 0.5. If the \textit{t1} value is adjusted so that there is a 180° first-order phase correction, \texttt{fpmult1} should be 1.0.

Values: Default value is 0.5. If \texttt{fpmult1} is set to 'n', it takes on its default value.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{fpmult} \textit{First point multiplier for np FID data (P)} \texttt{fpmult2} \textit{First point multiplier for ni2 interferogram data (P)} \texttt{ni} \textit{Number of increments in 1st indirectly detected dimension (P)}

\textbf{fpmult2} \textit{First point multiplier for ni2 interferogram data (P)}

\textbf{Description:} Operates on \textit{ni2} hypercomplex or complex interferogram data in a manner analogous to \texttt{fpmult}. In many 3D experiments, the \textit{t2} value is adjusted so that there is no first-order phasing in the \textit{f1} and \textit{f2} dimensions. In this case, \texttt{fpmult2} should be 0.5. If the \textit{t2} value is adjusted so that there is a 180° first-order phase correction, \texttt{fpmult2} should be 1.0.

Values: Default value is 0.5. If \texttt{fpmult2} is set to 'n', it takes on its default value.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{fpmult} \textit{First point multiplier for np FID data (P)} \texttt{fpmult1} \textit{First point multiplier for ni interferogram data (P)} \texttt{ni2} \textit{Number of increments in 2nd indirectly detected dimension (P)}

\textbf{fr} \textit{Full recall of a display parameter set (M)}

\textbf{Syntax:} (1) \texttt{frset\_number} \\
(2) \texttt{fr(set\_number)}

\textbf{Description:} Performs a full recall of a display parameter set, setting all parameters to exactly as they were when the corresponding \texttt{s} command was entered.

\textbf{Arguments:} \texttt{set\_number} is the number of the display parameter set.

\textbf{Examples:} \texttt{fr2} \\
\texttt{fr(3)}

\textbf{Related:} \texttt{r} \textit{Recall display parameter set (M)} \texttt{s} \textit{Save display parameters as a set (M)}
**fread**  
Read parameters from file and load them into a tree (C)

**Syntax:**  
fread(file<,tree<,'reset'|'value'>>)

**Description:**  
Reads parameters from a file and loads the parameters into a tree. The tree can be global, current, processed, or systemglobal. fread can read from any file that has parameters stored in the correct VnmrJ format.

Note that if parameters are read into the global tree, certain important system parameters are not loaded because these parameters should not be changed. The parameters that are not loaded are userdir, systemdir, curexp, autodir, auto, vnmraddr, and acqaddr.

**Arguments:**
- **file** is the name of the file containing parameters stored in VnmrJ format.
- **tree** is one of the keywords 'global', 'current', 'processed', or 'systemglobal'. The default is 'current'. This argument specifies the type of tree into which the parameters are loaded. Refer to the create command for more information on types of trees.
- 'reset' is a keyword that causes the parameter tree to be cleared before the new parameter file is read. Without this option, parameters read from a file are added to the existing preloaded parameters. To use this option, tree must also be specified.
- 'value' is a keyword that causes only the values of the parameters 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, and enumerated values are not changed. To use this option, tree must also be specified.

**Examples:**
- fread('/vnmr/stdpar/H1.par/procpar')
- fread('sampvar','global')
- fread('setvar','current','reset')
- fread('var1','processed','value')

**See also:** User Programming

**Related:**
- auto  
  Automation mode active (P)
- autodir  
  Automation directory absolute path (P)
- create  
  Create new parameter in a parameter tree (C)
- curexp  
  Current experiment directory (P)
- destroy  
  Destroy a parameter (C)
- display  
  Display parameters and their attributes (C)
- fsave  
  Save parameters from a tree to a file (C)
- rtp  
  Retrieve parameters (C)
- systemdir  
  System directory (P)
- userdir  
  User directory (P)

**fsave**  
Save parameters from a tree to a file (C)

**Syntax:**  
fsave(file<,tree>)

**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)
fsq

**Frequency-shifted quadrature detection (P)**

**Description:** Selects whether to use frequency-shifted quadrature detection. When $fsq$ is turned on, if $dasp$ is on, the observe frequency is offset by $oslsfrq$, and the digital filter is also offset by $oslsfrq$. The default value of $oslsfrq$ is $1.25*sw$.

On MERCURYplus/Vx frequency-shifted quadrature detection can be done using inline DSP. 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 *VnmrJ Liquids NMR*.

**Values:**
- 'n' turns frequency-shifted quadrature detection off.
- 'y' turns it on.

**See also:** *VnmrJ Liquids NMR*

**Related:**
- $dasp$ 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)`

**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.
  - 'nodc' is a keyword to not perform the usual FID 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: VnmrJ Liquids NMR

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)

proc Type of processing on the np FID (P)

phfid Zero-order phasing constant for 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)

ft1d Fourier transform along f2 dimension (C)

Syntax:

(1) ft1d(element_number)
(2) ft1d<('nf',element_number)
(3) ft1d<(options,><coefficients>)>

Description: Performs the first Fourier transformation along the f2 dimension, without weighting, and matrix transposition. ft1d allows the display of t1 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 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 \texttt{lsfid}, \texttt{lsfid1}, \texttt{lsfid2}, \texttt{phfid}, \texttt{phfid1}, \texttt{phfid2}, \texttt{lsfrq}, \texttt{lsfrq1}, and \texttt{lsfrq2}, as appropriate. For information on the \texttt{lf}s (low-frequency suppression) and \texttt{zf}s (zero-frequency suppression) solvent suppression options, see the description of the parameters \texttt{ssfilter} and \texttt{ssorder}, and the macro \texttt{parfidss}.

Arguments: \texttt{element\_number} is a single array element to be weighted and transformed.

\texttt{options} can be the keywords \texttt{ptype}' or \texttt{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 \texttt{ntype}'.

\texttt{coefficients} are a series of coefficients according to the following scheme: \texttt{RR1} is the coefficient used to multiply the real part (\texttt{R}) of spectra set 1 before it is added to the real part (\texttt{R}) of the interferogram. \texttt{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.

\begin{verbatim}
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)

... RR1*REAL(w2,element=1) –> IMAG(t1)
II1*IMAG(w2,element=1) –> + IMAG(t1)
RR2*REAL(w2,element=2) –> + IMAG(t1)
II2*IMAG(w2,element=2) –> + IMAG(t1)

...
\end{verbatim}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{dconi} Interactive 2D data display (C)
\texttt{ft2d} Fourier transform 2D data (C)
\texttt{lsfid} Number of complex points to left-shift \texttt{np} FID (P)
\texttt{lsfid1} Number of complex points to left-shift \texttt{ni} interferogram (P)
\texttt{lsfid2} Number of complex points to left-shift \texttt{ni2} interferogram (P)
\texttt{lsfrq} Frequency shift of the \texttt{fn} spectrum (P)
\texttt{lsfrq1} Frequency shift of the \texttt{fn1} spectrum (P)
\texttt{lsfrq2} Frequency shift of the \texttt{fn2} spectrum (P)
\texttt{parfidss} Create parameters for time-domain solvent subtraction (M)
\texttt{phfid} Zero-order phasing constant for \texttt{np} FID (P)
\texttt{phfid1} Zero-order phasing constant for \texttt{ni} interferogram (P)
\texttt{phfid2} Zero-order phasing constant for \texttt{ni2} interferogram (P)
\texttt{proc} Type of processing on \texttt{np} FID (P)
\texttt{proc1} Type of processing on \texttt{ni} interferogram (P)
\texttt{proc2} Type of processing on \texttt{ni2} interferogram (P)
\texttt{pmode} Processing mode for 2D data (P)
\texttt{ssorder} Order of polynomial to fit digitally filtered FID (P)
\texttt{ssfilter} Full bandwidth of digital filter to yield a filtered FID (P)
\texttt{wft2d} Weight and Fourier transform 2D data (C)

\texttt{ft1da} Fourier transform phase-sensitive data (M)

Syntax: \texttt{ft1da\{\textbf{options}\}}
Description: Performs the first (f2) 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.

Arguments: options are the same as used with ft2da. See ft2da for details.

See also: VnmrJ Liquids NMR

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>)

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 f1 quadrature or with f1 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)

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>)

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 (t1) 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 f1 (ni dimension) and f2 (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 t1-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 $2n$ 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:

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Coefficient order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RR1, IR1, RI1, II1</td>
</tr>
<tr>
<td>2</td>
<td>RR1, IR1, RR2, IR2, RI1, II1, RI2, II2</td>
</tr>
<tr>
<td>3</td>
<td>RR1, IR1, RR2, IR2, RR3, IR3, RI1, II1, RI2, II2, RI3, II3</td>
</tr>
</tbody>
</table>

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 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 t2 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 t1 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 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 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, 6) are transformed along t1; the odd numbered regions are not transformed along t1.
- '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.

\[
\text{ft2d}(RR1, IR1, RR2, IR2, \ldots, RI1, II1, RI2, II2, \ldots)
\]

where:

\[
\begin{align*}
RR1 \times \text{REAL}(w2, \text{element}=1) & \rightarrow \text{REAL}(t1) \\
IR1 \times \text{IMAG}(w2, \text{element}=1) & \rightarrow + \text{REAL}(t1) \\
RR2 \times \text{REAL}(w2, \text{element}=2) & \rightarrow + \text{REAL}(t1) \\
IR2 \times \text{IMAG}(w2, \text{element}=2) & \rightarrow + \text{REAL}(t1) \\
\ldots \\
RI1 \times \text{REAL}(w2, \text{element}=1) & \rightarrow \text{IMAG}(t1) \\
II1 \times \text{IMAG}(w2, \text{element}=1) & \rightarrow + \text{IMAG}(t1) \\
RI2 \times \text{REAL}(w2, \text{element}=2) & \rightarrow + \text{IMAG}(t1) \\
II2 \times \text{IMAG}(w2, \text{element}=2) & \rightarrow + \text{IMAG}(t1)
\end{align*}
\]

'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:

\[
\begin{align*}
\text{ft2d}(1,0,0,0,0,0,1,0) \\
\text{ft2d}(1) \\
\text{ft2d}('nf',3) \\
\text{ft2d}('ptype',\ldots)
\end{align*}
\]

See also: VnmrJ Liquids NMR

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 f2 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)
- lsfrql Frequency shift of the fn1 spectrum (P)
- lsfrq2 Frequency shift of the fn2 spectrum (P)
- parfids 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)
F

ft2da  Fourier transform phase-sensitive data (M)

Syntax:  ft2da<(options)>

Description: Processes 2D FID data and 2D planes at particular t₁ or t₂ 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 f₂ Fourier transform.

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 hypercomplex and TPPI modes of data acquisition in t₁ and t₂ can be handled. For selective f₂f₃ processing, the numeric argument immediately following the 'ni2' keyword is interpreted to be the t₁ increment number, which specifies the particular f₂f₃ plane (plane_number, see below) to be processed. For selective f₁f₃ processing, the t₂ 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 f₂ spectra prior to the f₁ Fourier transform. The baseline regions must have been previously determined. The default polynomial order is 1, which leads to a spline fit. A different polynomial order can be specified by inserting a numerical argument following 'bc'.
- 'dc' is a keyword for a drift correction (dc) of the f₂ spectra prior to the f₁ 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: \textit{VnmrJ Liquids NMR}

\textbf{ft2dac} \hspace{2cm} \textit{Combine arrayed 2D FID matrices (M)}

\textbf{Syntax:} \texttt{ft2dac}(<\texttt{mult1},<\texttt{mult2},...<\texttt{multn}>)

\textbf{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 \( f1 \) quadrature or with \( f1 \) quadrature using the TPPI method. \texttt{ft2dac} is used with TOCSY (with multiple mixing times).

\textbf{Arguments:} \( \texttt{mult1},\texttt{mult2},...\texttt{multn} \) are multiplicative coefficients. The \( n \)th argument is a real number and specifies the coefficient for the \( n \)th 2D FID matrix.

\textbf{Related:} \texttt{ft1dac} \textit{Combine arrayed 2D FID matrices (M)} \hspace{2cm} \texttt{tocsy} \textit{Set up parameters for a TOCSY pulse sequence (M)}

\textbf{ft3d} \hspace{2cm} \textit{Perform a 3D Fourier transform on a 3D FID data set (M,U)}

\textbf{Syntax:} \texttt{(From VnmrJ) ft3d}(<\texttt{data\_directory}>,<\texttt{number\_files}>\texttt{,<nocoef'>,<t\_1t\_2'>,<f\_df'>,<nof\_df'>\texttt{,<plane\_type}>)\texttt{>}

\textbf{Description:} Transforms 3D FID data into 3D spectral data. \texttt{ft3d} can be entered from a macro or directly from UNIX. Each type of entry is described below. A final section explains the \texttt{ft3d} coefficient file.

Additional parameter control for the operation of \texttt{ft3d} is available. This allows drift corrections and partial Fourier transformation. See the descriptions of \texttt{specdc3d}, \texttt{fiddc3d}, and \texttt{ptspec3d} for information.

The 3D FID data must be loaded into the experiment in which the \texttt{ft3d} macro is to be run. \texttt{ft3d} is started up in background mode by this macro so that VnmrJ remains free for interactive processing. You 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 VnmrJ.

Distributed $f_1f_2$ processing has the following system and network requirements:

- The system on which the macro `ft3d` is executed from within VnmrJ must define the names of the networked computers that are to participate in the distributed processing. The file `/etc/hosts.3D` must contain these names in the following format:

  ```
  unity1
  unity2
  datastation1
  datastation2
  ```

- Each participating computer must recognize the name of the user that started up the master `ft3d` program as a valid user name on its system. For example, if user steve issues the `ft3d` command within VnmrJ running on computer `unity0`, steve must be a valid user on all other computer systems that are to be used in the distributed $f_1f_2$ processing.
- Each computer system must have NFS access to the 3D data directory.

**Arguments:** The order of the arguments is not important.

- `data_directory` (without the `/data` subdirectory appended) specifies the output directory for the 3D spectral data file(s). The default directory for the 3D spectral data is `curexp/datadir3d`.
- `number_files` sets the number of 3D data files (`data1, data2, ..., dataN`, where `N` is `number_files`) used to store the transformed 3D data. `number_files` must be an integer and be 32 or less. When `number_files` is entered, distributed $f_1f_2$ processing is performed by `ft3d` if possible.
- `nocoeff` is a keyword for the `set3dproc` command within the `ft3d` macro to not create a 3D coefficient file prior to invoking the `ft3d` program. This option is useful if you have modified an existing 3D coefficient file and do not want it to be overwritten prior to the 3D transform. See below for information on coefficient files. By default, `ft3d` calls the `make3dcoef` macro to create a coefficient file using the `f1coef` and `f2coef` string parameter values.
- `$t1t2$` and `$t2t1$` are keywords to explicitly define the order of the $t_1$ and $t_2$ arrays (other than $ni$ and $ni2$). By default, `ft3d` looks at the `array` parameter and if any parameter other than `phase` and `phase2` are arrayed, the macro aborts.
- `$fdf$` indicates that the output of `ft3d` is to be an FDF (Flexible Data Format) file named `data.fdf`. This is the default if the parameter `appmode` is set to `imaging`. Distributed processing can still be performed if `number_files` is set appropriately. 3D FDF files can be viewed with the `disp3d` program, or selected slices can be extracted with `ImageBrowser` (started by the `browser` command from UNIX).
- `$nofdf$` indicates that the final output is the group of `data1, data2, ...` files, and that no FDF format file should be produced. This is the default if the parameter `appmode` is not set to `imaging`.

`plane_type` sets plane extraction following the complete 3D FT with the following keywords:

- `$xall$` indicates that all three 2D plane types, $f_1f_3$, $f_2f_3$, and $f_1f_2$, are to be automatically extracted at the end of the 3D Fourier transform.
- `$f1f3$`, `$f2f3$`, and `$f1f2$` can be used to select any combination of plane types to be extracted.
Any of these options can be submitted more than once to the \texttt{ft3d} program, but the \texttt{getplane} program will display an error and abort if any one plane type is defined for extraction more than once.

Examples:

\begin{verbatim}
ft3d
ft3d('nocoeff', 'f1f3', 'f2f3')
\end{verbatim}

\texttt{ft3d} Entered from UNIX

(From UNIX) \texttt{ft3d -e exp_number -f -r <options>}

The \texttt{ft3d} program can also be run directly from the UNIX environment on the host computer. An information file must be present before \texttt{ft3d} can execute successfully but it need contain only valid processing information for the \texttt{t3} dimension and valid Fourier numbers for the \texttt{t1} and \texttt{t2} transforms. Valid weighting and phasing parameters for the \texttt{f1} and \texttt{f2} dimension do not need to be set while \texttt{wftt3} executes. After several FIDs have been collected, you can determine acceptable \texttt{f1} weighting and phasing parameters. After setting \texttt{fn1} and \texttt{fn2} to the desired values, the 3D processing information file can be created by typing \texttt{set3dproc} in the VnmrJ command line. At that point, the next invocation of \texttt{ft3d} by the macro \texttt{wftt3} causes all \texttt{(t1,t2)} increment sets up to and including the current increment in \texttt{t3} to be processed.

To start \texttt{ft3d} on a remote computer running as a data station for the system, log in as \texttt{root} and enter one of the following commands so that the master \texttt{ft3d} program can properly communicate with the computer:

- On \textit{UNITY/INOVA} systems, enter \\
  \texttt{/vnmr/acqbin/Infoprc &}

With the \texttt{Infoprc} or \texttt{acqinfo\_svc} program running, enter \texttt{ft3d} with the \texttt{-h} option and the necessary arguments. The \texttt{ft3d} program invoked with the \texttt{-h} option is considered to be the master program and is responsible for spawning additional remote \texttt{ft3d} processes.

Each remote computer must be able to access the 3D data directory as if it were stored on a local disk, must recognize the user name under which the master \texttt{ft3d} program is being run, and must also have permission to read from and write to that directory. If the 3D data directory contains four \texttt{f3} transformed data files (\texttt{data1–data4}), the master \texttt{ft3d} program uses the first three remote computer systems listed in file \texttt{hosts.3D} that respond.

If the multihost processing option is selected, the number of computers involved will be no more than the number of sets the \texttt{f3} spectral data is partitioned into. This number is selected with the \texttt{-m} option (see below).

If you are unsure of whether to use \texttt{Infoprc} or \texttt{acqinfo\_svc} on the remote computer, change directories to \texttt{/vnmr/acqbin}, enter \texttt{lf}, and check which program is present.

Note that if the host computer is rebooted, the background command (\texttt{Infoprc} or \texttt{acqinfo\_svc}) has to be entered again.

Arguments:

Note that entering \texttt{ft3d} with an ampersand (\&) after the arguments makes the command execute in the background. As a result, the UNIX prompt reappears after the command is entered and further commands can be entered and executed while the \texttt{ft3d} command is processing.

\begin{itemize}
  \item \texttt{-e exp\_number} is the experiment number where 3D processing is to occur. This argument is required. It must be written as a minus sign, the letter \texttt{e}, a space, and a valid experiment number from 1 to 9 (e.g., \texttt{-e 3} sets experiment 3). The experiment must already exist.
\end{itemize}

The following two options should always be set for reliable operation:

\begin{itemize}
  \item \texttt{-f} specifies that any existing 3D data sets in the experiment should be deleted. This option requires no additional value.
  \item \texttt{-r} calls for explicit data reduction after the 3D Fourier transform. Data reduction consists of retaining only the “real-real-real” part of the
completely transformed 3D data set. The -r option is mandatory and is enforced within ft3d regardless of the user command line input.

Options can be any of the following:

- **-F header_file** indicates that an FDF (Flexible Data Format) output file should be produced, using the FDF header found in header_file. The output file will be named data.fdf, and the data1, data2, ... files will not be produced.

- **-h** selects the multihost processing option. The /etc/hosts.3D file must exist and contain the names of the remote hosts, one host name per line. Each remote host must also have either the program Infoprc or the program acqinfo_svc running in the background (one of these programs is already running on any computer being used as a spectrometer host).

- **-l** specifies that a log file be generated in the data subdirectory of the datadir3d directory.

- **-m** partitions the f3 transformed spectral data over more than one data file. This partitioning is necessary if the distributed processing capability of ft3d is to be used in performing the remaining f1 and f2 transforms. The syntax **-mnfiles** is used to specify nfiles, the number of data files into which the 3D spectral data is to be divided (e.g., -m4 specifies 4 data files). Each such data file contains an f3 subset of the f1f2 spectral planes. If nfiles is not specified, ft3d reports an error and aborts. If nfiles is less than an internally calculated value (based on memsize and the maximum size for a single 2D transform), the number of data files is set to the internally calculated value; otherwise, nfiles determines the number of data files to be used. The maximum number of such files is currently defined to be 32. These 3D data files are labeled data1, data2, ..., datan.

- **-o** specifies an alternative output directory for the processed 3D data. The default directory is datadir3d within the current experiment. A full UNIX path must follow the -o option.

- **-p** specifies the time-domain dimensions to be processed. If -p is used, the processed dimensions can be specified as f3f2f1, f3f2, f2f3, f2f1, f1f2, f3, f2, and f1. The values f3f1 and f1f3 are not allowed because processing must be done sequentially in the order f3, then f2, and then f1. If the -p option is not invoked, ft3d defaults to f3f2f1, resulting in a completely transformed 3D data set.

- **-s** specifies processing of the f3 dimension of the 3D FID data concurrently with data acquisition. In practice, concurrent f3 processing is realized by setting wnt='wftt3' in the VnmrJ parameter set and starting the 3D acquisition by entering au. The macro wftt3 handles the call to ft3d at the appropriate times during data collection.

- **-x** specifies that plane extractions be performed at the end of 3D processing. The available planes are defined as f1f2, f1f3, and f2f3. If more than one plane extraction is desired, the planes are separated by a colon. For example, -x f1f2:f1f3:f2f3 would extract all three planes. The planes are placed in the extr subdirectory of datadir3d.

Examples: (From UNIX) ft3d -r -f -l -e 2 &
(From UNIX) ft3d -r -f -l -e 2 -x f1f2:f1f3:f2f3 &
See also: VnmrJ Liquids NMR

Related: appmode Application mode (P)
browser Start ImageBrowser application (U)
dconci Interactive 2D data display (C)
disp3d Display 3D data (U)
fiddc3d 3D time-domain dc correction (P)
fc1coef Coefficient to construct F1 interferogram (P)
f2coef Coefficient to construct F2 interferogram (P)
getplane Extract planes from a 3D spectral data set (M)
killft3d Terminate any ft3d process started in an experiment (M,U)
make3dcoef Make 3D coefficients file from 2D coefficients (M)
ptspec3d Region-selective 3D processing (P)
set3dproc Set 3D processing (C)
specdc3d 3D spectral dc correction (P)
wftt3 Process f3 dimension during 3D acquisition (M)

**full**

*Set display limits for a full screen (C)*

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)
w2c 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)
w2c Width of chart in second direction (P)

**fullt**

*Set display limits for a full screen with room for traces (C)*

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 (dconci). 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)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g2pul</td>
<td>Set up pulse sequence for gradient evaluation (M)</td>
</tr>
<tr>
<td>ga</td>
<td>Submit experiment to acquisition and FT the result (M)</td>
</tr>
<tr>
<td>gain</td>
<td>Receiver gain (P)</td>
</tr>
<tr>
<td>gap</td>
<td>Find gap in the current spectrum (M)</td>
</tr>
<tr>
<td>gap</td>
<td>Slice gap (P)</td>
</tr>
<tr>
<td>gaussian</td>
<td>Set up unshifted Gaussian window function (M)</td>
</tr>
<tr>
<td>gcal</td>
<td>Gradient calibration constant (P)</td>
</tr>
<tr>
<td>gcoil</td>
<td>Current gradient coil (P)</td>
</tr>
<tr>
<td>gcosy</td>
<td>Set up pulse sequence for gradient COSY (M)</td>
</tr>
<tr>
<td>gCOSY</td>
<td>Change parameters for gCOSY experiment (M)</td>
</tr>
<tr>
<td>Gcosy</td>
<td>Convert the parameter to a gradient COSY experiment (M)</td>
</tr>
<tr>
<td>gcrush</td>
<td>Crusher gradient level (P)</td>
</tr>
<tr>
<td>gdiff</td>
<td>Diffusion gradient level (P)</td>
</tr>
<tr>
<td>Gdqcosy</td>
<td>Convert the parameter to a gradient DQCOSY experiment (M)</td>
</tr>
<tr>
<td>get1d</td>
<td>Select a 1D experiment for processing (M)</td>
</tr>
<tr>
<td>get2d</td>
<td>Select a 2D experiment for processing (M)</td>
</tr>
<tr>
<td>getActiveStacks</td>
<td>Get active overlay (C)</td>
</tr>
<tr>
<td>getCoronal</td>
<td>Get coronal overlay (C)</td>
</tr>
<tr>
<td>getDefaultSize</td>
<td>Get default FOV</td>
</tr>
<tr>
<td>getDefaultSlices</td>
<td>Get slices (C)</td>
</tr>
<tr>
<td>getDefaultStacks</td>
<td>Get overlay based on scout image (C)</td>
</tr>
<tr>
<td>getDefaultThk</td>
<td>Get slice thickness (C)</td>
</tr>
<tr>
<td>getdim</td>
<td>Return dimensionality of experiment (M)</td>
</tr>
<tr>
<td>getFile</td>
<td>Get information about directories and files (C)</td>
</tr>
<tr>
<td>getGapMode</td>
<td>Get gap mode (C)</td>
</tr>
<tr>
<td>getgcal</td>
<td>Get gcal value from table (M)</td>
</tr>
<tr>
<td>getll</td>
<td>Get intensity and line frequency of line (C)</td>
</tr>
<tr>
<td>getMilestoneStacks</td>
<td>Get overlay from saved parameters (C)</td>
</tr>
<tr>
<td>getParam</td>
<td>Retrieve parameter from probe file (M)</td>
</tr>
<tr>
<td>getPlane</td>
<td>Extract planes from a 3D spectral data set (M)</td>
</tr>
<tr>
<td>getPrevStacks</td>
<td>Start planning with previous stacks</td>
</tr>
<tr>
<td>getReg</td>
<td>Get frequency limits of a specified region (C)</td>
</tr>
<tr>
<td>getSagittal</td>
<td>Get sagittal overlay (C)</td>
</tr>
<tr>
<td>getSN</td>
<td>Get signal-to-noise estimate of a spectrum (M)</td>
</tr>
<tr>
<td>getTransverse</td>
<td>Get transverse overlay (C)</td>
</tr>
<tr>
<td>getTXT</td>
<td>Get text file from VnmrJ data file (C)</td>
</tr>
<tr>
<td>getType</td>
<td>Get the type of a variable (C)</td>
</tr>
<tr>
<td>getValue</td>
<td>Get value of parameter in a tree (C)</td>
</tr>
<tr>
<td>gf</td>
<td>Prepare parameters for FID/spectrum display in acqi (M)</td>
</tr>
<tr>
<td>gf</td>
<td>Gaussian function in directly detected dimension (P)</td>
</tr>
<tr>
<td>gf1</td>
<td>Gaussian function in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>gf2</td>
<td>Gaussian function in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>gflow</td>
<td>Flow encoding gradient level (P)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>gfs</td>
<td>Gaussian shift const. in directly detected dimension (P)</td>
</tr>
<tr>
<td>gfs1</td>
<td>Gaussian shift const. in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>gfs2</td>
<td>Gaussian shift const. in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>ghMBC</td>
<td>Change parameters for gHMBC experiment (M)</td>
</tr>
<tr>
<td>Ghmhc</td>
<td>Convert the parameter to a gradient HMBC experiment (M)</td>
</tr>
<tr>
<td>ghmqc</td>
<td>Set up PFG HMQC pulse sequence (M)</td>
</tr>
<tr>
<td>ghMQC</td>
<td>Set up parameters for gHMQC experiment (M)</td>
</tr>
<tr>
<td>Ghmqc</td>
<td>Convert the parameter to a gradient HMQC experiment (M)</td>
</tr>
<tr>
<td>ghMQC15</td>
<td>Set up parameters for $^{15}$N gHMQC experiment (M)</td>
</tr>
<tr>
<td>ghMQC_d2</td>
<td>Set up parameters for $^{15}$N gHMQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>ghMQC_d213</td>
<td>Set up parameters for $^{13}$C gHMQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>ghmqcs</td>
<td>Set up a PFG HMQC phase-sensitive pulse sequence (M)</td>
</tr>
<tr>
<td>gHMQCTOXY</td>
<td>Change parameters for gHMQCTOXY experiment (M)</td>
</tr>
<tr>
<td>ghsqc</td>
<td>Set up a PFG HSQC pulse sequence (M)</td>
</tr>
<tr>
<td>gHSQC</td>
<td>Set up parameters for gHSQC experiment (M)</td>
</tr>
<tr>
<td>Ghsqc</td>
<td>Convert the parameter to a gradient HSQC experiment (M)</td>
</tr>
<tr>
<td>gHSQC15</td>
<td>Set up parameters for $^{15}$N gHSQC experiment (M)</td>
</tr>
<tr>
<td>gHSQC_d2</td>
<td>Set up parameters for $^{15}$N gHSQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>gHSQC_d213</td>
<td>Set up parameters for $^{13}$C gHSQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>gHSQCTOXY</td>
<td>Set up parameters for gHSQCTOXY experiment (M)</td>
</tr>
<tr>
<td>Ghsqctoxy</td>
<td>Convert parameters for gradient HSQCTOXY experiment (M)</td>
</tr>
<tr>
<td>gilson</td>
<td>Open the Gilson Liquid Handler window (C)</td>
</tr>
<tr>
<td>gin</td>
<td>Return current mouse position and button values (C)</td>
</tr>
<tr>
<td>globalauto</td>
<td>Automation directory name (P)</td>
</tr>
<tr>
<td>glue</td>
<td>Create a pseudo-2D dataset (P)</td>
</tr>
<tr>
<td>gmapshim</td>
<td>Start gradient autoshimming (M)</td>
</tr>
<tr>
<td>gmapshim_au</td>
<td>Start acquisition with gradient shimming (M)</td>
</tr>
<tr>
<td>gmapsys</td>
<td>Run gradient autoshimming, set parameters, map shims (M)</td>
</tr>
<tr>
<td>gmapz</td>
<td>Get parameters and files for gmapz pulse sequence (M)</td>
</tr>
<tr>
<td>gmap_findtof</td>
<td>Gradient shimming flag to first find tof (P)</td>
</tr>
<tr>
<td>gmap_z1z4</td>
<td>Gradient shimming flag to first shim z1-z4 (P)</td>
</tr>
<tr>
<td>gmax</td>
<td>Maximum gradient strength (P)</td>
</tr>
<tr>
<td>gmqcosy</td>
<td>Set up PFG absolute-value MQF COSY parameter set (M)</td>
</tr>
<tr>
<td>gnoesy</td>
<td>Set up a PFG NOESY parameter set (M)</td>
</tr>
<tr>
<td>go</td>
<td>Submit experiment to acquisition (M)</td>
</tr>
<tr>
<td>go_</td>
<td>Pulse sequence setup macro called by go, ga, and au (M)</td>
</tr>
<tr>
<td>gpat-gpat3</td>
<td>Gradient shape (P)</td>
</tr>
<tr>
<td>gpe</td>
<td>Phase encoding gradient increment (P)</td>
</tr>
<tr>
<td>gpe2</td>
<td>2nd phase encode gradient increment</td>
</tr>
<tr>
<td>gpe3</td>
<td>3rd phase encode gradient increment</td>
</tr>
<tr>
<td>gped</td>
<td>Phase encode dephasing gradient in the EPI sequence (P)</td>
</tr>
<tr>
<td>gpeqmult</td>
<td>Phase encode gradient increment multiplier (P)</td>
</tr>
<tr>
<td>gplan</td>
<td>Start interactive image planning (C)</td>
</tr>
<tr>
<td>gradaxis</td>
<td>Gradient axis (P)</td>
</tr>
<tr>
<td>gradientdisable</td>
<td>Disable PFG gradients (P)</td>
</tr>
<tr>
<td>gradstepsz</td>
<td>Gradient step size (P)</td>
</tr>
<tr>
<td>gradtype</td>
<td>Gradients for X, Y, and Z axes (P)</td>
</tr>
</tbody>
</table>
g2pul  
Set up pulse sequence for gradient evaluation (M)

Applicability: Systems with the pulsed field gradient or imaging module.

Description: Performs gradient recovery measurements. With \texttt{gzlvl1} on during \texttt{gt1}, the system recovery to homogeneity can be measured after delay \texttt{d2}. Typical values are \texttt{gt1}=0.040 (40 ms) and gradient strength on full (\texttt{gzlvl}=32767). \texttt{g2pul} sets an experiment environment suitable for these tests. The \texttt{gradaxis} parameter is used by \texttt{g2pul} to select the x, y, or z gradient axis.

See also: User Programming

Related: \texttt{gradaxis} Select gradient axis (P)

\texttt{ga}  
Submit experiment to acquisition and FT the result (M)

Syntax: \texttt{ga\langle \langle \text{nocheck}\rangle, \langle \text{next}\rangle, \langle \text{wait}\rangle \rangle}

\texttt{graphis}  
Return the current graphics display status (C)

\texttt{grayctr}  
Gray level window adjustment (P)

\texttt{graysl}  
Gray level slope (contrast) adjustment (P)

\texttt{grecovery}  
Eddy current testing (M)

\texttt{grid}  
Draw a grid on a 2D display (M)

\texttt{griserate}  
Gradient rise rate (P)

\texttt{gro}  
Readout gradient strength (P)

\texttt{groa}  
Readout gradient adjuster in EPI experiment (P)

\texttt{grof}  
Fine tune readout gradient compensation (P)

\texttt{gropat}  
Readout gradient shape (P)

\texttt{gror}  
Read out compensation gradient (P)

\texttt{grora}  
Readout dephasing gradient adjuster in EPI experiment (P)

\texttt{groupcopy}  
Copy parameters of group from one tree to another (C)

\texttt{gsh2pul}  
Set up parameters for shaped gradients tests (M)

\texttt{gspoil}  
Spoiler gradient level (P)

\texttt{gss}  
Slice selection gradient strength (P)

\texttt{gsf}  
Slice selection fractional refocusing (P)

\texttt{gsspat}  
Slice-select gradient shape (P)

\texttt{gssr}  
Slice selection refocusing gradient (P)

\texttt{gss2,gss3}  
Slice selection gradient level (P)

\texttt{gttnoesy}  
Set up a PFG TNNOESY parameter set (M)

\texttt{gttnroesy}  
Set up a PFG absolute-value ROESY parameter set (M)

\texttt{gtotlimit}  
Gradient total limit (P)

\texttt{gtrim}  
Trim gradient level (P)

\texttt{gvox1-gvox3}  
Gradient strength for voxel selection (P)

\texttt{gx, gy, gz}  
Gradient strength for X, Y, and Z gradients (P)

\texttt{gxcal,gycal,gzcal}  
Gradient calibration constants (P)

\texttt{gxmax,gymax,gzmax}  
Maximum gradient strength for each axis (P)

\texttt{gzlvl}  
Pulsed field gradient strength (P)

\texttt{gzsize}  
Number of z-axis shims used by gradient shimming (P)

\texttt{gzw}\texttt{i}n  
Spectral width percentage used for gradient shimming (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 `UNITY/INOV4`, 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.

On `MERCUryplus/Vx`, 0 to 38, in steps of 2 dB (38 represents the highest possible receiver gain and 0 the lowest).

See also: `VnmrJ Liquids NMR`

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`

---

**gap**

Slice gap (P)

**Applicability:** Systems with imaging capabilities.

**Description:** Gap between slices.

---

**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: *VnmrJ Liquids NMR*

**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:**
- `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)

**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 `creategtabl` 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 parameters boresize, gmax, and trise are 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.

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.

On the right is a gradient table (gradient coil name: tc203) for a three-axis gradient set with unequal maximum gradient strength.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>boresize</td>
<td>22.50 cm</td>
</tr>
<tr>
<td>gmax</td>
<td>5.00 gauss/cm</td>
</tr>
<tr>
<td>trise</td>
<td>0.000500 sec</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>boresize</td>
<td>5.10 cm</td>
</tr>
<tr>
<td>trise</td>
<td>0.000200 sec</td>
</tr>
<tr>
<td>gxmax</td>
<td>29.00 gauss/cm</td>
</tr>
<tr>
<td>gymax</td>
<td>27.00 gauss/cm</td>
</tr>
<tr>
<td>gzmax</td>
<td>70.00 gauss/cm</td>
</tr>
</tbody>
</table>

Related:
- boresize: Magnet bore size (P)
- creategtable: Generate new gradient calibration file (M)
- gmax: Maximum gradient strength (P)
- setgcoil: Assign sysgcoil configuration parameter (M)
- sysgcoil: System gradient coil (P)
- trise: Gradient rise time (P)
- updtgcoil: Update gradient coil (M)

See also: User Programming

**gcosy**

Set up pulse sequence for gradient COSY (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: VnmrJ Liquids NMR

**gCOSY**

Change parameters for gCOSY experiment (M)

Description: Converts the current parameter set to a gCOSY experiment.

**Gcosy**

Convert the parameter to a gradient COSY experiment (M)

Description: Convert the parameter to a gradient COSY experiment

**gcrush**

Crusher gradient level (P)

Description: Predefined parameter available for use in setting a crusher gradient level, often paired with the timing parameter tcrush.
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`.

Gdqc

**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:** VnmrJ Liquids NMR

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')
```

**Related:**
- `gspoil` Spoiler gradient level (P)
- `tspoil` Gradient spoiling time (P)
getActiveStacks Get active overlay (C)
Applicability: Systems with imaging capabilities.
Description: Starts planning with overlays calculated from current VnmrJ parameters.
See also: VnmrJ Imaging NMR
Related: get1d Select a 1D experiment for processing (M)
svf Save FIDs in current experiment (C)

getcOronal Get coronal overlay (C)
Applicability: Systems with imaging capabilities.
Description: Starts planning with overlays determined with default parameters and coronal orientation.
See also: VnmrJ Imaging NMR
Related: gplan Start interactive image planning (C)

getDefaultSize Get default FOV
Applicability: Systems with imaging capabilities.
Description: Gets default field-of-view.
See also: VnmrJ Imaging NMR
Related: gplan Start interactive image planning

getDefaultSlices Get slices (C)
Applicability: Systems with imaging capabilities.
Description: Gets default number of slices.
Used to update data entry. Because default parameters are not VnmrJ variable, they can be accessed only through functions.
See also: VnmrJ Imaging NMR
Related: gplan Start interactive image planning (C)

getDefaultStacks Get overlay based on scout image (C)
Applicability: Systems with imaging capabilities.
Description: Starts planning with overlays determined from default parameters and orientation of scout image.
See also: VnmrJ Imaging NMR
Related: gplan Start interactive image planning (C)

getDefaultThk Get slice thickness (C)
Applicability: Systems with imaging capabilities.
Description: Gets default thickness of slices.
See also: VnmrJ Imaging NMR
Related: gplan Start interactive image planning (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: *VnmrJ Liquids NMR*

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)

---

**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*
getGapMode  Get gap mode (C)
Applicability: Systems with imaging capabilities.
Description: Gets gap mode.
See also: VnmrJ Imaging NMR
Related: gplan  Start interactive image planning (C)

getgcal  Get gcal value from table (M)
Applicability: Systems with the imaging module.
Syntax: getgcal<(ecc_file)>
Description: Retrieves value of the gradient calibration constant gcal from the reference table ecctabl in the directory $vnmrsystem/imaging/eddylib. If the value would overwrite the current value of gcal, the monitor displays a prompt to confirm the overwrite.
Arguments: ecc_file specifies the name of the ecc file in the reference table ecctabl. The default value is 'curecc'.
Examples: getgcal getgcal('test1')
See also: VnmrJ Imaging NMR
Related: ecc  Set up parameters to obtain compensation data (M)
ecctabl  Put gcal value and ecc file into table (M)
gcal  Gradient calibration constant (P)

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)

getMilestoneStacksGet overlay from saved parameters (C)
Applicability: Systems with imaging capabilities.
Description: Starts planning with overlays determined from saved milestone parameters.
See also: VnmrJ Imaging NMR
Related: gplan  Start interactive image planning (C)

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: VnmrJ Liquids NMR

getplane Extract planes from a 3D spectral data set (M)

Applicability:  
All systems; however, although getplane is available on MERCURYplus/Vx, such systems can only process 3D data and cannot acquire 3D data.

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: VnmrJ Liquids NMR

Related:  
dplane Display a 3D plane (M)
dproj Display a 3D plane projection (M)
dplanes Display a series of 3D planes (M)
ft3d Perform a 3D Fourier transform (M)
nextpl Display the next 3D plane (M)
getPrevStacks Start planning with previous stacks

Applicability: Systems with imaging capabilities.

Description: Starts planning with previous parameter set.

See also: VnmrJ Imaging NMR

Related: gplan Start interactive image planning (C)

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)
umreg Return the number of regions in a spectrum (C)
region Divide spectrum into regions (C)
z Add integral reset point at cursor position (C)

getsagittal Get sagittal overlay (C)

Applicability: Systems with imaging capabilities.

Description: Starts planning with overlays determined with default parameters and sagittal orientation.

See also: VnmrJ Imaging NMR

Related: gplan Start interactive image planning (C)

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:  
VnmrJ Liquids NMR

getTransverse  
Get transverse overlay (C)

Applicability:  
Systems with imaging capabilities.

Description:  
Starts planning with overlays determined with default parameters and transverse orientation.

See also:  
VnmrJ Imaging NMR

Related:  
gplan  
Start interactive image planning (C)

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:  
VnmrJ Liquids NMR

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’.

Related:  
car  
Completed transients (P)

t  
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)

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)
See also: `gettype('pw')`: $\text{$\texttt{int}$, $\text{$\texttt{name}$ sets $\text{$\texttt{int}$ to 6 and $\text{$\texttt{name}$ to 'pulse'.}}$}}$

### getValue

**Get value of parameter in a tree (C)**

**Syntax:**
```
getvalue(parameter<,index><,tree>)
```

**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.

**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.

**Examples:**
```
getvalue('arraydim')
```

See also: *User Programming*

### 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: *VnmrJ Liquids NMR*

**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 $\exp\left(-\frac{(t/\text{gf})^2}{2}\right)$ 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: *VnmrJ Liquids NMR*

Related:  
- $gf_1$: Gaussian function in 1st indirectly detected dimension (P)
- $gf_2$: Gaussian function in 2nd indirectly detected dimension (P)
- $gfs$: Gaussian shift constant in directly detected dimension (P)

$gf_1$  
**Gaussian function in 1st indirectly detected dimension (P)**

Description: Defines a Gaussian time constant of the form $\exp(-((t/gf_1)^2))$ along the first indirectly detected dimension. This dimension is referred to as the $f_1$ dimension of a multidimensional data set. $gf_1$ 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: *VnmrJ Liquids NMR*

Related:  
- $gf$: Gaussian function in directly detected dimension (P)

$gf_2$  
**Gaussian function in 2nd indirectly detected dimension (P)**

Description: Defines a Gaussian time constant of the form $\exp(-((t/gf_2)^2))$ along the second indirectly detected dimension. This dimension is referred to as the $f_2$ dimension of a multidimensional data set. $gf_2$ works analogously to the parameter $gf$. The $wti$ program can be used to set $gf_2$ on the 2D interferogram data.

Values: Number, in seconds.

See also: *VnmrJ Liquids NMR*

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 $f_1$ dimension in 3D data sets, etc. Typical value is $gfs = 'n'$.

See also: *VnmrJ Liquids NMR*

Related:  
- $gf$: Gaussian function in directly detected dimension (P)
- $gfs_1$: Gaussian shift const. in 1st indirectly detected dimension (P)
- $gfs_2$: Gaussian shift const. in 2nd indirectly detected dimension (P)

$gfs_1$  
**Gaussian shift const. in 1st indirectly detected dimension (P)**

Description: Working in combination with the $gf_1$ parameter, $gfs_1$ allows shifting the center of the Gaussian function $\exp(-((t-gfs_1)/gf_1)^2)$ along the first indirectly detected dimension. This dimension is referred to as the $f_1$ dimension
in multidimensional data sets. \textit{gfs1} works analogously to the parameter \textit{gfs}. The “conventional” parameters (i.e., \textit{lb, gf, etc.}) operate on the detected FIDs, while this “2D” parameter is used during processing of the interferograms.

See also: \textit{VnmrJ Liquids NMR}

Related: \textit{gf} Gaussian function in directly detected dimension (P)
\textit{gf1} Gaussian function in 1st indirectly detected dimension (P)
\textit{gfs} Gaussian shift const. in directly detected dimension (P)

\textbf{\textit{gfs2}} \hspace{1cm} \textbf{Gaussian shift const. in 2nd indirectly detected dimension (P)}

Description: Working in combination with the \textit{gf2} parameter, \textit{gfs2} allows shifting the center of the Gaussian function \[ \exp\left(-\left(\frac{t-gfs2}{gf2}\right)^2\right) \] along the second indirectly detected dimension. This dimension is referred to as the \textit{f2} dimension in multidimensional data sets. \textit{gfs2} works analogously to the parameter \textit{gfs}. The \textit{wti} program can be used to set \textit{gfs2} on the 2D interferogram data.

See also: \textit{VnmrJ Liquids NMR}

Related: \textit{gf} Gaussian function in directly detected dimension (P)
\textit{gf2} Gaussian function in 2nd indirectly detected dimension (P)
\textit{gfs} Gaussian shift const. in directly detected dimension (P)
\textit{wti} Interactive weighting (C)

\textbf{\textit{gHMBC}} \hspace{1cm} \textbf{Change parameters for gHMBC experiment (M)}

Description: Converts the current parameter set to a gHMBC experiment.

\textbf{\textit{Ghmbc}} \hspace{1cm} \textbf{Convert the parameter to a gradient HMBC experiment (M)}

Description: Convert the parameter to a gradient HMBC (gHMBC) experiment

\textbf{\textit{ghmqc}} \hspace{1cm} \textbf{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: \textit{VnmrJ Liquids NMR}

\textbf{\textit{gHMQC}} \hspace{1cm} \textbf{Set up parameters for gHMQC experiment (M)}

Description: Converts the current parameter set to a $^{13}$C gHMQC experiment.

\textbf{\textit{Ghmqc}} \hspace{1cm} \textbf{Convert the parameter to a gradient HMQC experiment (M)}

Description: Convert the parameter to a gradient HMQC experiment

\textbf{\textit{gHMQC15}} \hspace{1cm} \textbf{Set up parameters for $^{15}$N gHMQC experiment (M)}

Description: Converts the current parameter set to a gHMQC experiment for $^{15}$N.

\textbf{\textit{gHMQC_d2}} \hspace{1cm} \textbf{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 \(^{13}\text{C}\) gHMQC experiment using dec. 2 (M)  
Description: Converts the current parameter set to a gHMQC experiment for \(^{13}\text{C}\) with decoupler 2 as \(^{13}\text{C}\).

**ghmqcps**  
Set up a PFG HMQC phase-sensitive pulse sequence (M)  
Applicability: Systems with a pulsed field gradient module. Not available on MERCURYplus/Vx.  
Description: Prepares an experiment for a PFG (pulsed field gradient) HMQC, phase-sensitive version.  
See also: *VnmrJ Liquids NMR*

**gHMQCTOXY**  
Change parameters for gHMQCTOXY experiment (M)  
Description: Converts the current parameter set to a gHMQCTOXY experiment.

**ghsqc**  
Set up a PFG HSQC pulse sequence (M)  
Applicability: Systems with a pulsed field gradient module (except MERCURYplus/Vx).  
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 \(^{13}\text{C}\) or \(^{15}\text{N}\). The default is \(^{13}\text{C}\).  
See also: *VnmrJ Liquids NMR*

**gHSQC**  
Set up parameters for gHSQC experiment (M)  
Description: Converts the current parameter set to a \(^{13}\text{C}\) gHSQC experiment.

**Ghsqc**  
Convert the parameter to a gradient HSQC experiment (M)  
Description: Convert the parameter to a gradient HSQC experiment.

**gHSQC15**  
Set up parameters for \(^{15}\text{N}\) gHSQC experiment (M)  
Description: Converts the current parameter set to a gHSQC experiment for \(^{15}\text{N}\).

**gHSQC_d2**  
Set up parameters for \(^{15}\text{N}\) gHSQC experiment using dec. 2 (M)  
Description: Converts the current parameter set to a gHSQC experiment for \(^{15}\text{N}\) with decoupler 2 as \(^{15}\text{N}\).

**gHSQC_d213**  
Set up parameters for \(^{13}\text{C}\) gHSQC experiment using dec. 2 (M)  
Description: Converts the current parameter set to a gHSQC experiment for \(^{13}\text{C}\) with decoupler 2 as \(^{13}\text{C}\).

**gHSQCTOXY**  
Set up parameters for gHSQCTOXY experiment (M)  
Description: Converts the current parameter set to a gHSQCTOXY experiment.
Ghsqctoxy  Convert paramaters 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: VnmrJ Liquids NMR

gin  Return current mouse position and button values (C)
Syntax: gin<(<'Bn_press'>)<,'Bn_release'>>:x,y,$b1,$b2,$b3
Description: Returns the mouse pointer position and button values. gin is most often used with the draw, move, and box commands.
Arguments: 'Bn_press' is a keyword for the mouse button pressed: 'B1_press' for the left button, 'B2_press' for the middle button, or 'B3_press' for the right button. gin waits until a button is pressed. For example, given 'B1_press', gin waits until button 1 or any key is pressed. If gin is waiting for a button press and a keyboard key is pressed, all buttons are set to released (0). The default is to immediately report the mouse position.
'Bn_release' is a keyword for the mouse button released: 'B1_release' for the left button, 'B2_release' for the middle button, or 'B3_release' for the right button. gin waits until a button is released. For example, given 'B1_release', gin waits until button 1 or any key is released. If gin is waiting for a release, all buttons are set to released (1). The default is to immediately report the mouse position.
$x$ 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. If the pointer position is outside the graphics window in the $x$ direction, $x$ returns –1.
$y$ is the value in the $y$ direction, in millimeters, of the pointer. The range of $y$ is –20 at the bottom of the chart and $wc2max$ at the top. If the pointer position is outside the graphics window in the $y$ direction, $y$ returns –10000.
$b1$, $b2$, $b3$ report the state of the left, middle, and right mouse buttons, respectively. The value is 1 if the corresponding mouse button is down; 0 if the corresponding mouse button is up.
Examples: gin:$x,y,b1,b2$
gin('B2_press'):$x,y,b1,b2,b3$
gin('B1_release'):$x,y,b1$
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)
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: VnmrJ Liquids NMR; VnmrJ Walkup NMR
Related: 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. The default is that `glue` looks for a parameter `nscans` and assumes that all experiments are to be used. 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: VnmrJ Liquids NMR

Related:
- `nscans` Number of scout/real scan repetitions (P)
- `savefile` Base file name for saving FIDs or data sets (P)

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` program is done, it automatically exits, and the previous data set is retrieved.

CAUTION: Do not spin the sample during gradient shimming.

Arguments:  `'files'` is a keyword to enter the gradient autoshimming files menu.
- `'mapname'` is a keyword to display the current map name.
- `'quit'` is a keyword to exit from gradient autoshimming and retrieve the previous data set.

See also: VnmrJ Liquids NMR

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')`.

gmaps

Run gradient autoshimming, set parameters, map shims (M)

Applicability: Systems with gradient shimming installed.

Syntax:  
1. `gmaps<(option)>`
2. `gmaps('shimmmap.'<,shimmmap_option>)`
Description: Enters the Gradient Shimming System menu for setting parameters, mapping
the shims, and performing autoshimming. This is the only entry point to the
gradient shimming system menu.

If the gmapz pulse sequence is not loaded, retrieve parameters from the last
shimmap used (see current mapname) or from gmapz.par if no shimmap
exists.

**CAUTION:** Do not spin the sample during gradient shimming.

Arguments: `option` is one of the following keywords:

- `'addpar'` adds gradient shimming parameters to the current parameter
  set.
- `'findgzlv1'` runs an experiment to calibrate gzlv1, 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'` starts autoshimming (same as 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: *VnmrJ Liquids NMR*

**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_findtof` Gradient shimming flag to first find tof (P)
- `gmap_z1z4` Gradient shimming flag to first shim z1-z4 (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: VnmrJ Liquids NMR

Related: gmapshim Start gradient autoshimming (M)
gmapsyt Run gradient autoshimming, set parameters, map shims (M)
gmap_z1z4 Gradient shimming flag to first shim z1-z4 (P)
gmax Maximum gradient strength (P)
gmaxz1z4 Maximum gradient strength for each axis (P)
gmaxz2z4 Maximum gradient strength for each axis (P)
gmaxz3z4 Maximum gradient strength for each axis (P)
gmaxz1z2 Maximum gradient strength for each axis (P)
gmaxz1z3 Maximum gradient strength for each axis (P)
gmaxz2z3 Maximum gradient strength for each axis (P)
gmaxz1z2z3 Maximum gradient strength for each axis (P)

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: VnmrJ Liquids NMR

Related: gmapshim Start gradient autoshimming (M)
gmapsyt Run gradient autoshimming, set parameters, map shims (M)
gmap_z1z4 Gradient shimming flag to first shim z1-z4 (P)
gmax Maximum gradient strength (P)
gmaxz1z4 Maximum gradient strength for each axis (P)
gmaxz2z4 Maximum gradient strength for each axis (P)
gmaxz3z4 Maximum gradient strength for each axis (P)
gmaxz1z2 Maximum gradient strength for each axis (P)
gmaxz1z3 Maximum gradient strength for each axis (P)
gmaxz2z3 Maximum gradient strength for each axis (P)
gmaxz1z2z3 Maximum gradient strength for each axis (P)

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: VnmrJ Liquids NMR

Related: gmapshim Start gradient autoshimming (M)
gmapsyt Run gradient autoshimming, set parameters, map shims (M)
gmap_z1z4 Gradient shimming flag to first shim z1-z4 (P)
gmax Maximum gradient strength (P)
gmaxz1z4 Maximum gradient strength for each axis (P)
gmaxz2z4 Maximum gradient strength for each axis (P)
gmaxz3z4 Maximum gradient strength for each axis (P)
gmaxz1z2 Maximum gradient strength for each axis (P)
gmaxz1z3 Maximum gradient strength for each axis (P)
gmaxz2z3 Maximum gradient strength for each axis (P)
gmaxz1z2z3 Maximum gradient strength for each axis (P)

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: VnmrJ Liquids NMR

Related: gmapshim Start gradient autoshimming (M)
gmapsyt Run gradient autoshimming, set parameters, map shims (M)
gmap_z1z4 Gradient shimming flag to first shim z1-z4 (P)
gmax Maximum gradient strength (P)
gmaxz1z4 Maximum gradient strength for each axis (P)
gmaxz2z4 Maximum gradient strength for each axis (P)
gmaxz3z4 Maximum gradient strength for each axis (P)
gmaxz1z2 Maximum gradient strength for each axis (P)
gmaxz1z3 Maximum gradient strength for each axis (P)
gmaxz2z3 Maximum gradient strength for each axis (P)
gmaxz1z2z3 Maximum gradient strength for each axis (P)

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: VnmrJ Liquids NMR

Related: gmapshim Start gradient autoshimming (M)
gmapsyt Run gradient autoshimming, set parameters, map shims (M)
gmap_z1z4 Gradient shimming flag to first shim z1-z4 (P)
gmax Maximum gradient strength (P)
gmaxz1z4 Maximum gradient strength for each axis (P)
gmaxz2z4 Maximum gradient strength for each axis (P)
gmaxz3z4 Maximum gradient strength for each axis (P)
gmaxz1z2 Maximum gradient strength for each axis (P)
gmaxz1z3 Maximum gradient strength for each axis (P)
gmaxz2z3 Maximum gradient strength for each axis (P)
gmaxz1z2z3 Maximum gradient strength for each axis (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:** *VnmrJ Liquids NMR*

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:** *VnmrJ Liquids NMR*

g** Submit experiment to acquisition (M)**

**Syntax:**

```
g<mac>'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 
\texttt{wexp} processing of the experiment, started with \texttt{go('wait')}, is finished.

Examples:
\begin{verbatim}
\texttt{go}
\texttt{\ ('nosafe')}
\texttt{go('next')}
\end{verbatim}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{acqi} Interactive acquisition display process (C)
\texttt{au} Submit experiment to acquisition and process data
\texttt{change} Submit a change sample experiment to acquisition (M)
\texttt{gain} Receiver gain (P)
\texttt{ga} Submit experiment to acquisition and FT the result (C)
\texttt{gf} Prepare parameters for FID/spectrum display in \texttt{acqi} (M)
\texttt{go} Pulse sequence setup macro called by \texttt{go, ga, and au} (M)
\texttt{load} Load status of displayed shims (P)
\texttt{loc} Location of sample in tray (P)
\texttt{lock} Submit an Autolock experiment to acquisition (C)
\texttt{method} Autoshim method (P)
\texttt{probe\_protection} Probe protection control (P)
\texttt{sample} Submit change sample, Autoshim exp. to acquisition (M)
\texttt{seqfil} Pulse sequence name (P)
\texttt{shim} Submit an Autoshim experiment to acquisition (C)
\texttt{spin} Submit a spin setup experiment to acquisition (C)
\texttt{spin} Sample spin rate (P)
\texttt{su} Submit a setup experiment to acquisition (M)
\texttt{usergo} Experimental setup macro called by \texttt{go, ga, and au} (M)
\texttt{wshim} Conditions when shimming is performed (P)

\textbf{go\_}

\textbf{Pulse sequence setup macro called by \texttt{go, ga, and au} (M)}

\textbf{Syntax}: \texttt{go\_\_macro}

\textbf{Description}: Called by the macros \texttt{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 \texttt{go\_} with the name of the pulse sequence macro (from \texttt{seqfil}) to be used.

Examples:
\begin{verbatim}
\texttt{go\_dept}
\texttt{go\_noesy}
\texttt{go\_s2pul}
\end{verbatim}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{au} Submit experiment to acquisition and process data (M)
\texttt{ga} Submit experiment to acquisition and FT the result (M)
\texttt{go} Submit experiment to acquisition (M)
\texttt{seqfil} Pulse sequence name (P)
\texttt{usergo} Experimental setup macro called by \texttt{go, ga, and au} (M)

\textbf{gpat-gpat3}

\textbf{Gradient shape (P)}

\textbf{Description}: Predefined string parameters available to specify gradient shapes.

See also: \textit{VnmrJ Imaging NMR}

\textbf{gpe}

\textbf{Phase encoding gradient increment (P)}

\textbf{Applicability}: Systems with imaging capabilities.
**Description:** Value of the change in phase encode gradient level from one phase encode step to the next. More precisely, the product of the parameters \(gpe\) and \(tpe\) is used internally within the pulse sequence to determine the phase encode gradient increment based on the computed refocusing time for readout and slice selection. \(gpe\) depends on the field of view and the phase encode gradient duration according to the expression \(\gamma \cdot gpe \cdot tpe \cdot lpe = 1\) and is set by either the `imprep` or `setgpe` macros.

See also: *VnmrJ Imaging NMR*

**Related:**
- `imprep` Set up rf pulses, imaging and voxel selection gradients (M)
- `gmax` Maximum gradient strength (P)
- `gpe2` Second phase encoding gradient increment (P)
- `gpe3` Third phase encoding gradient increment (P)
- `lpe` Field of view parameter for phase encode in cm (P)
- `nv` Number of 2D phase encode steps to be acquired (P)
- `setgpe` Set phase encode gradient levels (M)
- `tpe` Duration of the phase encoding gradient pulse (P)

**gpe2**

2nd phase encode gradient increment

**Applicability:** Systems with imaging capabilities.

**Description:** Phase encode gradient increment for 3D or 4D phase encoded applications. \(gpe2\) should be used when a second phase encode gradient is required. For example, a 3D volume imaging application would use both \(gpe\) and \(gpe2\), as would a 3D chemical shift imaging experiment (that is, two spatial dimensions plus chemical shift dimension).

**Related:**
- `imprep` Set up rf pulses, imaging and voxel selection gradients (M)
- `gmax` Maximum gradient strength (P)
- `gpe` Phase encoding gradient increment (P)
- `gpe3` 3rd phase encoding gradient increment (P)
- `lpe2` Field of view size for 2nd phase-encode axis (P)
- `setgpe` Set phase encode gradient levels (M)
- `tpe2`, `tpe3` Duration of the 2nd and 3rd phase encoding gradient periods (P)

**gpe3**

3rd phase encode gradient increment

**Applicability:** Systems with imaging capabilities.

**Description:** Phase encode gradient increment for 3D or 4D phase encoded applications. \(gpe3\) should be used when a third phase encode gradient is required. It is available for use in a 4D CSI experiment (three spatial dimensions, one chemical shift).

**Related:**
- `imprep` Set up rf pulses, imaging and voxel selection gradients (M)
- `gpe` Phase encoding gradient increment (P)
- `gpe2` 2nd phase encoding gradient increment (P)
- `lpe2` Field of view size for 3rd phase-encode axis (P)
- `setgpe` Set phase encode gradient levels (M)
- `tpe2`, `tpe3` Duration of the 2nd and 3rd phase encoding gradient periods (P)

**gped**

Phase encode dephasing gradient in the EPI sequence (P)

**Applicability:** Systems with imaging capabilities.

**Description:** Determines echo position in the phase-encode direction. A blipped gradient phase encodes the signal with respect to the phase-encode direction. \(gped\) determines the center of the k-space along the phase-encode direction. \(gped\) is
usually set so that \texttt{eff\_echo} appears at the center of the phase encode dimension, $t_1$.

 Related: \texttt{eff\_echo}  
 Effective echo position in EPI experiments (P)

\textbf{gpmult}  
\textbf{Phase encode gradient increment multiplier (P)}

\textbf{Applicability:} Systems with imaging capabilities.

\textbf{Description:} Multiplier used to correct phase encode gradient increment when using a non-rectangular phase encode gradient shape. For example, a rectangular shaped phase encode gradient has a gradient-time integral equal to 1.571 that of a half-sine gradient of equal duration and peak amplitude. In this case, set \texttt{gpmult} to 1.571 to yield the expected field of view.

\textbf{See also:} \textit{VnmrJ Imaging NMR}

\textbf{gplan}  
\textbf{Start interactive image planning (C)}

\textbf{Syntax:} \texttt{gplan(function\_name, arg1, arg2,...)}

\textbf{Description:} In VnmrJ, starts an image planning session.

\textbf{Arguments:} ‘\texttt{function\_name}' , \texttt{path} is the name of an image planning function surrounded by single quotation marks.
arg1, arg2,... are arguments for the function, if relevant.

\textbf{Examples:} gplan 'clearStacks()'  
get 'PrevStacks()'  

\textbf{See also:} \textit{VnmrJ Liquids NMR}

\textbf{gradaxis}  
\textbf{Gradient axis (P)}

\textbf{Applicability:} Systems with imaging capabilities.

\textbf{Description:} Selects the gradient axis in macros such as \texttt{g2pul} and \texttt{profile}.

\textbf{Values:} ‘x’, ‘y’, ‘z’

\textbf{See also:} \textit{VnmrJ Imaging NMR}

\textbf{Related:} \texttt{g2pul}  
Set up pulse sequence for gradient evaluation (M)

\texttt{profile}  
Set up pulse sequence for gradient calibration (M)

\textbf{gradientdisable}  
\textbf{Disable PFG gradients (P)}

\textbf{Description:} \texttt{gradientdisable} is an optional global parameter for disabling the gradient pulses. If \texttt{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 \texttt{gradientdisable} as a global parameter of type ‘flag’. If \texttt{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, \texttt{gradtypes} of ‘l’, ‘p’, and ‘t’.

\textbf{Related:} \texttt{pfgon}  
Pulsed field gradient amplifiers on/off control (P)

\texttt{gradtype}  
Gradients for X, Y, and Z axes (P)
gradsteplsz | Gradient step size (P)
---|---
**Description:** The maximum gradient DAC value. gradsteplsz 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 CONFIG 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., 'npn'). 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 CONFIG 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; VnmrJ Liquids NMR*

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:**
```plaintext
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)
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: VnmrJ Liquids NMR
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)

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: VnmrJ Liquids NMR
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)

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, VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related: plgrid Plot a grid on a 2D plot (M)

griserate Gradient rise rate (P)

Applicability: Systems with imaging capabilities.

Description: Sets the gradient rise rate.

See also: VnmrJ Imaging NMR

Related: gcoil Read data from gradient calibration tables (P)
gxcal, gycal, gzcal Gradient calibration constants (P)

gro Readout gradient strength (P)

Applicability: Systems with the or imaging capabilities.

Description: Controls the level of the readout gradient, if present. imprep sets gro based on its internal algorithm; or use setgro(value), which sets gro to a specific value and updates at and sw. gro, sw, and at are related by the expression sw=g*lro*gro, but a change in lro does not automatically update gro and sw.

See also: VnmrJ Imaging NMR

Related: at Acquisition time (P)
gmax Maximum gradient strength (P)
grof Read out fractional compensation (P)
gror Read out compensation gradient (P)
imprep Set up rf pulses, imaging and voxel selection gradients (M)
lro Field of view size for readout axis (P)
setgro Set readout gradient (M)
sw Spectral width in directly directed dimension (P)

groa Readout gradient adjuster in EPI experiment (P)

Applicability: Systems with echo planar imaging (EPI) capabilities.

Description: Corrects readout gradient imperfections in EPI experiment by adding an offset (G/cm) to the odd readgradient.
See also: *VnmrJ Imaging NMR*

**grof**

**Fine tune readout gradient compensation (P)**

Applicability: Systems with imaging capabilities.

Description: Factor for fine tuning of the readout gradient compensation.

*See also: VnmrJ Imaging NMR*

<table>
<thead>
<tr>
<th>Related</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>at</strong></td>
<td>Acquisition time (P)</td>
</tr>
<tr>
<td><strong>gmax</strong></td>
<td>Maximum gradient strength (P)</td>
</tr>
<tr>
<td><strong>gror</strong></td>
<td>Read out compensation gradient (P)</td>
</tr>
<tr>
<td><strong>imprep</strong></td>
<td>Set up rf pulses, imaging and voxel selection gradients (M)</td>
</tr>
<tr>
<td><strong>lro</strong></td>
<td>Field of view size for readout axis (P)</td>
</tr>
<tr>
<td><strong>setgro</strong></td>
<td>Set readout gradient (M)</td>
</tr>
<tr>
<td><strong>sw</strong></td>
<td>Spectral width in directly directed dimension (P)</td>
</tr>
</tbody>
</table>

**gropat**

**Readout gradient shape (P)**

Applicability: Systems with imaging capabilities.

Description: Predefined string parameter to specify a readout gradient shape.

**gror**

**Read out compensation gradient (P)**

Applicability: Systems with imaging capabilities.

Description: Controls the level of the readout refocusing gradient when `pilot='n'`. When `pilot='y'`, `gror` is ignored by the pulse sequence, and computed internally. In this case the internal value is printed in the window used to start VnmrJ.

`gror` is opposite in sign to `gro` for gradient echo experiments (e.g., FLASH), and has the same sign as `gro` for spin-echo experiments (e.g. SEMS).

Values: Sequence dependent, specified in gauss/cm up to `±gmax`.

**grora**

**Readout dephasing gradient adjuster in EPI experiment (P)**

Applicability: Systems with echo planar imaging (EPI) capabilities.

Description: Correction gradient value added to the readout refocusing gradient (G/cm) in EPI experiments to center the echo position in the acquisition window.

<table>
<thead>
<tr>
<th>Related</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>episet</strong></td>
<td>Set up parameters in EPI experiment (M)</td>
</tr>
<tr>
<td><strong>groa</strong></td>
<td>Readout gradient adjuster in EPI experiment (P)</td>
</tr>
<tr>
<td><strong>tep</strong></td>
<td>Post-acquisition delay in EPI experiment (P)</td>
</tr>
</tbody>
</table>

**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)

**gsh2pul**

*Set up parameters for shaped gradients tests (M)*

Applicability: Systems with the imaging module.

Description: During imaging installation, `gsh2pul` is used to load parameters sets for shaped `gsh2Dpul` gradients tests. `gsh2Dpul` steps the amplifier with the value of `ni`.

Description: *VnmrJ Imaging NMR*

Related: `ni` Number of increments in 1st indirectly detected dimension (P)

**gspoil**

*Spoiler gradient level (P)*

Description: Predefined parameter to set a spoiler gradient level. It is often paired with the timing parameter `tspoil`.

Related: `tspoil` Spoiling gradient control (P)

**gss**

*Slice selection gradient strength (P)*

Applicability: Systems with imaging capabilities.

Description: Controls the level of the slice-select gradient, if present. `imprep` sets `gss` based on the slice thickness and rf pulse bandwidths; or use `setgss` to update only `gss`.

Values: Number less than ±`gmax`, in gauss/cm.

Related: `gmax` Maximum gradient strength (P)
- `gssf` Slice selection fractional gradient (P)
- `gssr` Slice selection refocusing gradient (P)
- `imprep` Set up rf pulses, imaging and voxel selection gradients (M)
- `setgss` Select slice or voxel selection gradient levels (M))
- `thk` 2D imaging plane slice thickness (P)

**gssf**

*Slice selection fractional refocusing (P)*

Applicability: Systems with imaging capabilities.

Description: Fractional multiplier used as a fine tuning adjustment for the `gssr` slice refocusing gradient level.

Values: 1.0, when the theoretical gradient calculations are correct.

See also: *VnmrJ Imaging NMR*

Related: `grof` Read out fractional compensation (P)
- `gss` Slice selection gradient strength (P)
- `gssr` Slice selection refocusing gradient (P)

**gsspat**

*Slice-select gradient shape (P)*

Description: Predefined string parameter to specify a slice-select gradient shape.
sgr
Slice selection refocusing gradient (P)
Applicability: Systems with imaging capabilities.
Description: Controls the level of the slice-select refocusing gradient when \texttt{pilot}='n'.
When \texttt{pilot}='y', \texttt{gssr} is ignored by the pulse sequence, and internally
computed. The internal value is printed in the window used to start VnmrJ.
\texttt{gssr} is normally be opposite in sign to \texttt{gss}.
Values: Number in gauss/cm up to $\pm g_{\max}$. Nominal value is $g_{ssr}=-0.5\times g_{ss}$.
See also: \textit{VnmrJ Imaging NMR}
Related: \texttt{gmax} Maximum gradient strength (P)
\texttt{gss} Slice selection gradient strength (P)
\texttt{gssf} Slice selection fractional gradient (P)
\texttt{gror} Read out compensation gradient (P)
\texttt{pilot} Automatic sequence setup (P)

gss2,gss3
Slice selection gradient level (P)
Description: Predefined parameters for specifying gradient levels for different slice selection
events in an imaging pulse sequence.
See also: \textit{VnmrJ Imaging NMR}
Related: \texttt{gss} Slice selection gradient strength (P)

gtnnoesy
Set up a PFG TNNOESY parameter set (M)
Applicability: Systems with the pulsed field gradient (PFG) module. Not available on
\textit{MERCURY}plus/\textit{Vx}.
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 \texttt{gtnnoesy} experiment.

gtnroesy
Set up a PFG absolute-value ROESY parameter set (M)
Applicability: Systems with the pulsed field gradient (PFG) module. Not available on
\textit{MERCURY}plus/\textit{Vx}.
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 \texttt{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: \texttt{creategtable} Generate system gradient table (M)
\texttt{gcoil} Read data from gradient calibration tables (P)

gtrim
Trim gradient level (P)
Description: Predefined parameter to set a trim gradient level.
gvox1-gvox3  Gradient strength for voxel selection (P)

Applicability: Systems with imaging capabilities.

Description: Voxel-select gradient levels for the first, second, and third dimensions of a voxel in a localized spectroscopy experiment. For example, `imprep` sets `gvox1` based on the corresponding voxel dimension `vox1`, and rf pulse bandwidth. For nonoblique voxels, the orientation of `gvox1` lies along one of the three main gradient axes, X, Y, or Z. Oblique angle voxel orientation is also available, and for this reason the name `gvox1` is used instead of, for example, `gx`.

Values: Number less than ±gmax, in gauss/cm.

See also: VnmrJ Imaging NMR

Related: `gx`, `gy`, `gz`  Gradient strength for X, Y, and Z gradients (P)

gx, gy, gz  Gradient strength for X, Y, and Z gradients (P)

Applicability: Systems with imaging capabilities.

Description: Defines the gradient strength of the X, Y, and Z gradients, respectively, for localized spectroscopy experiments such as ISIS and VOSY. The gradient strength in conjunction with the length of the selective pulse defines the size of the region of interest.

Values: Number less than to ±gmax, in gauss/cm (older pulse sequences, such as `isis.c` and `vosy.c`, use DAC units). The sign is often not important.

See also: VnmrJ Imaging NMR

Related: `gmax`  Maximum gradient strength (P)

Related: `gcoil`  Read data from gradient calibration tables (P)

Related: `setgcoil`  Update system gcoil configuration (M)

Related: `gcal`  Gradient calibration constants (P)

Related: `gcoil`  Read data from gradient calibration tables (P)

Related: `setgcoil`  Update system gcoil configuration (M)

Related: `gmax`  Maximum gradient strength for each axis (P)

gxcal,gycal,gzcal  Gradient calibration constants (P)

Applicability: Systems with the older SISCO imaging module.

Description: Stores the proportionality constant for each gradient. The gradients generated in the magnet require calibration so that coordinate data, slice positions, and the field of view can be set up correctly.

Values: Number less than to ±gmax, in gauss/cm/DAC (on older SISCO systems).

See also: VnmrJ Imaging NMR

Related: `gmax`  Maximum gradient strength (P)

Related: `gcoil`  Read data from gradient calibration tables (P)

Related: `gcal`  Gradient calibration constants (P)

Related: `gcoil`  Read data from gradient calibration tables (P)

Related: `setgcoil`  Update system gcoil configuration (M)
imaging systems usually have gradients set to the same maximum value, and
\texttt{gmax} can be used.

See also: \textit{VnmrJ Liquids NMR: User Programming, VnmrJ Imaging NMR}

\textbf{\texttt{gzlvl}} \hspace{1cm} \textbf{Pulsed field gradient strength (P)}

\begin{tabular}{ll}
\textbf{Applicability:} & All systems with pulsed field gradient modules. \\
\textbf{Description:} & Specifies the pulsed field gradient DAC value. \\
\textbf{Values:} & Range from +2047 to –2048 for 12-bit gradient module, and from +32767 to \-32768 for a 16-bit gradient module. \\
\textbf{Related:} & \texttt{gzsize} Number of z-axis shims used by gradient shimming (P) \\
& \texttt{gzwin} Spectral window percentage used for gradient shimming (P)
\end{tabular}

\textbf{\texttt{gzsize}} \hspace{1cm} \textbf{Number of z-axis shims used by gradient shimming (P)}

\begin{tabular}{ll}
\textbf{Applicability:} & Systems with the pulsed field gradient module. \\
\textbf{Description:} & Specifies the number of z-axis shims used by gradient shimming. For example, \texttt{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 \texttt{shimset} value. \\
\textbf{Values:} & Integer from 1 to 8. \\
\textbf{Related:} & \texttt{gmapshim} Start gradient autoshimming (M) \\
& \texttt{gmapsys} Run gradient autoshimming, set parameters, map shims (M) \\
& \texttt{gmapz} Get parameters and files for gmapz pulse sequence (M) \\
& \texttt{gzlvl} Pulsed field gradient strength (P) \\
& \texttt{gzwin} Spectral window percentage used for gradient shimming (P) \\
& \texttt{shimset} Type of shimset (P) \\
& \texttt{gmap_z1z4} Gradient shimming flag to first shim z1-z4 (P)
\end{tabular}

\textbf{\texttt{gzwin}} \hspace{1cm} \textbf{Spectral width percentage used for gradient shimming (P)}

\begin{tabular}{ll}
\textbf{Applicability:} & Systems with the pulsed field gradient module. \\
\textbf{Description:} & Specifies the percentage of the spectral width \texttt{sw} used by gradient shimming for shimmap calculations. The value is set automatically with the buttons Find \texttt{gzlvl/gzwin} and Find \texttt{gzwin} in the gradient shimming system menu opened by \texttt{gmapsys}. \\
\textbf{Values:} & A real number between 0 and 100. The typical value is 50. \\
\textbf{Related:} & \texttt{gmapshim} Start gradient autoshimming (M) \\
& \texttt{gmapsys} Run gradient autoshimming, set parameters, map shims (M) \\
& \texttt{gmapz} Get parameters and files for gmapz pulse sequence (M) \\
& \texttt{gzlvl} Pulsed field gradient strength (P) \\
& \texttt{gzsize} Number of z-axis shims used by gradient shimming (P) \\
& \texttt{sw} Spectral width in directly detected dimension (P) \\
& \texttt{tof} Frequency offset for observe transmitter (P)
\end{tabular}
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>Automated proton acquisition (M)</td>
</tr>
<tr>
<td>h1freq</td>
<td>Proton frequency of spectrometer (P)</td>
</tr>
<tr>
<td>hlp</td>
<td>Process 1D proton spectra (M)</td>
</tr>
<tr>
<td>h2cal</td>
<td>Calculate strength of the decoupler field (C)</td>
</tr>
<tr>
<td>halt</td>
<td>Abort acquisition with no error (C)</td>
</tr>
<tr>
<td>hc</td>
<td>Automated proton and carbon acquisition (M)</td>
</tr>
<tr>
<td>hcapt</td>
<td>Automated proton, carbon, and APT acquisition (M)</td>
</tr>
<tr>
<td>hcchtoesy</td>
<td>Set up parameters for HCCHTOCSY pulse sequence (M)</td>
</tr>
<tr>
<td>hccorr</td>
<td>Automated proton, carbon, and HETCOR acquisition (M)</td>
</tr>
<tr>
<td>hcdept</td>
<td>Automated proton, carbon, and DEPT acquisition (M)</td>
</tr>
<tr>
<td>hcosy</td>
<td>Automated proton and COSY acquisition (M)</td>
</tr>
<tr>
<td>hcmult</td>
<td>Execute protocol actions of apptype hcmult (M)</td>
</tr>
<tr>
<td>hdwshim</td>
<td>Hardware shimming (P)</td>
</tr>
<tr>
<td>hdwshimlist</td>
<td>List of shims for hardware shimming (P)</td>
</tr>
<tr>
<td>het2dj</td>
<td>Set up parameters for HET2DJ pulse sequence (M)</td>
</tr>
<tr>
<td>HETCOR</td>
<td>Change parameters for HETCOR experiment (M)</td>
</tr>
<tr>
<td>hetcor</td>
<td>Set up parameters for HETCOR pulse sequence (M)</td>
</tr>
<tr>
<td>hetcorcp1</td>
<td>Set up parameters for solids HETCOR pulse sequence (M)</td>
</tr>
<tr>
<td>hetcorps</td>
<td>Set up parameters for HETCORPS pulse sequence (M)</td>
</tr>
<tr>
<td>hidecommand</td>
<td>Execute macro instead of command with same name (C)</td>
</tr>
<tr>
<td>hetero2d</td>
<td>Execute protocol actions of apptype hetero2d (M)</td>
</tr>
<tr>
<td>Hmbc</td>
<td>Convert the parameter to a HMBC experiment (M)</td>
</tr>
<tr>
<td>HMBC</td>
<td>Change parameters for HMBC experiment (M)</td>
</tr>
<tr>
<td>hmqc</td>
<td>Set up parameters for HMQC pulse sequence (M)</td>
</tr>
<tr>
<td>Hmqc</td>
<td>Convert the parameter to a HMQC experiment (M)</td>
</tr>
<tr>
<td>HMQC</td>
<td>Set up parameters for HMQC experiment (M)</td>
</tr>
<tr>
<td>HMQC15</td>
<td>Set up parameters for $^{15}$N HMQC experiment (M)</td>
</tr>
<tr>
<td>HMQC_d2</td>
<td>Set up parameters for $^{15}$N HMQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>HMQC_d213</td>
<td>Set up parameters for $^{13}$C HMQC experiment using dec. 2 (M)</td>
</tr>
<tr>
<td>hmqcr</td>
<td>Set up parameters for HMQCR pulse sequence (M)</td>
</tr>
<tr>
<td>hmqctoesy</td>
<td>Set up parameters for HMQCTOCSY pulse sequence (M)</td>
</tr>
<tr>
<td>Hmqctoxy</td>
<td>Convert the parameter to a HMQCTOXY experiment (M)</td>
</tr>
<tr>
<td>HMQCTOXY</td>
<td>Set up parameters for HMQCTOXY experiment (M)</td>
</tr>
<tr>
<td>HMQCTOXY15</td>
<td>Set up parameters for $^{15}$N HMQCTOXY experiment (M)</td>
</tr>
<tr>
<td>HMQCTOXY_d2</td>
<td>Set up parameters for $^{15}$N HMQCTOXY using decoupler 2 (M)</td>
</tr>
<tr>
<td>HMQCTOXY_d213</td>
<td>Set up parameters for $^{13}$C HMQCTOXY using decoupler 2 (M)</td>
</tr>
<tr>
<td>hmqctoxy3d</td>
<td>Set up parameters for HMQC-TOCSY 3D pulse sequence (M)</td>
</tr>
<tr>
<td>ho</td>
<td>Horizontal offset (P)</td>
</tr>
<tr>
<td>hold</td>
<td>Post-trigger delay (P)</td>
</tr>
<tr>
<td>hom2dj</td>
<td>Set up parameters for HOM2DJ pulse sequence (M)</td>
</tr>
<tr>
<td>HOMODEC</td>
<td>Change parameters for HOMODEC experiment (M)</td>
</tr>
<tr>
<td>homdec</td>
<td>Proton homonuclear decoupler present (P)</td>
</tr>
<tr>
<td>homo</td>
<td>Homodecoupling control for first decoupler (P)</td>
</tr>
</tbody>
</table>
homo2d   Execute protocol actions of apptype homo2d (M)
homo2    Homodecoupling control for second decoupler (P)
homo3    Homodecoupling control for third decoupler (P)
homo4    Homodecoupling control for fourth decoupler (P)
hoult    Set parameters alfa and rof2 according to Hoult (M)
hs       Homospoil pulses (P)
hsqc     Set up parameters for HSQC pulse sequence (M)
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)
Hsqctoxy Convert parameters to a HSQCTOXY experiment (M)
HSQCTOXY Set up parameters for 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)
hzmm     Scaling factor for plots (P)
hztoomm  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 \textit{wexp} is set to \textit{procplot} for standard processing. If h1 is used as the command for automation via the \textit{enter} command, then \textit{au} is supplied automatically and should not be entered on the MACRO line of the \textit{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 \textit{enter} program. The default is \textit{CDCl3}.

Examples:
h1
h1 ('DMSO')

See also: \textit{VnmrJ Liquids NMR}

Related: au Submit experiment to acquisition and process data (M)
        enter Enter sample information for automation run (C)
        hip Process 1D proton spectra (M)
        proplot 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 CONFIG window (opened from `config`).

*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), and referencing to the TMS signal if present (`setref` macro, then `tmsref` macro).

*See also:* VnmrJ Liquids NMR

*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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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('dms0')

See also: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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)

Applicability: Sequence is not supplied with MERCURYplus/Vx.
Description: Used for sidechain assignments in fully 13C-enriched molecules.
See also: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related:
c13 Automatic carbon acquisition (M)
dept Set up parameters for DEPT experiment (M)
enter Enter sample information for automation run (M,U)
H

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>Automated proton acquisition (M)</td>
</tr>
<tr>
<td>rttmp</td>
<td>Retrieve experiment data from experiment subfile (M)</td>
</tr>
</tbody>
</table>

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: *VnmrJ Liquids NMR*

Related:
- `enter` Enter sample information for automation run (C)
- `h1` Automated proton acquisition (M)
- `rttmp` Retrieve experiment data from experiment subfile (M)

hcmult  **Execute protocol actions of apptype hcmult (M)**

- **Applicability:** Liquids systems.
- **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

hdwshim  **Hardware shimming (P)**

- **Applicability:** *UNITY/NOVA* 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. On *UNITY/NOVA*, 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 (only during a delay on *UNITY/NOVA*).
  - `'p'` turns hardware shimming on during presaturation pulse (power level change followed by pulse). Available on *UNITY/NOVA* only.
  - `'n'` turns shimming off.

See also: *VnmrJ Liquids NMR*

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)**

- **Applicability:** *UNITY/NOVA* systems
- **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 \( z_1, z_{1c}, z_2, z_{2c}, x_1, y_1 \). Commas and blank space are ignored. Shimming is done in the order \( z_1, z_2, x_1, y_1 \), regardless of the order in the string.

Examples: 
```
hdwshimlist='z1'
hdwshimlist='z1z2x1y1'
```

See also: *VnmrJ Liquids NMR*

Related: 
- `create` Create new parameter in a parameter tree (C)
- `hdwshim` Hardware shimming (P)

**het2dj**  
Set up parameters for HET2DJ pulse sequence (M)

Description: Sets up a HET2DJ (heteronuclear 2D-J) experiment.

See also: *VnmrJ Liquids NMR*

Related: `foldj` Fold J-resolved 2D spectrum about \( f_1=0 \) 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: *VnmrJ Liquids NMR*

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 with XPOLAR1, 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)

Applicability: Not supplied with *MERCURYplus/Vx* systems.

Description: Sets up parameters for a heteronuclear chemical shift correlation experiment (absolute value and phase sensitive).

See also: *VnmrJ Liquids NMR*

**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

**hetero2d** Execute protocol actions of apptype hetero2d (M)

Description: This macro is used to execute the protocol actions of the hetero2d `apptype`.

Examples: `hetero2d('setup')` execute hetero2d experimental setup  
`hetero2d('process')` execute hetero2d processing  
`hetero2d('plot')` execute hetero2d plotting

**Hmbc** Convert the parameter to a HMBC experiment (M)

Description: Convert the parameter to a HMBC experiment.

**HMBC** Change parameters for HMBC experiment (M)

Description: Converts the current parameter set to a HMBC experiment.

**hmqc** Set up parameters for HMQC pulse sequence (M)

Applicability: All systems, except that presaturation or homospoil are not available on `MERCURYplus/Vx`.

Syntax: `hmqc<(isotope)>`

Description: Sets up a HMQC heteronuclear multiple-quantum coherence) experiment.

Arguments: `isotope` is the isotope number for the heteronucleus of interest (e.g., 13 for $^{13}C$).

See also: VnmrJ Liquids NMR

**Hmqc** Convert the parameter to a HMQC experiment (M)

Description: Convert the parameter to a HMQC experiment.

**HMQC** Set up parameters for HMQC experiment (M)

Description: Converts the current parameter set to a $^{13}C$ 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: VnmrJ Liquids NMR

hmqctoosy  Set up parameters for HMQCTOCSY pulse sequence (M)
Applicability: Sequence is not supplied with MERCURYplus/Vx.
Description: Sets up a HMQCTOCSY experiment with an option to null or invert the direct responses.

Hmqctoxy  Convert the parameter to a HMQCTOXY experiement (M)
Description: Convert the parameter to a HMQCTOXY experiment.

HMQCTOXY  Set up parameters for HMQCTOXY experiment (M)
Description: Converts the current parameter set to a $^{13}$C 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)
Applicability: Not supplied with MERCURYplus/Vx systems.
Description: Sets up parameters for a HMQC-TOCSY 3D experiment with a presaturation option.
**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.

**Post-trigger delay (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Specifies a hold time between an external trigger and the start of the actual pulse sequence events. For example, in cardiac triggered imaging, `hold` provides a mechanism for offsetting the start of the sequence by a variable amount to obtain images at different times in the cardiac cycle.

**See also:** *VnmrJ Imaging NMR*

**Related:**
- `ticks` Number of trigger pulses (P)

**Set up parameters for HOM2DJ pulse sequence (M)**

**Description:** Sets up a HOM2DJ (homonuclear J-resolved 2D) experiment.

**See also:** *VnmrJ Liquids NMR*

**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.

**Proton homonuclear decoupler present (P)**

**Applicability:** *MERCURYplus/Vx* systems.

**Description:** Sets whether the proton homonuclear decoupler board is present and communicating.

**Values:**
- `'y'` indicates the proton homonuclear decoupler board is present and communicating. This is the default.
- `'n'` disables the board. If `homdec='n'`, no communication with the board is possible: if the board is on, it will stay on, and if it is off, it will stay off.

**Homodecoupling control for first decoupler (P)**

**Description:** Enables time-shared decoupling. Unlike the `dm`, `dmm`, and `hs` parameters, `homo` is not under “status” control. On systems with type 2 or 3 interface board (`apinterface=2` or `apinterface=3`), `homo` does not control any signal routing; the position of the relevant relays is controlled by whether homonuclear decoupling (`tn` equals `dn`) or heteronuclear decoupling (`tn` not equal to `dn`) is in effect.

**Values:** On *UNITY/INOVA*, the values are `'n'` or `'y'`, where:
- `'n'` specifies no gating.
- `'y'` specifies that the receiver is gated, which is done by controlling the observe L.O. (local oscillator) line. If `dm='y'`, first decoupler rf, amplifier (blanked/unblanked), and preamplifier are gated. If
dm = 'n', no gating of these signals takes place. When homo is set to 'y',
dmm should be set to 'c' for continuous wave (CW) modulation.

**homo2d**

**Execute protocol actions of apptype homo2d (M)**

**Description:** This macro is used to execute the protocol actions of the hetero2d **apptype**.

**Examples:**
- `homo2d('setup')` execute homo2d experimental setup
- `homo2d('process')` execute homo2d processing
- `homo2d('plot')` execute homo2d plotting

**homo2**

**Homodecoupling control for second decoupler (P)**

**Applicability:** Systems with a second decoupler.

**Description:** Equivalent to the parameter homo. It works in conjunction with the parameters dm2 and dmm2.

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

**See also:** VnmrJ Liquids NMR

**Related:**
- **dm2** Decoupler mode for second decoupler (P)
- **dmm2** Decoupler modulation mode for second decoupler (P)
- **dn2** Nucleus for second decoupler (P)
- **homo** Homodecoupling control for first decoupler (P)

**homo3**

**Homodecoupling control for third decoupler (P)**

**Applicability:** Systems with a third decoupler.

**Description:** Equivalent to the parameter homo. It works in conjunction with the parameters dm3 and dmm3.

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

**See also:** VnmrJ Liquids NMR

**Related:**
- **dm3** Decoupler mode for third decoupler (P)
- **dmm3** Decoupler modulation mode for third decoupler (P)
- **dn3** Nucleus for third decoupler (P)
- **homo** Homodecoupling control for first decoupler (P)

**homo4**

**Homodecoupling control for fourth decoupler (P)**

**Applicability:** Systems with a deuterium decoupler channel as the fourth decoupler.

**Description:** Equivalent to the parameter homo. It works in conjunction with the parameters dm4 and dmm4.

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

**See also:** VnmrJ Liquids NMR

**Related:**
- **dm4** Decoupler mode for fourth decoupler (P)
- **dmm4** Decoupler modulation mode for fourth decoupler (P)
- **dn4** Nucleus for fourth decoupler (P)
- **homo** Homodecoupling control for first decoupler (P)

**houte**

**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: VnmrJ Liquids NMR

Related:
alfa  Set alpha delay before acquisition (P)
calpha  Recalculate alpha so that first-order phase is zero (M)
rof2  Receiver gating time following pulse (P)

\textbf{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: VnmrJ Liquids NMR

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)

\textbf{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 \text{bc}, depend on the correct integration: \text{bc} can either fail or it can make broad, unintegrated lines disappear from the spectrum. \text{hregions} was specifically designed for proton spectra and should not be used for other types of spectra. The result of \text{hregions} also depends on the lineshape and the signal-to-noise ratio of a spectrum.

See also: VnmrJ Liquids NMR

Related:
bca  1D and 2D baseline correction (C)
integrate  Automatically integrate 1D spectrum (M)

\textbf{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. \text{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 \text{d1} and \text{d2}): \text{hs}='yn' gives a homospoil pulse at the beginning of \text{d1}, \text{hs}='ny' gives a pulse during \text{d2}, and \text{hs}='yy' gives homospoil pulses during both \text{d1} and \text{d2}. The desired value is generally \text{hs}='nn'.

See also: VnmrJ Liquids NMR

Related:
d1  First delay (P)
d2  Incremented delay in 1st indirectly detected dimension (P)
hst  Homospoil time (P)

\textbf{hsqc  Set up parameters for HSQC pulse sequence (M)}

Applicability: Not supplied with \text{MERCURY}plus/\text{Vx} systems.

Description: Sets up parameters for a heteronuclear Overbodenhausen experiment using reverse INEPT.
See also: *VnmrJ Liquids NMR*

**Hsqc**

*Convert the parameter to a HSQC experiment (M)*

Description: Convert the parameter to a HSQC experiment.

**HSQC**

*Set up parameters for HSQC experiment (M)*

Description: Converts the current parameter set to a $^{13}$C 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.

**Hsqctoxy**

*Convert parameters to a HSQCTOXY experiment (M)*

Description: Convert the parameter to a HSQCTOXY experiment.

**HSQCTOXY**

*Set up parameters for HSQCTOXY experiment (M)*

Description: Converts the current parameter set to a $^{13}$C 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)*

Applicability: Not supplied with *MERURYplus/Vx* systems.

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 CONFIG window opened from config), 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 srates 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.

**Values:**
- 'n' makes srates unmodified by acquisition and turns off the rotor speed display in Acqstat.
- 'y' makes the hardware information from the rotor synchronization board update srates 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)
- srates: Spinning speed (P)
- xpolar1: Set up parameters for XPOLAR1 pulse sequence (M)

**hs**

**Description:** Controls pulse length if homospoil is activated by the hs parameter.

**Values:**
- On UNITY/INOVA, 0 to 20 ms (limited by hardware).
- On MERCURYplus/Vx, 0 to 20 ms (limited by software, 8 ms is standard).

**hzmm**

**Description:** Contains the quotient of wp divided by wc, a scaling factor useful for plotting. hzmm applies to 1D only.

**See also:** VnmrJ Liquids NMR

**Related:**
- wc: Width of chart (P)
- wp: Width of plot (P)

**hztomm**

**Description:** Converts 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>

**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: VnmrJ Liquids NMR
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)
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.
ihwinfo  

Hardware status of UNITY/INOVA console (U)

Applicability: UNITY/INOVA consoles (not available for any other type of console).

Syntax: (From UNIX) ihwinfo('startup'|'abort')

Description: Displays status of digital hardware in the UNITY/INOVA 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: VnmrJ Liquids NMR

Related:

abortallacqs  Reset acquisition computer in a drastic situation (C)

showconsole  Show UNITY/INOVA console configuration parameters (U)

il  

Interleave arrayed and 2D experiments (P)

Description: Controls experimental interleaving in arrayed experiments. When interleaving is active, bs transients are performed for each member of the array, followed by bs more transients for each member of the array, and so on, until nt transients have been collected for each member of the array. Thus, il is only relevant if bs is less than nt.

Values: 'y' turns on interleaving and 'n' turns off interleaving.

See also: VnmrJ Liquids NMR

Related:

bs  Block size (P)

nt  Number of transients (P)

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 \( \text{ilfid} \) command:

\[
11, 12, 13, 14; 21, 22, 23, 24; 31, 32, 33, 34
\]

Data after the \( \text{ilfid} \) command:

\[
11, 21, 31, 12, 22, 32, 13, 23, 33, 14, 24, 34
\]

See also: \textit{VnmrJ Liquids NMR}

Related:
- \textit{nf} Number of FIDs (P)
- \textit{np} Number of data points (P)
- \textit{sw} Spectral width in directly detected dimension (P)

\textbf{image} \quad \textbf{Display noninteractive gray scale image (M)}

Applicability: Systems with imaging capabilities.

Description: Brings up a \textit{dcon} 2D display of an image (using grayscale and linear scaling of the intensity) that can be used for adjusting the display while using \textit{dconi}.

See also: \textit{VnmrJ Imaging NMR}

Related:
- \textit{dcon} Display noninteractive color intensity map (C)
- \textit{dconi} Interactive 2D data display (C)
- \textit{dconn} Display color intensity map without erasing screen (C)

\textbf{image} \quad \textbf{Control phase encoding gradient in EPI experiments (P)}

Applicability: Systems with echo planar imaging (EPI) capabilities.

Description: Turns on and off the phase encoding gradient in EPI experiments. \textit{image} also specifies the number of EPI images to collect in an arrayed experiment.

Values: 0 specifies that the phase encoding gradient is turned off.
1 specifies that the phase encoding gradient is turned on.

Examples: \textit{image} = 0, 1, 1, 1 collects a set of four EPI images. The first dataset refers to the reference scan.

See also: \textit{VnmrJ Imaging NMR}

\textbf{imageprint} \quad \textbf{Plot noninteractive gray scale image (M)}

Description: Sends to the plotter a \textit{dcon} color intensity map with linear instead of logarithmic increments and with grayscale instead of colors.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \textit{dcon} Display noninteractive color intensity map (C)
- \textit{image} Display noninteractive gray scale image (M)

\textbf{imark} \quad \textbf{Annotate an image display (M)}

Applicability: Systems with imaging capabilities.

Syntax: \texttt{imark(string<,color>)}

Description: Used to label an image display with characters or strings in any color provided by the \textit{write} command. The labeling is only available inside the axis box of the image and is directed by the 2D cursors.

Arguments: \textit{string} is a text string.
\textit{color} is color of the text on a color display: \texttt{'red'}, \texttt{'yellow'}, \texttt{'green'}, \texttt{'cyan'}, \texttt{'blue'}, \texttt{'magenta'}, and \texttt{'white'}. The default is \texttt{'yellow'}. 
Examples: `imark('Muscle','red')`

See also: *VnmrJ Imaging NMR*

Related: `write` Write formatted text to a device (C)

**imcalc**

**Calculate 2D phasefiles (M,U)**

Applicability: Systems with imaging capabilities.

Syntax: (From VnmrJ) `imcalc(optype,phf1,<phf2,outphf,args>)`  
(From UNIX) `imcalc optype phf1 <phf2 outphf args>`

Description: Provides a means, along with the supporting macros, of performing arithmetic operations at a pixel-by-pixel basis on images. As operands, phasefiles are required that have been previously saved with the VnmrJ command `svphf`. A new phasefile is generated that represents the result of the selected action.

The UNIX program `imcalc` may be called from a UNIX shell using syntax 1, or called from VnmrJ with the macro `imcalc` using syntax 2. The macro `imcalci` serves as an interactive interface to the `imcalc` macro by prompting for any required inputs, which vary with the operation type. For unary operations, such as `log`, `imcalci` uses the phasefile resident in the current experiment by default.

Arguments: `optype` can be any of the following keywords (place single quotes around the keyword when entering `imcalc` from VnmrJ):

- `abs` takes the absolute value of an image.
- `add` adds two images.
- `addc` adds a constant value to each pixel in an image.
- `clipmax` sets pixel values above a user-supplied maximum to zero.
- `clipmin` sets pixel values below a user-supplied minimum to zero.
- `div` divides the first image by the second.
- `exp` sets the antilog of an image: \(10^{\text{image}}\).
- `f1roll` wraps an image in the \(f_1\) direction a selected number of pixels.
- `f2roll` wraps an image in the \(f_2\) direction a selected number of pixels.
- `flip_diag` flips an image about \(x=y\) diagonal (square images only).
- `flip_horiz` flips an image about the central horizontal axis.
- `flip_vert` flips an image about the central vertical axis.
- `gmean` sets the geometric mean of two images: \(\sqrt{\text{image}_1 \times \text{image}_2}\).
- `hline` replaces a selected horizontal trace by the average of the two adjacent traces.
- `log` sets a logarithm of an image: \(\log(\text{image})\).
- `mean` sets the arithmetic mean of two images: \(\frac{\text{image}_1 + \text{image}_2}{2}\).
- `mult` multiplies two images.
- `multc` multiplies each pixel in an image by a constant value.
- `phase` computes a resultant image from the phase angle determined by the arctangent of two orthogonal component images.
- `pow` sets exponentiation of an image (\(\text{image}^{\text{constant}}\)). To invert an image (1/pixel), use `pow` with an exponent of -1. To get a square root image, use `pow` with an exponent of 1/2.
- `reverse` sets linear inversion of pixel intensities in an image.
- `rotate_90` rotates an image clockwise 90° (square images only).
- `rotate_180` rotates an image 180°.
- `sub` subtracts the second image from the first (use add with a negative multiplier in direct call to UNIX `imcalc` program).
- `thresh` compresses all pixel values above a selected threshold to 1, and below to 0.
- `thresh2` compresses all pixel values above a user-supplied minimum and below a user-supplied maximum to 1, all others to 0.
- `vadd` adds two orthogonal “component” images to form the vector sum:
  \[ \sqrt{\text{image}_1^2 + \text{image}_2^2} \].
- `vline` replaces a selected vertical trace by the average of the two adjacent traces.

Examples: (From UNIX) `imcalc add phf1 phf2 outphf 0.5`
(From VnmrJ) `imcalc('add','phf1','phf2','destphf' 0.5)

See also: *VnmrJ Imaging NMR*

Related:
- `add` Add current FID to add/subtract experiment (C)
- `makephf` Transform and save images as phasefiles (M)
- `spadd` Add current spectrum to add/subtract experiment (C)
- `svphf` Save phasefiles (C)

**`imcalci`**

**Format arguments for `imcalc` macro (M)**

Applicability: Systems with imaging capabilities.

Syntax: `imcalci(optype)`

Description: Interactively formats arguments for the `imcalc` macro from prompted user inputs.

Arguments: `optype` has the same values as `optype` for the `imcalc` macro.

Examples: `imcalci('add')`

See also: *VnmrJ Imaging NMR*

Related: `imcalc` Calculate 2D phasefiles (M,U)

**`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')`.

**`imfit`**

**Fit arrayed imaging data to $T_1$ or $T_2$ exponential data (M,U)**

Applicability: Systems with imaging capabilities.

Syntax: (From VnmrJ) `imfit('t1'|'t2',basename,min_threshold)`
(From UNIX) `imfit t1|t2 basename min_threshold` `time1 time2 ... timeN`

Description: Performs fitting at each pixel to exponential $T_1$ or $T_2$ data. The `imfit` macro from VnmrJ provides a convenient link to the UNIX `imfit` fitting procedure by setting up and passing the correct arguments to the external program. If data cannot be handled by the VnmrJ macro, the UNIX `imfit` command can be called directly.

Three synthetic images are created by the `imfit` program, and placed in the `planes` directory of the current experiment. The $T_1$ or $T_2$ image are named...
basename1 or basename2. An error image basenamesigma represents the standard deviation of the fit at each pixel, and a \( t=0 \) image, basename0, represent the intercept of the original data at zero time.

The imfit macro automatically extracts the timing values for each array element in the data set from whichever parameter has been arrayed, providing these times to the fitting routine. For this reason, the imfit macro does not function properly if more than one parameter is arrayed.

Two macros, t1image and t2image, are provided to do all of the preprocessing required for fitting. They query for the base phasefile names and lower-limit noise threshold, transform and save all of the images, and call the imfit macro to complete the fitting process.

\( T_1 \) fitting type requires phase-sensitive images progressing from negative to positive in the normal inversion-recovery model.

Arguments: 't1' and 't2' are keywords for the fitting type, either 't1' for inversion-recovery or 't2' for decaying exponential ('t2' can also be used for saturation-recovery data).

basename is the name of a phasefile that represents the arrayed set of images. The phasefile should reside in the planes directory and must end in consecutive integer extensions, starting with 1.

min_threshold is a value for the lower limit for the fitting program. Pixels whose values in the first image are less than this threshold will not be fit and will be assigned values of zero in the synthesized resultant images.

See also: VnmrJ Imaging NMR

Related: makephf Transform and save images as phasefiles (M)
        t1image Fit arrayed imaging data to \( T_1 \) exponential data (M)
        t2image Fit arrayed imaging data to \( T_2 \) exponential data (M)
        vs Vertical scale (P)

**imprep**

Set up rf pulses, imaging and voxel selection gradients (M)

Applicability: Systems with imaging capabilities.

Description: Sets up rf pulses, imaging gradients, and voxel selection gradients as required by the application, thus providing a universal “one pass” set up of rf power and gradient levels after sequence timing, field of view, and voxel selection parameters have been chosen. imprep scans the configuration parameter lists plist and sslist to determine which rf pulse parameters and gradients are active and then proceeds to set up parameter values.

See also: VnmrJ Imaging NMR

Related: plist Active pulse length parameter list (P)
        sslist Conjugate gradient list (P)

**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 \texttt{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 \texttt{spin} is set to a particular value and the spin speed goes out of regulation.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{spin} \hspace{1cm} Sample spin rate (P)

\textbf{inadqt} \hspace{1cm} \textbf{Set up parameters for INADEQUATE pulse sequence (M)}

\textbf{Description:} Sets up parameters for 2D INADEQUATE (Incredible Natural Abundance Double-Quantum Transfer Experiment).

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{foldcc} \hspace{1cm} Fold INADEQUATE data about 2-quantum axis (C)

\textbf{index2} \hspace{1cm} \textbf{Projection or 3D plane index selected (P)}

\textbf{Applicability:} All systems; however, although \texttt{index2} is available on MERCURY\textsuperscript{plus}Vx such systems can only process 3D data and cannot acquire 3D data.

\textbf{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.”

\textbf{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: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{dplane} \hspace{1cm} Display a 3D plane (M)

\texttt{dproj} \hspace{1cm} Display a 3D plane projection (M)

\texttt{nextpl} \hspace{1cm} Display the next 3D plane (M)

\texttt{prevpl} \hspace{1cm} Display the previous 3D plane (M)

\texttt{select} \hspace{1cm} Select a spectrum or 2D plane without displaying it (C)

\textbf{inept} \hspace{1cm} \textbf{Set up parameters for INEPT pulse sequence (M)}

\textbf{Description:} Sets up parameters for the INEPT (Insensitive Nuclei Enhanced by Polarization Transfer) experiment.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{ppcal} \hspace{1cm} Proton decoupler pulse calibration (M)

\textbf{initialize_iterate} \hspace{1cm} \textbf{Set iterate string to contain relevant parameters (M)}

\textbf{Description:} Takes the current spin system (contained in \texttt{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, \texttt{iterate} is set to 'A, JAB, JAC, B, JBC, C').

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{iterate} \hspace{1cm} 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: *VnmrJ Liquids NMR*

Related: `dlini`  
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: *VnmrJ Liquids NMR*

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 \((\text{integral value}) \times \frac{\text{ins}}{\text{insref}} \times \frac{1}{\text{fn}}\). 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 \((\text{integral value}) \times \frac{\text{ins}}{\text{fn}}\).

See also: *VnmrJ Liquids NMR*

Related: `fn`  
Fourier number in directly detected dimension (P)
- `ins`  
Integral normalization scale (P)
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 \( \text{volume} \times \frac{\text{ins2}}{\text{ins2ref}} \times \frac{\text{fn}}{\text{fn1}} \). If \( \text{ins2ref} \) is “not used”, sum
of all volumes is \( \text{ins2} \). The “not used” mode is equivalent to a normalized
volume mode. If \( \text{ins2ref} \) is zero or not defined, the reported volume is
\( \text{volume} \times \frac{\text{ins2}}{\text{fn}} \times \frac{\text{fn1}}{\text{fn}} \).

See also: VnmrJ Liquids NMR

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 \( \text{insert} \).

See also: VnmrJ Liquids NMR

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 \( \text{inset} \), run the \( \text{ds} \) 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 \( \text{inset} \),
you can shift the displayed spectrum, expand it, or even contract it with the left
and right mouse buttons.

See also: VnmrJ Liquids NMR

Related:
- ds  Display a spectrum FID (C)

integ  Find largest integral in a specified region (C)

Syntax: \( \text{integ}<(\text{highfield,lowfield})><:\text{size},\text{value}> \)

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: \( \text{highfield} \) and \( \text{lowfield} \) are the limits of the region. The default values
are the parameters \( \text{sp} \) and \( \text{sp+wp} \), respectively.

\( \text{size} \) is a return value with the size of the largest integral. The size depends on
the value of the parameter \( \text{is} \) and can be positive or negative.

\( \text{value} \) is a return argument with the value of the largest integral. This value
depends on \( \text{ins}, \text{insref}, \) and \( \text{fn} \), and is independent of \( \text{is} \).

Examples: \( \text{integ}:r1,r2 \)
\( \text{integ}(500,1000):$height \)
\( \text{integ}(100+\text{sp},300+\text{sp}):$ht,$val \)
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: VnmrJ Liquids NMR
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)

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: VnmrJ Liquids NMR
Related: plc Plot carbon spectrum (M)
plh Plot proton spectrum (M)
plp Plot phosphorus spectrum (M)

intvast  
Produces 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: VnmrJ Liquids NMR
Related: pintvast Plot the integrals (M)

iplan  
Open interactive image planning tools (M)

Applicability: Systems with imaging capabilities.
Description: *iplan* is an interactive image planning server loop with drawn-on screen control buttons. It captures mouse control in VnmrJ so that you click the screen Exit button to leave. The server opens the *tbox* transverse slice specification tool. By choosing a button in the graphics area, *tbox* can be stretched, tilted, and moved. The number of slices and the area that they cover can also be adjusted. The Exit button calls the *rsliceplan* macro to load these settings for the next images.

See also: *VnmrJ Imaging NMR*

Related: *sliceplan*  Set slice parameters for target slice (M)

Related: *tbox*  Draw a tilted box (C)

### io

**Integral offset (P)**

**Description:** Offset of the integral with respect to the spectrum.

**Values:** 0 to 200, in mm.

See also: *VnmrJ Liquids NMR*

### ir

**Inversion recovery mode (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Specifies whether to run in inversion recovery mode or in normal mode. In inversion recovery mode, the parameters *pipat*, *tpwri*, *pi*, and *ti* become active, providing a prepulse and delay for inversion recovery experiments.

**Values:** ‘n’ specifies normal mode and ‘y’ specifies inversion recovery mode.

See also: *VnmrJ Imaging NMR*

Related: *pi*  Width of an inversion pulse (P)

Related: *pipat*  Shape of an inversion pulse (P)

Related: *ti*  Second delay in an inversion recovery sequence (P)

Related: *tpwri*  Intensity of an inversion pulse in dB (P)

### 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: *VnmrJ Liquids NMR*

Related: *ins*  Integral normalization scale (P)

Related: *ins2*  2D volume value (P)

Related: *insref*  Fourier number scaled value of an integral (P)

Related: *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: VnmrJ Liquids NMR

Related:

is
Integral scale (P)
isadj
Automatic integral scale adjustment (M)
isadj2
Automatic integral scale adjustment by powers of two (M)

isadj2

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: VnmrJ Liquids NMR

Related:
is
Integral scale (P)
isadj
Automatic integral scale adjustment (M)

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: VnmrJ Liquids NMR

Related: initialize_iterate Set iterate string to contain relevant parameters (M)
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR; VnmrJ Walkup NMR
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:** *VnmrJ Liquids NMR*

**Related:**
- `cexp` Create an experiment (M)
- `delexp` Delete an experiment (M)
- `jexp` Join existing experiment (C)
- `unlock` Remove inactive lock and join experiment (C)

**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:** *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**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 \texttt{jplotscale} applies the adjusted parameters to the plot.

\textbf{See also:} \textit{VnmrJ Liquids NMR}

\textbf{Related:} \texttt{jplot} \hspace{1cm} \text{Plot from Plot Designer program (C)}
\texttt{jplotscale} \hspace{1cm} \text{Scale plot parameters (M)}

\textbf{\texttt{jprint}} \hspace{1cm} \textbf{Prints the selected images to a printer or file (M)}
\textbf{Description:} The \texttt{jprint} macro takes the value of the parameters \texttt{printregion}, \texttt{printsend}, \texttt{printfile}, \texttt{printlayout}, \texttt{printformat}, \texttt{printsize}.

\textbf{\texttt{jumpret}} \hspace{1cm} \textbf{Set up parameters for JUMPRET pulse sequence (M)}
\textbf{Applicability:} Sequence is not supplied with \textit{MERCURYplus/Vx}.
\textbf{Description:} Sets up parameters for a jump-and-return water suppression sequence.
\textbf{See also:} \textit{VnmrJ Liquids NMR}

\textbf{\texttt{jwin}} \hspace{1cm} \textbf{Activate and record activity in current window (M)}
\textbf{Syntax:} \texttt{jwin(pane\_number)}
\textbf{Description:} Activates and records the activity in a specific window pane, created by \texttt{setgrid}, in the VnmrJ graphics window. \texttt{jwin} is executed when you double-click the left mouse button in a multiple-paned graphics window.
\textbf{Arguments:} \texttt{pane\_number} is the number of the pane to join.
\textbf{Examples:} \texttt{jwin(2)}
\textbf{See also:} \textit{VnmrJ Liquids NMR}
\textbf{Related:} \texttt{curwin} \hspace{1cm} \text{Current window (P)}
\texttt{fontselect} \hspace{1cm} \text{Open FontSelect window (C)}
\texttt{mapwin} \hspace{1cm} \text{List of experiment numbers (P)}
\texttt{setgrid} \hspace{1cm} \text{Activate selected window (M)}
\texttt{setwin} \hspace{1cm} \text{Activate selected window (C)}
**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:**  
*VnmrJ Liquids NMR*

**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 `pscale page` is followed by a print command `ptext (vnmruser+ '/psglib/noesy.c')`, entering `killplot` deletes both jobs.
**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 reinitialize 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:** VnmrJ Liquids NMR

**Related:**
- **killplot**  Stop plot jobs and remove from plot queue (M)
- **ptext**  Print out a text file (M)
- **showplotq**  Display plot jobs in plot queue (M)

**kind**  
**Kinetics analysis, decreasing intensity (M)**

**Description:** If the signal decreases exponentially toward a limit, the output is matched by

\[ I = A1 \times \exp(-T/TAU) + A3 \]

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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**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

\[ I = -A1 \times \exp(-T/TAU) + A3 - A1 \]

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**.
kinis

Kinetics analysis, increasing intensity, short form (M)

Description: Produces a summary of the results from kini.

See also: VnmrJ Liquids NMR

Related:
- kind: Kinetics analysis, decreasing intensity (M)
- kini: Kinetics analysis, increasing intensity (M)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lastlk</td>
<td>Last lock solvent used (P)</td>
</tr>
<tr>
<td>lastmenu</td>
<td>Menu to display when Return button is selected (P)</td>
</tr>
<tr>
<td>latch</td>
<td>Frequency synthesizer latching (P)</td>
</tr>
<tr>
<td>lb</td>
<td>Line broadening in directly detected dimension (P)</td>
</tr>
<tr>
<td>lb1</td>
<td>Line broadening in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>lb2</td>
<td>Line broadening in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>lc1d</td>
<td>Pulse sequence for LC-NMR (M)</td>
</tr>
<tr>
<td>lcpar2d</td>
<td>Create 2D LC-NMR acquisition parameters (M)</td>
</tr>
<tr>
<td>lceak</td>
<td>Peak number (P)</td>
</tr>
<tr>
<td>lcplot</td>
<td>Plot LC-NMR data (M)</td>
</tr>
<tr>
<td>lcpsgset</td>
<td>Set up parameters for various LC-NMR pulse sequences (M)</td>
</tr>
<tr>
<td>lcset2d</td>
<td>General setup for 2D LC-NMR experiments (M)</td>
</tr>
<tr>
<td>left</td>
<td>Set display limits to left half of screen (C)</td>
</tr>
<tr>
<td>legrelay</td>
<td>Independent control of magnet leg relay (P)</td>
</tr>
<tr>
<td>length</td>
<td>Determine length of a string (C)</td>
</tr>
<tr>
<td>lf</td>
<td>List files in directory (C)</td>
</tr>
<tr>
<td>liamp</td>
<td>Amplitudes of integral reset points (P)</td>
</tr>
<tr>
<td>lifrq</td>
<td>Frequencies of integral reset points (P)</td>
</tr>
<tr>
<td>listenoff</td>
<td>Disable receipt of messages from send2Vnmr (M)</td>
</tr>
<tr>
<td>listenon</td>
<td>Enable receipt of messages from send2Vnmr (M)</td>
</tr>
<tr>
<td>lkof</td>
<td>Track changes in lock frequency (P)</td>
</tr>
<tr>
<td>ll2d</td>
<td>Automatic and interactive 2D peak picking (C)</td>
</tr>
<tr>
<td>ll2dbackup</td>
<td>Copy current ll2d peak file to another file (M)</td>
</tr>
<tr>
<td>ll2dmode</td>
<td>Control display of peaks picked by ll2d (P)</td>
</tr>
<tr>
<td>llamp</td>
<td>List of line amplitudes (P)</td>
</tr>
<tr>
<td>llfrq</td>
<td>List of line frequencies (P)</td>
</tr>
<tr>
<td>ln</td>
<td>Find natural logarithm of a number (C)</td>
</tr>
<tr>
<td>load</td>
<td>Load status of displayed shims (P)</td>
</tr>
<tr>
<td>loadcolors</td>
<td>Load colors for graphics window and plotters (M)</td>
</tr>
<tr>
<td>loadPrescription</td>
<td>Load prescription (C)</td>
</tr>
<tr>
<td>loc</td>
<td>Location of sample in tray (P)</td>
</tr>
<tr>
<td>location</td>
<td>Get coordinate information from an image display (M)</td>
</tr>
<tr>
<td>lock</td>
<td>Submit an Autolock experiment to acquisition (C)</td>
</tr>
<tr>
<td>lockacqtc</td>
<td>Lock loop time constant during acquisition (P)</td>
</tr>
<tr>
<td>lockfreq</td>
<td>Lock frequency (P)</td>
</tr>
<tr>
<td>lockgain</td>
<td>Lock gain (P)</td>
</tr>
<tr>
<td>lockphase</td>
<td>Lock phase (P)</td>
</tr>
<tr>
<td>lockpower</td>
<td>Lock power (P)</td>
</tr>
<tr>
<td>locktc</td>
<td>Lock time constant (P)</td>
</tr>
<tr>
<td>logate</td>
<td>Transmitter local oscillator gate (P)</td>
</tr>
<tr>
<td>lookup</td>
<td>Look up words and lines from a text file (C)</td>
</tr>
<tr>
<td>lp</td>
<td>First-order phase in directly detected dimension (P)</td>
</tr>
<tr>
<td>lp1</td>
<td>First-order phase in 1st indirectly detected dimension (P)</td>
</tr>
</tbody>
</table>
**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:** *VnmrJ Liquids NMR*

**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)**

**Applicability:** All systems except *MERCURYplus/Vx*.

**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 CONFIG window (opened from `config`).

**Values:**
- 'n' indicates the synthesizers do not have latching capabilities (Not Present choice from the CONFIG window).
- 'y' indicates the synthesizers have latching capabilities (Present choice from the CONFIG window). This value is used with all *UNITY* *INOVA*.

**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 $\exp(-t \pi lb)$. A negative value gives a resolution enhancement function (increasing exponential) of the form $\exp(-t \pi lb)$.

- 'n' turns off line broadening and exponential weighting.

**See also:** *VnmrJ Liquids NMR*

**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 $\exp(-t \pi lb1)$. 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: VnmrJ Liquids NMR

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)

1b2 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. 1b2 works analogously to the parameter lb. 1b2 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 $\exp(-t \pi lb2)$.

A negative value gives a resolution enhancement function (increasing exponential) of the form $\exp(-t \pi lb2)$.

'n' turns off line broadening and exponential weighting.

See also: VnmrJ Liquids NMR

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 multifrequency 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.

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('lcnoesy','ds','ap','ss','d1','axis','phase')

**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 f2 digital resolution desired, in Hz/pt.
- `F1_dig_res` is the f1 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)

Applicability: All systems except *MERCURYplus/-Vx*.

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)

---

**Ll**

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: *VnmrJ Liquids NMR*

Related: `dir` List files in directory (C)
- `ls` List files in directory (C)

---

**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 \(\text{fn}\). The values displayed by the \text{dli}, \text{pir}, and \text{dpir} programs are related to \text{liamp} values by the relationship:

\[
\text{Displayed or plotted integral} = \text{liamp}[i] \cdot \text{is/}(\text{fn/128}) \cdot \text{ins}
\]

See also: \textit{VnmrJ Liquids NMR}

Related:
- \textit{display} Display parameters and their attributes (C)
- \textit{dli} Display list of integrals (C)
- \textit{dpir} Display integral amplitudes below spectrum (C)
- \textit{fn} Fourier number in directly detected dimension (P)
- \textit{liamp} Amplitudes of integral reset points (P)
- \textit{pir} Plot integral amplitudes below spectrum (C)
- \textit{lifrq} Frequencies of integral reset points (P)
- \textit{rfl} Ref. peak position in directly detected dimension (P)
- \textit{rfp} Ref. peak frequency in directly detected dimension (P)

\textbf{lifrq} \ \ \textbf{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 \textit{not} adjusted by the reference parameters \text{rfl} and \text{rfp}.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \textit{liamp} Amplitudes of integral reset points (P)
- \textit{rfl} Ref. peak position in directly detected dimension (P)
- \textit{rfp} Ref. peak frequency in directly detected dimension (P)

\textbf{listenoff} \ \ \textbf{Disable receipt of messages from send2Vnmr (M)}

Description: Deletes the file \$\text{vnmruser/.talk}, thereby disallowing \text{send2Vnmr} to send commands to \text{VnmrJ}

See also: \textit{User Programming}

Related:
- \textit{listenon} Enable receipt of messages from \text{send2Vnmr} (M)
- \textit{send2vnmr} Send a command to \text{VnmrJ} (U)

\textbf{listenon} \ \ \textbf{Enable receipt of messages from send2Vnmr (M)}

Description: Writes files with the \text{VnmrJ} port number that \text{/vnmr/bin/send2vnmr} needs to talk to \text{VnmrJ}. The command then to send commands to \text{VnmrJ} is \text{/vnmr/bin/send2vnmr \$\text{vnmruser/.talk} command}.

See also: \textit{User Programming}

Related:
- \textit{listenoff} Disable receipt of messages from \text{send2Vnmr} (M)
- \textit{send2vnmr} Send a command to \text{VnmrJ} (U)

\textbf{lkof} \ \ \textbf{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 \text{lkof} are in Hz, analogous to \text{sfrq} and \text{tof}, or \text{dfrq} and \text{dof}. \text{lkof} affects two components of the system: autolock on the console and \text{acqi} on the host computer. On \textit{UNITY/INOVA} systems, if \text{lkof} exists, it offsets the current value of the \text{lockfreq} parameter.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \textit{lockfreq} Lock frequency (P)
**112d**

**Automatic and interactive 2D peak picking (C)**

Syntax:

1. `112d (options) < $num`

2. `112d (options) < $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 `112d` 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, `112d` 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 `112d (options) < $peak_number, $f1, $f2, $amplitude, $volume, $label, $comment, $FWHH1, $FWHH2, $f1_min, $f1_max, $f2_min, $f2_max`.

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.
  - `'peaks'` is a keyword to find all peaks in the displayed area above a threshold `th`. If `dconi` is active and in the box mode, `112d` finds peaks only in the area defined by the cursors. The `'peaks'` option is the default if no arguments are entered.
  - `'pos'` or `'neg'` keywords can be used in addition to `'peak'`, `'volume'`, 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 `112d` 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 `112d` 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 `$peak_number`. This value can be used to set the `ins2ref` parameter.
- `$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 `$FWHH2` are return values set to full-width at half-height of peak `$peak_number`.
- `$f1_min$, $f1_max$, $f2_min$, $f2_max` are return values set to the bounds of peak `$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: *VnmrJ Liquids NMR*

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)
- `plll2d` Plot results of 2D peak picking (C)
- `th` Threshold (P)
- `th2d` Threshold for excluding diagonal peaks when peak picking (P)
- `xdiag` Threshold for integrating peaks in 2D spectra (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: *VnmrJ Liquids NMR*
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: *VnmrJ Liquids NMR*
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: *VnmrJ Liquids NMR*
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: *VnmrJ Liquids NMR*
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**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 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. On UNITY INOVA systems, 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: VnmrJ Liquids NMR

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)

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)

Load prescription (C)

Applicability: Systems with imaging capabilities.

Syntax: loadPrescription(char* path)

Description: Loads a prescription from a given file.

See also: VnmrJ Liquids NMR

Related: gplan Start interactive image planning (C)
**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:** *VnmrJ Liquids NMR; VnmrJ Walkup NMR*

**Related:**
- `auto_au` Controlling macro for automation (M)
- `saveglobal` Save selected parameters from global tree (P)
- `traymax` Sample changer tray size (P)

---

**location**

**Get coordinate information from an image display (M)**

**Applicability:** Systems with imaging capabilities.

**Description:** Provides coordinate information from an image display using the 2D cursor package. This program can be used, along with the interactive image viewing program `dconi`, to provide coordinate data. You should position the 2D cursor at the desired point and enter `location` in the input window. Coordinates are printed on line 3 in the VnmrJ status window. Coordinate values are supplied in both the magnet frame (X, Y, Z) and logical frame (R, P, S), where the letters R, P, and S denote read, phase encode, and slice select axes, respectively. A typical use for `location` is to set the value of the parameter `pro` for FOV position of the image center. Position the cursor at the point desired to become the new image center, enter `location`, and set the value of `pro` to the R coordinate for the logical frame.

**See also:** *VnmrJ Imaging NMR*

**Related:**
- `dconi` Interactive 2D contour display (C)
- `pro` Position of image center on the readout axis (P)

---

**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 (required on *UNITY INOVA, MERCURYplus/Vx*). `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:** *VnmrJ Liquids NMR*

**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)**

**Applicability:** All systems except **MERCURYplus/-Vx**.

**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:** On **UNITY INOVA**: 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 on a **UNITY INOVA**. 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:

```matlab
create('lockacqtc','integer','global')
setlimit('lockacqtc',4,1,1,'global')
lockacqtc=n
```

where **n** is the new value.

**See also:** *VnmrJ Liquids NMR*

**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 CONFIG window (opened from **config**). **The value of lockfreq must be set correctly in order to observe NMR signals.**

On **UNITY INOVA** systems, **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'

**UNITY INOVA, MERCURYplus/Vx** use the true **2H** frequency. Typical values of **lockfreq** are shown in the chart below. On **UNITY INOVA**, step size is approximately 2.384 Hz; on **MERCURYplus/Vx**, step size is 0.05 Hz.

<table>
<thead>
<tr>
<th>1H Frequency</th>
<th>UNITY INOVA</th>
<th>MERCURY plus/-Vx</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>30.710</td>
<td>30.6976</td>
</tr>
<tr>
<td>300</td>
<td>46.044</td>
<td>46.0625</td>
</tr>
<tr>
<td>400</td>
<td>61.395</td>
<td>61.471</td>
</tr>
<tr>
<td>500</td>
<td>76.729</td>
<td>...</td>
</tr>
<tr>
<td>600</td>
<td>92.095</td>
<td>...</td>
</tr>
<tr>
<td>750</td>
<td>115.250</td>
<td>...</td>
</tr>
</tbody>
</table>

For all systems, refer to the manual *VnmrJ Installation and Administration* for details on finding the correct lock frequency.

Commands such as **go**, **lock**, **shim**, and **su** reset the lock frequency in the console to the current value of **lockfreq**. On **UNITY INOVA, MERCURYplus/Vx**, lock frequency in the console can be set with the **sethw** command.

Note that on the **UNITY INOVA** only, **lockfreq** is offset by the value of **lkof**, if that parameter exists, but **sethw** directly uses its numeric argument, without any offset by **lkof**.
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: On `UNITY` `INOVA`, 0 to 48 dB, in 1-dB steps.
On `MERCURYplus/Vx`, 0 to 38 dB, in 1-dB steps.

See also: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: On `UNITY` `INOVA`, 0 to 68 dB, in 1-dB steps, 68 is full power.
On `MERCURYplus/Vx`, 0 to 48 dB, in 1-dB steps, 48 is full power.

See also: *VnmrJ Liquids NMR*

locktc **Lock time constant (P)**

Applicability: All systems except `MERCURYplus/Vx`.

Description: Controls lock loop time constant when system is not performing acquisition (idle, lock display, shim display, FID display, autoshim, autolock, etc.).

Values: On `UNITY` `INOVA`: 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.,

```plaintext
create('locktc','integer','global')
setlimit('locktc',4,1,'global')
locktc=2
```

See also: *VnmrJ Liquids NMR*

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)
logate  
Transmitter local oscillator gate (P)

Applicability:  
UNITY/INOVA systems.

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)

Syntax:  
lookup(options):return1,return2,...,number_returned

Description:  
Searches a text file from top to bottom for a word and returns to the user subsequent words or lines. In this context, word is defined as any string of characters delimited by "whitespace." By default, whitespace includes the space character, a tab, a newline, a carriage return, and a comma. The whitespace characters can also be specified. Therefore, a word can be a string of digits, a string of letters, or a combination of letters and digits. Punctuation marks, unless defined as whitespace (as the comma is by default), can also form words or be part of a word. A line is any string of characters from the current word to the next carriage return. A line includes all whitespace characters except the carriage return. Note that word searches are case-insensitive.

Arguments:  
options is one or more of the seven keywords ('file', 'seek', 'skip', 'read', 'readline', 'count', and 'delimiter') and other arguments used as follows:

- 'file' is a keyword to specify that the next argument is the name of the text file to be searched. If the 'file' keyword is used, it must be the first argument and the name of the file must be the second argument. 'file' resets the start of a search to the top of the text file, and subsequent searches through the file continue from where the previous search stopped, provided the 'file' keyword is not used again. Using 'file' as an argument also resets the whitespace characters back to default values.

- 'seek' is a keyword to search the text file for words that match those supplied as arguments following the 'seek' argument. When lookup is executed the first time, an implicit 'seek' is assumed as an argument. lookup maintains a pointer to the word following the last successful 'seek'. The first argument following an explicit 'seek' argument is interpreted as a word to search for, not a potential keyword. The second or later argument following an explicit 'seek' is interpreted as a keyword if it matches one of the seven lookup keywords. For example, you can search for the word file without having it interpreted as a keyword by having 'file' immediately follow the 'seek' keyword in the argument list.
'seekcs' is a keyword that is the case sensitive equivalent to the seek keyword. In all other respects, it is the same as 'seek'. One can alternate between case sensitive and case insensitive searches.

'skip' is a keyword to move the word pointer to the next word in the text file. 'skip' can optionally be followed by a number specifying how many words to skip.

'read' is a keyword to return to the user the word currently being pointed to and then move the pointer to the next word. 'read' can optionally be followed by a number specifying how many words to return.

'readline' is a keyword to return to the user the word currently being pointed to and all the following words until the end of the current line. The pointer is then moved to the first word of the next line. 'readline' can optionally be followed by a number specifying how many lines to return.

'count' is a keyword to return 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 file.

'countcs' is a keyword that is the case sensitive equivalent to the count keyword. In all other respects, it is the same as 'count'. If you use 'countcs' keyword to count the instances of the word "The", it will return the number of words that exactly match "The".

'delimiter' is a keyword to specify that the next supplied argument is a list of characters identifying the whitespace used to delimit words. Characters are specified by \n (newline), \t (tab), \r (carriage return), \\ (backslash), and \' (single quote). The arguments 'delimiter', '\t\n\r\', reselect the default whitespace. The 'file' keyword also reselects the default whitespace. The distinction is that using 'file' restarts the search from the beginning of the file while using 'delimiter' continues from the current search position. Following the 'delimiter' keyword and its argument, an implicit 'seek' is assumed.

{return1,return2,...} are words or lines returned from the search.

'number_returned' is the number of arguments returned from the file.

Examples:

```
lookup('file',systemdir + '/manual/lookup')
lookup('user','skip',2,'read',2,'readline')
:$$1, $$2, $$3, $$4, $$5
lookup('skip',8,'read','skip',3,'read',2,'seek','comma'):$$$3, $$4, $$5
lookup('delimiter','\',\'\n\t\\"\', 'seek','file','must','skip',6,'read'):$n
```

For a more detailed example of using lookup, see the text file /manual/lookup in the VnmrJ system directory (systemdir).

See also:

- User Programming
- dialog Display a dialog box from a macro (C)
- systemdir VnmrJ system directory (P)

**1p**

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

\[
\text{absorption spectrum}(\omega) = \text{real channel}(\omega) * \sin \theta + \text{imaginary channel}(\omega) * \cos \theta
\]
where the phase angle $\theta$ is a function of frequency, i.e.

$$\theta = \wp + (\omega - \omega_o) \cdot 1p$$

$\omega_o$ is defined to be the right end of the spectrum (i.e., $1p$ has zero effect at the right edge of the spectrum and a linearly increasing effect going to the left). In multidimensional data sets, $1p$ 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: *VnmrJ Liquids NMR*

**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)

---

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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 `adddpar('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.
- `'lpafft'` 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: VnmrJ Liquids NMR

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 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: VnmrJ Liquids NMR

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. 
lalg2 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: VnmrJ Liquids NMR

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)

### lpe

**Field of view size for phase-encode axis (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Specifies the actual size of the image field of view (FOV) for phase encode axis, in cm. The size and shape of the FOV is set through the selection of the parameters sw, gro, lro, sw1, gpe, and lpe. The size of the FOV in frequency units is sw*sw1, in terms of distance measure (in cm) is lro*lpe. The values of these parameters are related by the following equalities, where gcalt is the appropriate calibration constant.

\[
sw = \frac{gcal*sfrq*1000000*gro*lro}{gcal*sfrq*1000000*gpe*lpe}
\]

\[
sw1 = \frac{gcal*sfrq*1000000*lro}{gcal*sfrq*1000000*gpe*lpe}
\]
See also: *VnmrJ Imaging NMR*

**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 $(\text{strtext}-\text{lpext}) \geq 0$ for $\text{lpopt}='b'$ and by $(\text{strtext}+\text{lpext}-1) \leq \text{fn}/2$ for $\text{lpopt}='f'$. In the np direction, if $(\text{strtext}-\text{lpext})=0$ and $\text{lpopt}='b'$ (backwards linear prediction with calculation of the first point), $\text{fpmult}$ 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)

**Related:**  
- `addpar` Add selected parameters to the current experiment (M)  
- `lpext` LP data extension in np dimension (P)

**Related:**  
- `addpar` Add selected parameters to the current experiment (M)  
- `lpext` LP data extension in np dimension (P)  
- `np` Number of data points (P)

**Related:**  
- `addpar` Add selected parameters to the current experiment (M)  
- `lpext` LP data extension in np dimension (P)  
- `np` Number of data points (P)  
- `strtext` Starting point for LP data extension in np dimension (P)
in either the forward or backward direction. \texttt{lpext2} functions analogously to \texttt{lpext}. Enter \texttt{addpar('lp',2)} to create \texttt{lpext2} and other \texttt{ni2} dimension LP parameters in the current experiment.

**Related:**
- \texttt{addpar} Add selected parameters to the current experiment (M)
- \texttt{lpext} LP data extension in \texttt{np} dimension (P)
- \texttt{ni2} Number of increments in 2nd indirectly detected dimension (P)

### \texttt{lpfilt}

**LP coefficients to calculate in \texttt{np} dimension (P)**

**Description:** Specifies number of complex LP (linear prediction) coefficients in the \texttt{np} dimension to be calculated from a specified region of the time-domain data. \texttt{lpfilt} should be greater than \texttt{nsignals}, where \texttt{nsignals} is the number of sinusoidal signals contained in that FID (or interferogram). Enter \texttt{addpar('lp')} to create \texttt{lpfilt} and other \texttt{np} dimension LP parameters in the current experiment.

**Related:**
- \texttt{addpar} Add selected parameters to the current experiment (M)
- \texttt{lpalg} LP algorithm in \texttt{np} dimension (P)
- \texttt{lpfilt1} LP coefficients to calculate in \texttt{ni} dimension (P)
- \texttt{lpfilt2} LP coefficients to calculate in \texttt{ni2} dimension (P)
- \texttt{np} Number of data points (P)

### \texttt{lpfilt1}

**LP coefficients to calculate in \texttt{ni} dimension (P)**

**Description:** Specifies number of complex LP (linear prediction) coefficients in the \texttt{ni} dimension to be calculated from a specified region of the time-domain data. \texttt{lpfilt1} functions analogously to \texttt{lpfilt}. Enter \texttt{addpar('lp',1)} to create \texttt{lpfilt1} and other \texttt{ni} dimension LP parameters in the current experiment.

**Related:**
- \texttt{addpar} Add selected parameters to the current experiment (M)
- \texttt{lpfilt} LP coefficients to calculate in \texttt{np} dimension (P)
- \texttt{ni} Number of increments in 1st indirectly detected dimension (P)

### \texttt{lpfilt2}

**LP coefficients to calculate in \texttt{ni2} dimension (P)**

**Description:** Specifies number of complex LP (linear prediction) coefficients in the \texttt{ni2} dimension to be calculated from a specified region of the time-domain data. \texttt{lpfilt2} functions analogously to \texttt{lpfilt}. Enter \texttt{addpar('lp',2)} to create \texttt{lpfilt1} and other \texttt{ni2} dimension LP parameters in the current experiment.

**Related:**
- \texttt{addpar} Add selected parameters to the current experiment (M)
- \texttt{lpfilt} LP coefficients to calculate in \texttt{np} dimension (P)
- \texttt{ni} Number of increments in 1st indirectly detected dimension (P)

### \texttt{lpnupts}

**LP number of data points in \texttt{np} dimension (P)**

**Description:** Specifies number of complex time-domain data points in the \texttt{np} dimension to be used in constructing the autoregressive (\texttt{lpalg='lparfft'}) or least-squares (\texttt{lpalg='lpnefft'}) matrix from which the complex LP (linear prediction) coefficients are calculated. Note that \texttt{lpnupts} greater than or equal to \(2 \times \texttt{lpfilt}\) is required for both algorithms. Enter \texttt{addpar('lp')} to create \texttt{lpnupts} and other \texttt{np} dimension LP parameters in the current experiment.

**Related:**
- \texttt{addpar} Add selected parameters to the current experiment (M)
- \texttt{lpalg} LP algorithm in \texttt{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 **f2**: 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: *VnmrJ Liquids NMR*

**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`. Enter `addpar('lp',1)` to create `lpprint1` and other *ni* dimension LP parameters in the current experiment.

See also: *VnmrJ Liquids NMR*

**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`. Enter `addpar('lp',2)` to create `lpprint2` and other *ni2* dimension LP parameters in the current experiment.

See also: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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 lpprint2 and other ni dimension LP parameters in the current experiment.

See also: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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)

lro  Field of view size for readout axis (P)
Applicability: Systems with imaging capabilities.
Description: Specifies the actual size of the image field of view (FOV) for readout axis, in cm. The size and shape of the image FOV is set through the selection of the parameters sw, gro, lro, sw1, gpe, and lpe. The size of the FOV in frequency units is sw*sw1, or in terms of distance measure (cm) is lro*lpe. The values of these parameters are related by the following equalities, where gcal is the appropriate calibration constant:

\[
sw = \frac{gcal*sfrq*1000000*gro*lro}{gcal*sfrq*1000000*gpe*lpe}
\]

See also: VnmrJ Imaging NMR
Related: gcal Gradient calibration constant (P) gpe Phase encoding gradient increment (P) gro Readout gradient strength (P) lpe Field of view size for phase encode axis (P) sw Spectral width in directly detected dimension (P) sw1 Spectral width in 1st indirectly detected dimension (P)

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 t2 dimension in a 2D experiment or as the t3 dimension in a 3D experiment. lsfid is in the processing group and is properly handled by a wti operation (display).

Values: \(-\frac{fn}{2}\) to \(\frac{np}{2}\) (or \(-\frac{fn}{2}\) to \(\frac{fn}{2}\) 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 f2 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 f2 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 t1 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: \(-\frac{fn1}{2}\) to \(\frac{ni}{2}\) (or \(-\frac{fn1}{2}\) to \(\frac{fn1}{2}\) 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)
- `lsfrq` 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 t2 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 t2 dimension in a 2D experiment or as the t3 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)
da Display a spectrum FID (C)
f Fourier number in directly detected dimension (P)
ft Fourier transform 1D data (C)
ft1d Fourier transform along f2 dimension (C)
ft2d Fourier transform 2D data (C)
l 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)
wt Weight and Fourier transform 1D data (C)
wt1d Weight and Fourier transform f2 of 2D data (C)
wt2d 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 t3 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.
**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 t2 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*  
- *lp2*  
- *lsfid1*  
- *lsfrq*  
- *lsfrq2*  
- *ni*  
- *phfid1*  
- *wti*  

---

**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.

**Values:**
- The default value is 1.0. Larger values make the adjustments larger. Smaller values make the adjustments smaller.

**Related:**
- *cdc*  
- *lvltlt*  
- *tlt*  

---

**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*  
- *ds*  
- *lvl*
### Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maclibpath</td>
<td>Path to user's macro directory (P)</td>
</tr>
<tr>
<td>macro</td>
<td>Macro name (P)</td>
</tr>
<tr>
<td>macrocat</td>
<td>Display a user macro file in text window (C)</td>
</tr>
<tr>
<td>macrocp</td>
<td>Copy a user macro file (C)</td>
</tr>
<tr>
<td>macroadir</td>
<td>List user macro files (C)</td>
</tr>
<tr>
<td>macroedit</td>
<td>Edit a macro with user-selectable editor (M)</td>
</tr>
<tr>
<td>macrold</td>
<td>Load a macro into memory (C)</td>
</tr>
<tr>
<td>macrorm</td>
<td>Remove a user macro (C)</td>
</tr>
<tr>
<td>macrosyscat</td>
<td>Display a system macro file in text window (C)</td>
</tr>
<tr>
<td>macrosyscp</td>
<td>Copy a system macro to become a user macro (C)</td>
</tr>
<tr>
<td>macrosysdir</td>
<td>List system macros (C)</td>
</tr>
<tr>
<td>macrosysrm</td>
<td>Remove a system macro (C)</td>
</tr>
<tr>
<td>macrovi</td>
<td>Edit a user macro with the vi text editor (M)</td>
</tr>
<tr>
<td>make3dcoef</td>
<td>Make a 3D coefficients file from 2D coefficients (M)</td>
</tr>
<tr>
<td>makedosyparams</td>
<td>Create parameters for DOSY processing (M)</td>
</tr>
<tr>
<td>makefid</td>
<td>Make a FID element using numeric text input (C)</td>
</tr>
<tr>
<td>makephf</td>
<td>Transform and save images as phasefiles (M)</td>
</tr>
<tr>
<td>makeslice</td>
<td>Synthesize 2D projection of 3D DOSY experiment (C)</td>
</tr>
<tr>
<td>man</td>
<td>Display online description of command or macro (M)</td>
</tr>
<tr>
<td>managedb</td>
<td>Update user files (U)</td>
</tr>
<tr>
<td>manualpath</td>
<td>Path to user's manual directory (P)</td>
</tr>
<tr>
<td>manvi</td>
<td>Edit online description of a command or macro (M)</td>
</tr>
<tr>
<td>mapwin</td>
<td>List of experiment numbers (P)</td>
</tr>
<tr>
<td>mark</td>
<td>Determine intensity of spectrum at a point (C)</td>
</tr>
<tr>
<td>masvt</td>
<td>Type of variable temperature system (P)</td>
</tr>
<tr>
<td>maxattench1-4</td>
<td>Maximum limit for attenuator setting for rf channel 1-4 (P)</td>
</tr>
<tr>
<td>maxpen</td>
<td>Maximum number of pens to use (P)</td>
</tr>
<tr>
<td>maxsw_loband</td>
<td>Maximum spectral width of Input board (P)</td>
</tr>
<tr>
<td>md</td>
<td>Move display parameters between experiments (C)</td>
</tr>
<tr>
<td>menu</td>
<td>Change status of menu system (C)</td>
</tr>
<tr>
<td>menulibpath</td>
<td>Path to user's menu directory (P)</td>
</tr>
<tr>
<td>menuvi</td>
<td>Edit a menu with vi text editor (M)</td>
</tr>
<tr>
<td>method</td>
<td>Autoshim method (P)</td>
</tr>
<tr>
<td>mfblk</td>
<td>Move FIDs between experiments (C)</td>
</tr>
<tr>
<td>mfclose</td>
<td>Copy FID block (C)</td>
</tr>
<tr>
<td>mfdata</td>
<td>Close memory map FID (C)</td>
</tr>
<tr>
<td>mfdata</td>
<td>Move FID data (C)</td>
</tr>
<tr>
<td>mfopen</td>
<td>Memory map open FID file (C)</td>
</tr>
<tr>
<td>mftrace</td>
<td>Move FID trace (C)</td>
</tr>
<tr>
<td>minsw</td>
<td>Reduce spectral width to minimum required (M)</td>
</tr>
<tr>
<td>mkdir</td>
<td>Create new directory (C)</td>
</tr>
<tr>
<td>mlabel</td>
<td>Menu label (P)</td>
</tr>
<tr>
<td>move</td>
<td>Move to an absolute location to start a line (C)</td>
</tr>
</tbody>
</table>
maclibpath  Path to user's macro directory (P)
Description: Contains an absolute path to a user’s macro files directory. If maclibpath exists for a user, it must be defined in the global parameter file for the user. Enter the command create('maclibpath','string','global') to create maclibpath.
See also: User Programming
Related: create Create new parameter in a parameter tree (C)
        exists Determine if a parameter, file, or macro exists (C)

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)

macrocata 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)
**macrolld**  
**Load a macro into memory (C)**

**Syntax:**  
macrolld(file)<:dummy>

**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.

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 macrolld.

**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, then a directory specified by maclibpath, 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:**
macrolld('pa')
macrolld('_sw'):$noline3

See also: *User Programming*

**Related:**  
maclibpath Path to user’s macro directory (P)  
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)  
macrodire 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 -1 0' and f2coef='1 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 -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'.

The parameters f1coef and f2coef must be 2D coefficients that give proper ni and ni2 first planes with the same rp (assuming ip 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 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: *VnmrJ Liquids NMR*

### 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: *VnmrJ Liquids NMR*

### 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.

Example: `makfid('fid.in',2,'32-bit')`

See also: *VnmrJ Liquids NMR; 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)

**makephf**

Transform and save images as phasefiles (M)

Applicability: Systems with imaging capabilities.

Description: Transforms and saves images as phasefiles.

See also: *VnmrJ Imaging NMR*

Related:
- `imcalc` Calculate 2D phasefiles (M,U)
- `imfit` Fit arrayed imaging data to $T_1$ or $T_2$ exponential data (M,U)

**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 $10^{-10}$ m²/s) to be displayed.

`upperlimit` is the upper diffusion limit (in units of $10^{-10}$ m²/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: *VnmrJ Liquids NMR*

Related:
- `dosy` Process DOSY experiments (M)
- `showoriginal` Restore first 2D spectrum in 3D DOSY spectrum (M)
**man**  
Display online description of command or macro (M)  

**Syntax:**  
`man(file)`  

**Description:** Displays in the text window a description of commands and system macros from files in the directory `/vnmr/manual`.  

**Arguments:**  
`file` is the name of a command or system macro in `/vnmr/manual`.  

**Examples:**  
`man('mark')`  

**See also:**  
*VnmrJ Liquids NMR; 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:**  
*VnmrJ Liquids NMR*  

**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 the online description of commands and system macros stored in the directory `/vnmr/manual`. You must have write permission to this directory in order to edit the files.  

**Arguments:**  
`file` is the name of a command or system macro in `/vnmr/manual`.  

**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:**  
`(l) 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 f1 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 f2 axis in the 2D cursor
mode. The default is delta1.
f1_start and f1_end define region along the f1 axis in the 2D box mode.
f2_start and f2_end define region along the f₂ 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(crl,cr) cursor mode for 2D data
mark(crl,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',crl,delta1) box mode for selected 2D trace

Alternate: MARK button in the ds program.

See also: VnmrJ Liquids NMR; User Programming

Related:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cr</td>
<td>Cursor position in directly detected dimension (P)</td>
</tr>
<tr>
<td>cr1</td>
<td>Cursor position in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>curexp</td>
<td>Current experiment directory (P)</td>
</tr>
<tr>
<td>dconi</td>
<td>Interactive 2D contour display (C)</td>
</tr>
<tr>
<td>delta</td>
<td>Difference of two frequency cursors (P)</td>
</tr>
<tr>
<td>dli</td>
<td>Display list of integrals (C)</td>
</tr>
<tr>
<td>ds</td>
<td>Display a spectrum (C)</td>
</tr>
<tr>
<td>ft1d</td>
<td>Fourier transform along f₂ dimension (C)</td>
</tr>
<tr>
<td>ft2d</td>
<td>Fourier transform 2D data (C)</td>
</tr>
<tr>
<td>ins</td>
<td>Integral normalization scale (P)</td>
</tr>
<tr>
<td>ins2</td>
<td>2D volume value (P)</td>
</tr>
<tr>
<td>insref</td>
<td>Fourier number scaled value of an integral (P)</td>
</tr>
<tr>
<td>ins2ref</td>
<td>Fourier number scaled volume of a peak (P)</td>
</tr>
<tr>
<td>mv</td>
<td>Move and/or rename a file (C)</td>
</tr>
<tr>
<td>ni</td>
<td>Number of increments in 1st indirectly detected dimension (P)</td>
</tr>
</tbody>
</table>

masvt Type of variable temperature system (P)

Applicability: All systems except MERCURYplus/Vx

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
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: VnmrJ Liquids NMR

Related:
- pen Select a pen or color for drawing (C)
- setpen Set maximum number of HP plotter pens (M)

**maxsw_loband** Maximum spectral width of Input board (P)

Applicability: Systems with imaging capabilities.

Description: Stores the maximum spectral width of the Input board. The system value is set using the Max. Narrowband Width label in the CONFIG window (opened from config).

See also: VnmrJ Installation and Administration; VnmrJ Imaging NMR

Related:
- config Display current configuration and possibly change it (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: VnmrJ Liquids NMR

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 VnmrJ 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)

menulibpath Path to user’s menu directory (P)

Description: Contains an absolute path to a user’s directory of Classic VNMR menu files. If menulibpath exists for a user, it must be defined in the user’s global parameter file. To create menulibpath, enter the command
create('menulibpath','string','global').

See also: User Programming

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')
method

Autoshim method (P)

Description: Selects the method for automatic shimming. Refer to the manual *VnmrJ Liquids NMR* 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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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](#)

---

### 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)

---

### 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_loc`, 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 the 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')
```
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:  mdata(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)
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.

dest_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)
mfwrite  Move FID trace (C)
rfclose  Memory map open FID file (C)
rfwrite  Reverse FID trace (C)
rfwrite  Reverse FID block (C)
rfwrite  Reverse FID data (C)
rfwrite  Reverse FID trace (C)
**minsw**  
Reduce spectral width to minimum required (M)

Description: Searches the spectrum for peaks, sets new limits according, and then calls *movesw* to calculate a new transmitter offset *tof* and spectral width *sw*.

See also: *VnmrJ Liquids NMR*

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)

**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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)
movedsww  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 movedsww.

See also: VnmrJ Liquids NMR

Related: downsamp  Downsampling factor applied after digital filtering (P)
ds  Display a spectrum (C)
dsdsfrq  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: VnmrJ Liquids NMR

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)

movepro  Move the imaging readout position (C)

Applicability: Systems with imaging capabilities.

Description: Sets the readout position for an image or image projection to a point defined by the position of the cursor (the cr parameter).

movepro works with a 1D display (a projection or trace along F2) or 2D display, in either single cursor or box modes (only the position of the cursor in the F2 readout dimension is used; the position of the cursor in the F1 phase-encode dimension does not matter).

movepro determines the position of the cursor relative to the gradient origin and sets the parameter pro to this value, independent of image orientation. Because pro is measured in dimensional units like mm or cm, and the cursor position is stored internally in hertz, movepro works in Hz, accounting for any spectral referencing that may have been set, and converts to cm or mm to assign the value of pro.

To use movepro, display an image, image projection or trace, move the cursor to the position along the readout axis you desire to be at the center of the next image acquisition, and type movepro. This command has no effect on the value of tof (which is normally not used to define any positional information in imaging). Unlike movetof, the image or projection display will be unchanged, and no redisplay in “full” mode should be necessary.

To accurately center an image or projection, move the box cursors to the edges of the imaged object. Then use the macro split to place the cursor at the exact midpoint of the box, and type movepro.
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:** *VnmrJ Liquids NMR*

`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:** *VnmrJ Liquids NMR*

`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)

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:

\begin{itemize}
\item mp(4)
\item mp(2, 3)
\end{itemize}

See also: VnmrJ Liquids NMR

mqcosy Set up parameters for MQCOSY pulse sequence (M)

Applicability: All systems, except sequence not supplied with MERCURYplus/Vx.

Syntax: mqcosy<(level)>

Description: Sets up a multiple-quantum filtered COSY experiment.

Arguments: level is the desired quantum level of filtration.

Examples:

\begin{itemize}
\item mqcosy
\item mqcosy(3)
\end{itemize}

See also: VnmrJ Liquids NMR

mrev8 Set up parameters for MREV8 pulse sequence (M)

Applicability: Systems with a solids module. This sequence not supplied with MERCURYplus/Vx.

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

See also: User Guide: Solid-State NMR

Related:

\begin{itemize}
\item br24 Set up parameters for BR24 pulse sequence (M)
\item cylmrev Set up parameters for cycled MREV8 pulse sequence (M)
\item flipflop Set up parameters for FLIPFLOP pulse sequence (M)
\item s2pul Set up parameters for standard two-pulse sequence (M)
\end{itemize}

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:

\begin{itemize}
\item 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.
\end{itemize}

Related:

\begin{itemize}
\item fb Filter bandwidth (P)
\end{itemize}

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)

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/seglib/d2pul',
           '/vnmr/seglib/d2pul')

See also: VnmrJ Liquids NMR

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)
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: VnmrJ Liquids NMR

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

dp Double precision (P)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1, n2, n3</td>
<td>Name storage for macros (P)</td>
</tr>
<tr>
<td>nactivercvrs</td>
<td>Return number of receivers currently active (M)</td>
</tr>
<tr>
<td>nD</td>
<td>Application dimension (P)</td>
</tr>
<tr>
<td>ne</td>
<td>Number of echoes to be acquired (P)</td>
</tr>
<tr>
<td>newmenu</td>
<td>Select a menu without immediate activation (C)</td>
</tr>
<tr>
<td>newshm</td>
<td>Interactively create a shim method with options (M)</td>
</tr>
<tr>
<td>nextpl</td>
<td>Display the next 3D plane (M)</td>
</tr>
<tr>
<td>nf</td>
<td>Number of FIDs (P)</td>
</tr>
<tr>
<td>ni</td>
<td>Number of increments in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>ni2</td>
<td>Number of increments in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>ni3</td>
<td>Number of increments in 3rd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>niter</td>
<td>Number of iterations (P)</td>
</tr>
<tr>
<td>nl</td>
<td>Position cursor at the nearest line (C)</td>
</tr>
<tr>
<td>nli</td>
<td>Find integral values (C)</td>
</tr>
<tr>
<td>nlivast</td>
<td>Produces a text file of integral regions without a sum region (M)</td>
</tr>
<tr>
<td>nlivast2</td>
<td>Produces a text file with normalized integral regions (M)</td>
</tr>
<tr>
<td>nlivast3</td>
<td>Produces a text file with normalized integral regions (M)</td>
</tr>
<tr>
<td>nll</td>
<td>Find line frequencies and intensities (C)</td>
</tr>
<tr>
<td>nm</td>
<td>Select normalized intensity mode (C)</td>
</tr>
<tr>
<td>nm2d</td>
<td>Select Automatic 2D normalization (M)</td>
</tr>
<tr>
<td>noDconi</td>
<td>Disable image planning (C)</td>
</tr>
<tr>
<td>noedif</td>
<td>Convert parameters for NOE difference experiment (M)</td>
</tr>
<tr>
<td>NOESY</td>
<td>Change parameters for NOESY experiment (M)</td>
</tr>
<tr>
<td>Noesy</td>
<td>Convert the parameter to a NOESY experiment (M)</td>
</tr>
<tr>
<td>noesy</td>
<td>Set up parameters for NOESY pulse sequence (M)</td>
</tr>
<tr>
<td>NOESY1D</td>
<td>Change parameters for NOESY 1D experiment (M)</td>
</tr>
<tr>
<td>Noesy1d</td>
<td>Convert the parameter set to a Noesy1d experiment (M)</td>
</tr>
<tr>
<td>noise</td>
<td>Measure noise level of FID (C)</td>
</tr>
<tr>
<td>noisemult</td>
<td>Control noise multiplier for automatic 2D processing (M)</td>
</tr>
<tr>
<td>noislm</td>
<td>Limit noise in spectrum (M)</td>
</tr>
<tr>
<td>notebook</td>
<td>Notebook name (P)</td>
</tr>
<tr>
<td>np</td>
<td>Number of data points (P)</td>
</tr>
<tr>
<td>npoint</td>
<td>Number of points for fp peak search (P)</td>
</tr>
<tr>
<td>nrecords</td>
<td>Determine number of lines in a file (M)</td>
</tr>
<tr>
<td>ns</td>
<td>Number of slices to be acquired (P)</td>
</tr>
<tr>
<td>nscans</td>
<td>Number of scout scan or real scan repetitions (P)</td>
</tr>
<tr>
<td>nt</td>
<td>Number of transients (P)</td>
</tr>
<tr>
<td>ntrig</td>
<td>Number of trigger signals to wait before acquisition (P)</td>
</tr>
<tr>
<td>ntype3d</td>
<td>Specify whether f1 or f2 display expected to be N-type (P)</td>
</tr>
<tr>
<td>numrcvrs</td>
<td>Number of receivers in the system (P)</td>
</tr>
<tr>
<td>numreg</td>
<td>Return the number of regions in a spectrum (C)</td>
</tr>
<tr>
<td>numrfch</td>
<td>Number of rf channels (P)</td>
</tr>
<tr>
<td>nv</td>
<td>Number of phase encode steps (P)</td>
</tr>
</tbody>
</table>
**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)  

**nactivercvrs**  
**Return number of receivers currently active (M)**  
**Applicability:** Systems with multiple receivers.  
**Description:** Calculates and returns the number of receivers currently active, based on the values of the *rcvrs* and *numrcvrs* parameters.  
**Examples:**  
- `nactivercvrs:$nact` sets `$nact` to the number of currently active receivers.  
**Related:**  
- *rcvrs* Which receivers to use (P)  
- *numrcvrs* Number of receivers in the system (P)  

**nD**  
**Application dimension (P)**  
**Applicability:** Systems with the imaging capabilities.  
**Description:** Defines the dimension of the experiment performed by the application code. The value of nD is the number of FFT (fast Fourier transform) operations used to reconstruct the data or the number of independent k space coordinates encoded in the data. The nD, seqcon, plist, patlist, pwrlist, fliplist and sslist parameters configure a particular parameter set for an application sequence defined by the value of the seqfil parameter.  
**Values:** 1, 2, 3, or 4.  
**See also:** *VnmrJ Imaging NMR*  
**Related:**  
- *fliplist* Standard flip angle list (P)  
- *patlist* Active pulse template parameter list (P)  
- *plist* Active pulse length parameter list (P)  
- *pwrlist* Active pulse power level parameter list (P)  
- *seqcon* Acquisition loop control (P)  
- *seqfil* Application object code name (P)  
- *sslist* Conjugate gradient list (P)  

**ne**  
**Number of echoes to be acquired (P)**  
**Applicability:** Systems with the imaging capabilities.  
**Description:** Sets number of echoes to be acquired for multiecho sequences.  
**Values:** 1 to desired number, in integer steps.  
**See also:** *VnmrJ Imaging NMR*  
**Related:**  
- *ns* Number of slices to be acquired (P)  

**newmenu**  
**Select a menu without immediate activation (C)**  
**Syntax:**  
1. `newmenu(menu_name)`  
2. `newmenu:$current_menu`
newmenu

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,VnmrJ Liquids NMR.

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: VnmrJ Liquids NMR

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)
nextpl

Display the next 3D plane (M)

Applicability: All systems; however, although nextpl is available on MERCURYplus/Vx systems, such systems can only process 3D data and cannot acquire 3D data.

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: VnmrJ Liquids NMR
Related: dplane Display a 3D plane (M)
dproj Display a 3D plane projection (M)
dplanes 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)

nf

Number of FIDs (P)

Applicability: Systems with imaging capabilities.

Description: Number of FIDs acquired by explicit acquisition.

Values: Positive integer. For example, in the COSY-NOESY experiment, nf is 2.

See also: User Guide Imaging
Related: cf Current FID (P)

ni

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: VnmrJ Liquids NMR; 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').
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: VnmrJ Liquids NMR

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)

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: VnmrJ Liquids NMR

nl

Position cursor at the nearest line (C)

Syntax: `nl<:height<,frequency>>`

Description: Moves the cursor to the nearest calculated line position.

Arguments: 
- `height`: a return value set to the height of the line.
- `frequency`: a return value set to the frequency of the line.

Examples: 
- `nl`
- `nl:r1,r2`

See also: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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)
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:** *VnmrJ Liquids NMR*

nlivast2  
**Produces a text file with normalized integral regions (M)**

**Applicability:** Systems with VAST accessory.

**Syntax:** `nlivast2(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:** *VnmrJ Liquids NMR*

nlivast3  
**Produces a text file with normalized integral regions (M)**

**Applicability:** Systems with VAST accessory.

**Syntax:** `nlivast3(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:** *VnmrJ Liquids NMR*

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.

d 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)

**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 nm 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: VnmrJ Liquids NMR

Related:

- `ai` Select absolute intensity mode (C)
- `aig` Absolute intensity group (P)
- `vs` Vertical scale (P)

**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: VnmrJ Liquids NMR

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)

noDconi Disable image planning (C)

Applicability: Systems with imaging capabilities.

Examples: Disables image planning using the dconi display.

See also: VnmrJ Liquids NMR

Related: gplan Start interactive image planning (C)

noedif Convert parameters for NOE difference experiment (M)

Applicability: MERCURYplus/Vx systems only.

Description: Converts a $^1$H parameter set to perform the NOE (Nuclear Overhauser Enhancement) difference experiment.

See also: VnmrJ Liquids NMR

Related: setup Set up parameters for basic experiments (M)
cyclenoe Set up parameters for CYCLENOEx pulse sequence (M)

NOESY Change parameters for NOESY experiment (M)

Description: Converts the current parameter set to a NOESY experiment.

Noesy Convert the parameter to a NOESY experiment (M)

Description: Convert the parameter to a NOESY experiment.

noesy Set up parameters for NOESY pulse sequence (M)

Description: Sets up parameters for the laboratory frame Overhauser experiment or the 2D exchange experiment.

See also: VnmrJ Liquids NMR

Related: foldt Fold COSY-like spectrum along diagonal axis (C)
NOESY1D  Change parameters for NOESY1D experiment (M)
Description: Converts the current parameter set to a NOESY1D (also known as DFGSE-noe) experiment. A 1D proton spectrum is displayed to do peak selection.

Noesy1d  Convert the parameter set to a Noesy1d experiment (M)
Description: Convert the parameter set to a Noesy1d experiment.
See also: Proton(M) selld(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: VnmrJ Liquids NMR
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
nn  noisemult(10)
See also: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**Related:**  
- `nm2d` Automatic 2D normalization (M)
- `proc2d` Process 2D spectra (M)

**notebook**  
**Notebook name (P)**

**Description:** Specifies the notebook name of a sample, which is saved with a liquids study.

See also: page (P)  
`samplename (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`.

On *MERCURYplus/Vx*, 64 to 128,000, in steps of 64 (`dp` does not affect the limit because on *MERCURYplus/Vx* `dp` is always 'y').

**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: *VnmrJ Liquids NMR*

**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:** VnmrJ Liquids NMR

**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

**ns**  
**Number of slices to be acquired (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Sets the number of slices to be acquired for multislice sequences.

**Values:** 1 to desired number, in integer steps.

**See also:** VnmrJ Imaging NMR

**See also:**
- ne Number of echoes to be acquired (P)

**nscans**  
**Number of scout scan or real scan repetitions (P)**

**Applicability:** Systems with LC-NMR accessory.

**Description:** For on-flow applications, nscans is set to the number of repetitions of the scout scan or real scan process to be performed (based on the time duration of the LC run). In stopped-flow applications, nscans must be set to a number that is greater than or equal to the number of peaks to be analyzed or detected. If nscans does not exist, the parlc macro can create it.

**See also:** VnmrJ Liquids NMR

**Related:**
- curscan Scan currently in progress (P)
- parlc Create LC-NMR parameters (M)

**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 MERCURYplus/Vx, the hardware limits nt to 16e6). For an indefinite acquisition, set nt to a very large number such as 1e9.

**See also:** VnmrJ Liquids NMR; 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 does not exist, the parlc macro can create it. ntrig is not normally entered by the user.

See also: VnmrJ Liquids NMR

Related: parlc  Create LC-NMR parameters (M)

ntype3d  Specify whether f1 or f2 display expected to be N-type (P)

Applicability: All systems; however, although ntype3d is available on MERCURYplus/Vx systems, such systems can only process 3D data and cannot acquire 3D data.

Description: Indicates whether the f1 or f2 display is expected to be N-type, that is, opposite to the sense of precession defined by f3, under normal 3D processing conditions.

Values: 'yn' specifies that f1 is expected to have an N-type display under normal 3D processing conditions.

'ny' specifies that f2 is expected to have an N-type display under normal 3D processing conditions.

'yy' specifies that both f1 and f2 are expected to have N-type displays under normal 3D processing conditions. Setting ntype3d = 'yy' changes the sense of precession in f1 and f2 by negating the imaginary portion of the t1 and t2 interferograms prior to Fourier transformation.

See also: VnmrJ Liquids NMR

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 dc 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

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.

Related: rcvrs  Which receivers to use (P)

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:  

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ds</td>
<td>Display a spectrum (C)</td>
</tr>
<tr>
<td>getreg</td>
<td>Get frequency limits of a specified region (C)</td>
</tr>
<tr>
<td>region</td>
<td>Divide spectrum into regions (C)</td>
</tr>
<tr>
<td>z</td>
<td>Add integral reset point at cursor position (C)</td>
</tr>
</tbody>
</table>

**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 CONFIG window (opened from config). 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:  
For UNITY INOVA, the fifth channel can only be used with the deuterium decoupler channel.
On MERCURYplus/Vx: 2. On other systems: 1, 2, 3, 4, or 5. The value does not include the lock channel.

See also:  
VnmrJ Installation and Administration

Related:  

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Display current configuration and possibly change it (M)</td>
</tr>
<tr>
<td>dn2</td>
<td>Nucleus for the second decoupler (P)</td>
</tr>
<tr>
<td>dn3</td>
<td>Nucleus for the third decoupler (P)</td>
</tr>
<tr>
<td>dn4</td>
<td>Nucleus for the fourth decoupler (P)</td>
</tr>
</tbody>
</table>

**nv**  
Number of phase encode steps (P)

Applicability:  
Systems with imaging capabilities.

Description:  
The number of phase encode steps for the first indirectly detected dimension in a multidimensional imaging or CSI experiment.

Values:  
0 to the desired number, in powers of 2. Typical values are 0, 64, 128, and 256.

See also:  
VnmrJ Imaging NMR
**off**  
Make a parameter inactive (C)

**Syntax:**  
off(parameter<,tree>)

**Description:**  
Turns 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 (C) Create new parameter in a parameter tree  
on (C) Make a parameter active or test its state

**off**  
Calculate frequency offset of cursor (M)

**Applicability:**  
Systems with imaging capabilities.

**Syntax:**  
offset<('silent')><:parameter>

**Description:**  
Reads value of the cursor parameter cr, and then calculates and displays the transmitter offset value, in Hz, that places the cursor position on resonance.

**Arguments:**  
- **'silent**' is a keyword to not display the frequency offset value. The default is to display the value.
- **parameter** is a variable (such as the parameter tof in the example below) that, if present, is loaded with the calculated offset frequency value.

**Examples:**  
- offset  
- offset('silent'):tof

**See also:**  
VnmrJ Imaging NMR

**Related:**  
cr (P) Current cursor position

---

**C**  
Make a parameter inactive

**M**  
Calculate frequency offset of cursor

**P**  
Slice plane orientation

**Digital filter coefficients for oversampling**

**Oversampling filter for real-time DSP**

**Bandpass filter offset for oversampling**

**Frequency synthesizer overrange**

**Oversampling factor for acquisition**
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,

```vnmr
on('var1'):$e
if $e then
   write('line3','if statement is true with value of %d',$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:

```vnmr
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:

```vnmr
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)

operatorlogin

Sets workspace and parameters for the operator (M)

Syntax: operatorlogin operator email panellevel

Description: Sets the workspace and parameters for the operator being logged in.

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)

orient **Slice plane orientation (P)**

Applicability: Systems with imaging capabilities.

Description: Controls the orientation of the slice plane in the gradient reference frame.

Values: A three-character string with any permutation of the letters x, y, z, and n: 'xyz', 'zyx', 'nxz', etc. The permutation chosen determines the orientation of the slice plane. The first character is the identity of the readout gradient, the second character is the identity of the phase encoding gradient, and the third character is the identity of the slice selection gradient. The character n causes no gradient to be sent, which is used to avoid zeroing values.

For imaging modules, only 'sag' (sagittal), 'trans' (transverse), 'cor', and 'oblique' are used. The choice 'oblique' is not user-enterable. Only the macro imprep can set up oblique imaging.

See also: *VnmrJ Imaging NMR*

Related: imprep Set up rf pulses, imaging, and voxel selection gradients (M)

oscoef **Digital filter coefficients for oversampling (P)**

Description: Specifies number of coefficients used in the digital filter. If oscoef 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, osfb, osfilt, oslsfrq, and oversamp.

Values: For inline DSP (dsp='i'), the default is 7.5*oversamp. 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.

For real-time DSP (dsp='r'), the number of coefficients is not adjustable but is determined by the hardware.

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 $\text{sw}/2$ rejects frequencies at the edges of the spectrum; a value greater than $\text{sw}/2$ aliases noise and signals at frequencies outside of $\pm \text{sw}/2$.

'n' sets the bandwidth to $\text{sw}/2$.

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 AnalogPlus™ 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: UNITY/INOVA 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 CONFIG window (opened from config).

Values: Not Present, 10000 Hz, or 100000 Hz

In CONFIG, 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. In the CONFIG window the 10000 Hz or 100000 Hz choices are determined by the letters H, J, or K found in the PTS Synthesizers model number. In CONFIG, 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 in the CONFIG window. 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 (\texttt{dsp='i'}), $np \times \text{oversamp}$ data points are acquired at a rate of $sw \times \text{oversamp}$. 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 (\texttt{dsp='r'}), the oversampling, digital filtering, and downsampling 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 \texttt{addpar('oversamp')} to add it. \texttt{addpar('oversamp')} creates digital filtering and oversampling parameters \texttt{def_osfilt}, \texttt{filtfile}, \texttt{oscoef}, \texttt{osfb}, \texttt{osfilt}, \texttt{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 manuallyreset.

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., \texttt{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:

- \texttt{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:

<table>
<thead>
<tr>
<th>System</th>
<th>Maximum sw*oversamp</th>
<th>Maximum np*oversamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNITY/NOVA</td>
<td>500 kHz</td>
<td>2M</td>
</tr>
<tr>
<td>MERCURYplus/-Vx</td>
<td>100 kHz</td>
<td>128K</td>
</tr>
</tbody>
</table>

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)
`oslsafq` Bandpass filter offset for oversampling (P)
`paros` Create additional parameters used by oversampling (M)
`sw` Spectral width in directly detected dimension (P)
p1  Enter pulse width for p1 in degrees (C)

p1  First pulse width (P)

p1pat  Shape of excitation pulse (P)

p2  180° refocus pulse width (P)

p2pat  RF pulse pattern of 180° refocus pulse p2 (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)

pal1  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)

pards  Create additional parameters used by downsampling (M)

parfix  Create parameters for time-domain solvent subtraction (M)

parfix  Update parameter sets (M)

parlc  Create parameters for LC-NMR experiments (M)

parl2d  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 oversampling (M)

parstep  Parameter step size values (P)

parversion  Version of parameter set (P)

path3d  Path to currently displayed 2D planes from a 3D data set (P)

patlist  Active pulse template parameter list (P)

paxis  Plot horizontal LC axis (M)

Pbox  Pulse shaping software (U)

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)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbox_pw</td>
<td>Extract pulse length from pbox.cal or Pbox shape file (M)</td>
</tr>
<tr>
<td>pbox_pwr</td>
<td>Extract power level from Pbox.cal or Pbox shape file (M)</td>
</tr>
<tr>
<td>pbox_pwrz</td>
<td>Extract fine power level from pbox.cal or Pbox shape file (M)</td>
</tr>
<tr>
<td>pboxget</td>
<td>Extract Pbox calibration data (M)</td>
</tr>
<tr>
<td>pboxpar</td>
<td>Add parameter definition to the Pbox.inp file (M)</td>
</tr>
<tr>
<td>pboxrst</td>
<td>Reset temporary Pbox variables (M)</td>
</tr>
<tr>
<td>pboxunits</td>
<td>Converts to Pbox default units (M)</td>
</tr>
<tr>
<td>pcmapply</td>
<td>Apply phase correction map to data in EPI experiments (C)</td>
</tr>
<tr>
<td>pcmclose</td>
<td>Close phase correction map in EPI experiments (C)</td>
</tr>
<tr>
<td>pcmopen</td>
<td>Open phase correction map in EPI experiments (C)</td>
</tr>
<tr>
<td>pccon</td>
<td>Plot contours on a plotter (C)</td>
</tr>
<tr>
<td>pcss</td>
<td>Calculate and show proton chemical shifts spectrum (M)</td>
</tr>
<tr>
<td>peak</td>
<td>Find tallest peak in specified region (C)</td>
</tr>
<tr>
<td>peak2d</td>
<td>Return information about maximum in 2D data (C)</td>
</tr>
<tr>
<td>pen</td>
<td>Select a pen or color for drawing (C)</td>
</tr>
<tr>
<td>pexp1</td>
<td>Plot exponential or polynomial curves (C)</td>
</tr>
<tr>
<td>pexp2pladd</td>
<td>Add another diffusion analysis to current plot (M)</td>
</tr>
<tr>
<td>pfgon</td>
<td>Pulsed field gradient amplifiers on/off control (P)</td>
</tr>
<tr>
<td>pfww</td>
<td>Plot FIDs in whitewash mode (C)</td>
</tr>
<tr>
<td>pge</td>
<td>Convert parameter set to PGE pulse sequence (M)</td>
</tr>
<tr>
<td>pge_calib</td>
<td>Calibrate gradient strengths for PGE pulse sequence (M)</td>
</tr>
<tr>
<td>pge_data</td>
<td>Extract data from single element of PGE pulse sequence (M)</td>
</tr>
<tr>
<td>pge_output</td>
<td>Output results from PGE pulse sequence (M)</td>
</tr>
<tr>
<td>pge_process</td>
<td>Automated processing of data from PGE pulse sequence (M)</td>
</tr>
<tr>
<td>pge_results</td>
<td>Calculate diffusion constant for integral region (M)</td>
</tr>
<tr>
<td>pge_setup</td>
<td>Set up gradient control parameters for PGE pulse sequence (M)</td>
</tr>
<tr>
<td>ph</td>
<td>Set phased mode in directly detected dimension (C)</td>
</tr>
<tr>
<td>ph1</td>
<td>Set phased mode in 1st indirectly detected dimension (C)</td>
</tr>
<tr>
<td>ph2</td>
<td>Set phased mode in 2nd indirectly detected dimension (C)</td>
</tr>
<tr>
<td>phase</td>
<td>Change frequency-independent phase rp (M)</td>
</tr>
<tr>
<td>phase1</td>
<td>Phase selection (P)</td>
</tr>
<tr>
<td>phase2</td>
<td>Phase selection for 3D acquisition (P)</td>
</tr>
<tr>
<td>phase3</td>
<td>Phase selection for 4D acquisition (P)</td>
</tr>
<tr>
<td>phasing</td>
<td>Control update region during interactive phasing (P)</td>
</tr>
<tr>
<td>phfid</td>
<td>Zero-order phasing constant for the np FID (P)</td>
</tr>
<tr>
<td>phfid1</td>
<td>Zero-order phasing constant for ni interferogram (P)</td>
</tr>
<tr>
<td>phfid2</td>
<td>Zero-order phasing constant for ni2 interferogram (P)</td>
</tr>
<tr>
<td>phi</td>
<td>Euler angle phi from magnet frame (P)</td>
</tr>
<tr>
<td>Phosphorus</td>
<td>Set up parameters for $^{31}$P experiment (M)</td>
</tr>
<tr>
<td>pi</td>
<td>Inversion pulse length (P)</td>
</tr>
<tr>
<td>pi3ssbsq</td>
<td>Set up pi/3 shifted sinebell-squared window function (M)</td>
</tr>
<tr>
<td>pi4ssbsq</td>
<td>Set up pi/4 shifted sinebell-squared window function (M)</td>
</tr>
<tr>
<td>pilot</td>
<td>Automatic sequence setup (P)</td>
</tr>
<tr>
<td>pintvast</td>
<td>Plots of integral regions (M)</td>
</tr>
<tr>
<td>pipat</td>
<td>Shape of an inversion pulse (P)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>pir</code></td>
<td>Plot integral amplitudes below spectrum (C)</td>
</tr>
<tr>
<td><code>pirn</code></td>
<td>Plot normalized integral amplitudes below spectrum (M)</td>
</tr>
<tr>
<td><code>pl</code></td>
<td>Plot spectra (C)</td>
</tr>
<tr>
<td><code>pl2d</code></td>
<td>Plot 2D spectra in whitewash mode (C)</td>
</tr>
<tr>
<td><code>plan</code></td>
<td>Display menu for planning a target scan (M)</td>
</tr>
<tr>
<td><code>plane</code></td>
<td>Currently displayed 3D plane type (P)</td>
</tr>
<tr>
<td><code>planlock</code></td>
<td>Planner lock (P)</td>
</tr>
<tr>
<td><code>plapt</code></td>
<td>Plot APT-type spectra automatically (M)</td>
</tr>
<tr>
<td><code>plarray</code></td>
<td>Plotting macro for arrayed 1D spectra (M)</td>
</tr>
<tr>
<td><code>plate_glue</code></td>
<td>Define a glue order for plotting and display (U)</td>
</tr>
<tr>
<td><code>plc</code></td>
<td>Plot a carbon spectrum (M)</td>
</tr>
<tr>
<td><code>plcosy</code></td>
<td>Plot COSY- and NOESY-type spectra automatically (M)</td>
</tr>
<tr>
<td><code>pldept</code></td>
<td>Plot DEPT data, edited or unedited (M)</td>
</tr>
<tr>
<td><code>plfid</code></td>
<td>Plot FIDs (C)</td>
</tr>
<tr>
<td><code>plfit</code></td>
<td>Plot deconvolution analysis (M)</td>
</tr>
<tr>
<td><code>plgrid</code></td>
<td>Plot a grid on a 2D plot (M)</td>
</tr>
<tr>
<td><code>plh</code></td>
<td>Plot proton spectrum (M)</td>
</tr>
<tr>
<td><code>plhet2dj</code></td>
<td>Plot heteronuclear J-resolved 2D spectra automatically (M)</td>
</tr>
<tr>
<td><code>plhom2dj</code></td>
<td>Plot homonuclear J-resolved 2D spectra automatically (M)</td>
</tr>
<tr>
<td><code>plhxcor</code></td>
<td>Plot X,H-correlation 2D spectrum (M)</td>
</tr>
<tr>
<td><code>plist</code></td>
<td>Active pulse length parameter list (P)</td>
</tr>
<tr>
<td><code>plll</code></td>
<td>Plot a line list (M)</td>
</tr>
<tr>
<td><code>plll2d</code></td>
<td>Plot results of 2D peak picking (C)</td>
</tr>
<tr>
<td><code>plot</code></td>
<td>Automatically plot spectra (M)</td>
</tr>
<tr>
<td><code>plot1d</code></td>
<td>Plotting macro for simple (non-arrayed) 1D spectra (M)</td>
</tr>
<tr>
<td><code>plot2D</code></td>
<td>Plot 2D spectra (M)</td>
</tr>
<tr>
<td><code>plotside</code></td>
<td>Plot spectrum on side (M)</td>
</tr>
<tr>
<td><code>plotter</code></td>
<td>Plotter device (P)</td>
</tr>
<tr>
<td><code>plottop</code></td>
<td>Plot spectrum on top (M)</td>
</tr>
<tr>
<td><code>plottopside</code></td>
<td>Plot spectrum on top and side (M)</td>
</tr>
<tr>
<td><code>plp</code></td>
<td>Plot phosphorus spectrum (M)</td>
</tr>
<tr>
<td><code>plplanes</code></td>
<td>Plot a series of 3D planes (M)</td>
</tr>
<tr>
<td><code>pltext</code></td>
<td>Plot text file (M)</td>
</tr>
<tr>
<td><code>pltmod</code></td>
<td>Plotter display mode (P)</td>
</tr>
<tr>
<td><code>plvast</code></td>
<td>Plot VAST data in a stacked 1D-NMR matrix format (M)</td>
</tr>
<tr>
<td><code>plvast2d</code></td>
<td>Plot VAST data in a stacked pseudo-2D format (M)</td>
</tr>
<tr>
<td><code>plww</code></td>
<td>Plot spectra in whitewash mode (C)</td>
</tr>
<tr>
<td><code>pmode</code></td>
<td>Processing mode for 2D data (P)</td>
</tr>
<tr>
<td><code>poly0</code></td>
<td>Display mean of the data in regression.inp file (M)</td>
</tr>
<tr>
<td><code>pos1 - pos3</code></td>
<td>Position of voxel center (P)</td>
</tr>
<tr>
<td><code>pp</code></td>
<td>Decoupler pulse length (P)</td>
</tr>
<tr>
<td><code>ppa</code></td>
<td>Plot a parameter list in plain English (M)</td>
</tr>
<tr>
<td><code>ppcal</code></td>
<td>Proton decoupler pulse calibration (M)</td>
</tr>
<tr>
<td><code>ppe</code></td>
<td>Position of image center on 2D phase encode axis (P)</td>
</tr>
<tr>
<td><code>ppf</code></td>
<td>Plot peak frequencies over spectrum (C)</td>
</tr>
<tr>
<td><code>pph</code></td>
<td>Print pulse header (M)</td>
</tr>
<tr>
<td><code>pplvl</code></td>
<td>Proton pulse power level (P)</td>
</tr>
</tbody>
</table>
ppmm Resolution on printers and plotters (P)
pprofile Plot pulse excitation profile (M)
pps Plot pulse sequence (C)
prep prepare a scan (M)
presat Set up parameters for PRESAT pulse sequence (M)
Presat Set up parameters for presat 1H experiment (M)
presig Preamplifier signal level selection (P)
prevpl Display the previous 3D plane (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)
pro Position of image center on the readout axis (P)
probe Probe type (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)
proclplot Automatically process FIDs (M)
profile Set up pulse sequence for gradient calibration (M)
proj Project 2D data (C)
Proton Set up parameters for 1H experiment (M)
prune Prune extra parameters from current tree (C)
pscale Plot scale below spectrum or FID (C)
psdefault 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)
psi Euler angle psi from magnet frame (P)
pslabel Pulse sequence label (P)
ps Slice position (P)
pss Stack center shift along z axis (P)
**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:** *VnmrJ Liquids NMR*

**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:**
- On MERCURY+plus/Vx systems: 0, 0.2 µs to 150,000 µs, in 0.1 µs steps
- On INOVA: 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.
See also: *VnmrJ Liquids NMR*

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)

**p2**

**180° refocus pulse width (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Sets the length of the 180° refocus rf pulse.

**Values:** Number, in \( \mu \)s.

See also: *VnmrJ Imaging NMR*

Related: **p1** First pulse width (P)

**p2pat**

**RF pulse pattern of 180° refocus pulse p2 (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Contains a string for the shape of the 180° refocus pulse \( p2 \).

See also: *VnmrJ Imaging NMR*

Related: **p2** 180° refocus pulse width (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 \texttt{enter} command, then the macro \texttt{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: VnmrJ Liquids NMR

p31p

Process 1D phosphorus spectra (M)

Syntax: p31p
Description: Processes non-arrayed 1D 31P 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: VnmrJ Liquids NMR

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 pal, av1, pwr1, or ph1 to yield the resultant contour display for the 2D data.

See also: VnmrJ Liquids NMR

Related: av Set abs. value mode in directly detected dimension (C)
    dmg Data display mode in directly detected dimension (P)
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 \texttt{dmg1} to the string value 'pa1'. If the parameter \texttt{dmg1} does not exist, \texttt{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 \texttt{rp1} and \texttt{lp1}.

The \texttt{pa1} command is only needed if mixed-mode display is desired. If the parameter \texttt{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 \texttt{dmg}). For the contour display of multidimensional data, the result of \texttt{pa1} is the same as for traces provided that \texttt{pmode='partial'} or \texttt{pmode=''}.

See also: \textit{VnmrJ Liquids NMRs}

Related: \texttt{av1} \hspace{1em} Set abs. value mode in 1st indirectly detected dimension (C)  
\texttt{dmg1} \hspace{1em} Data display mode in 1st indirectly detected dimension (P)  
\texttt{lp1} \hspace{1em} First-order phase in 1st indirectly detected dimension (P)  
\texttt{pa} \hspace{1em} Set phase angle mode in directly detected dimension (C)  
\texttt{ph1} \hspace{1em} Set phased mode in 1st indirectly detected dimension (C)  
\texttt{pmode} \hspace{1em} Processing mode for 2D data (P)  
\texttt{pwr1} \hspace{1em} Set power mode in 1st indirectly detected dimension (C)  
\texttt{rp1} \hspace{1em} 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 \texttt{fn=fn1} and \texttt{sw=sw1}. Symmetrization of the data with the command \texttt{foldt} is recommended, but not required. First, select a proper threshold and perform a 2D line listing with the command \texttt{ll2d}. Next, plot the 2D data with the contour plot command \texttt{pcon}, leaving enough room at the left side of the plot for the connectivity table. Then, \texttt{pacosy} will analyze the data and plot the connectivities on the plotter. \texttt{pacosy} gets its input from the file \texttt{ll2d.out} in the current experiment directory. The command \texttt{acosy} performs the same analysis and displays the connectivities on the screen.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{acosy} \hspace{1em} Automatic analysis of COSY data (C)  
\texttt{fn} \hspace{1em} Fourier number in directly detected dimension (P)
Preacquisition delay (P)

Description: Each NMR experiment starts with a single delay time equal to \( \text{pad} \) over and above the delay \( d_1 \) that occurs before each transient. Normally, \( \text{pad} \) 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 \( \text{pad} \) 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 \( \text{pad} \) to perhaps 300 seconds. This is especially true when running experiments involving arrays of temperatures. The \( \text{pad} \) parameter is most useful for running kinetics experiments. For example, \( \text{pad}=0,3600,3600,3600,3600 \) will run an experiment immediately when \( \text{go} \) is typed (\( \text{pad}=0 \)), then wait an hour (3600 seconds), run the second experiment, etc.

Values: INOVA, 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: VnmrJ Liquids NMR; VnmrJ Walkup NMR

perform adept analysis and plot resulting spectra (C)

Syntax: \( \text{padept}(<\text{'noll'}>,<\text{'coef'}>,<\text{'theory'}>) \)

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:
- \( \text{'noll'} \) is a keyword that specifies no line listing.
- \( \text{'coef'} \) is a keyword that causes the combination coefficients to be printed.
- \( \text{'theory'} \) is a keyword that causes the theoretical coefficients rather than optimized coefficients to be used.

Examples: \( \text{padept}('\text{\text{noll}'}','\text{\text{coef}'}') \)

See also: VnmrJ Liquids NMR

Related:
- adept Automatic DEPT analysis and spectrum editing (C)
- autodept Automated complete analysis of DEPT data (M)
- cdept Automated carbon and DEPT acquisition (C)
- 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: `VnmrJ Liquids NMR`

Related: `vnmrplot` Plot files (U)

Page 

Name of page (P)

Description: Specifies the page of a sample. It is saved with a liquids study.

See also: `notebook (P)` `samplename (P)`

Panellevel 

Display level for VnmrJ interface pages (P)

Description: Determines which VnmrJ interface pages are available under the tabs. 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 shims or lock and minimal parameter control. This might be used for automation mode.

10-19, typical for a Walkup user without a sample changer. Shim and lock are available, but pages are not fully populated, minimizing parameter control.

20-29, typical for the Experimental liquids interface. All pages are available.

30-100, typical for the system owner. All pages are available and fully populated.

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: `VnmrJ Liquids NMR, 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: `VnmrJ Liquids NMR`

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: `VnmrJ Liquids NMR`

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 dc 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 \texttt{dg2} and modified \texttt{ap} display templates from the parameter set \texttt{s2pul3rf} in the system \texttt{parlib} directory. These two templates support the display of second decoupler acquisition parameters and 3D acquisition and processing parameters.

See also: \textit{User Programming}

Related: \texttt{ap} “All” parameters display control (P)
\texttt{dg2} Control \texttt{dg2} parameter group display (P)

par4d  Create 4D acquisition parameters (M)

Applicability: Systems with a third decoupler.

Description: Creates the acquisition parameters \texttt{ni3,sw3,d4}, and \texttt{phase3} that can be used to acquire a 4D data set. The \texttt{par4d} macro is functionally the same as \texttt{addpar('4d')}.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{addpar} Add selected parameters to the current experiment (M)
\texttt{d4} Incremented delay for 3rd indirectly detected dimension (P)
\texttt{ni3} Number of increments in 3rd indirectly detected dimension (P)
\texttt{phase3} Phase selection for 4D acquisition (P)
\texttt{sw3} Spectral width in 3rd indirectly detected dimension (P)

paramedit  Edit a parameter and its attributes with user-selected editor (C)

Syntax: \texttt{paramedit(\textit{parameter},\textit{tree})}

Description: Opens a parameter file for editing with a user-selected text editor. The default editor is \texttt{vi}. If \texttt{vi} is used as the editor, \texttt{paramedit} is functionally the same as the \texttt{paramvi} command. To select another editor, set the UNIX environmental variable \texttt{vnmreditor} to the editor name (change \texttt{.login} line \texttt{setenv vnmreditor old_editor} to become \texttt{setenv vnmreditor new_editor} (e.g., \texttt{vnmreditor emacs}) and make sure a script with the prefix \texttt{vnmr} followed by the name of the editor is placed in the \texttt{bin} subdirectory of the system directory (e.g., \texttt{vnmr_emacs}). The script file makes adjustments for the type of graphic interface in use.

Scripts in the software release include \texttt{vnmr_vi} and \texttt{vnmr_textedit}. To create other scripts, refer to the \texttt{vnmr_vi} script for non-window editor interfaces and to \texttt{vnmr_textedit} for window-based editor interfaces. The \texttt{vnmreditor} variable must be set before starting VnmrJ.

Arguments: \texttt{parameter} is the name of the parameter file to be edited.
\texttt{tree} is a keyword for one of the parameter trees 'current', 'global', or 'processed'. The default is 'current'.

Examples: \texttt{paramedit('ap')}
\texttt{paramedit('b','global')}

See also: \textit{VnmrJ Liquids NMR}; \textit{User Programming}

Related: \texttt{paramvi} Edit a parameter and its attributes with vi editor (M)
\texttt{vi} Edit text file with the vi text editor (C)

paramvi  Edit a parameter and its attributes with vi editor (M)

Syntax: \texttt{paramvi(\textit{parameter},\textit{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: VnmrJ Liquids NMR, 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)

pards Create additional parameters used by downsampling (M)

Description: Creates the parameters downsamp, dscoef, dsfb, dsrlsfreq, and filtfile necessary for digital filtering and downsampling. The pards macro is functionally the same as addpar('downsamp').

See also: VnmrJ Liquids NMR

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)
dslsfreq Bandpass filter offset for downsampling (P)
filtfile File of FIR digital filter coefficients (P)
movedssw Set downsampling parameters for selected spectral region (M)

parfids Create parameters for time-domain solvent subtraction (M)

Description: Creates solvent subtraction parameters ssfilter, ssrlsfreq, ssntaps, and ssorder. Entering addpar('ss') is functionally equivalent to parfids.

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 \texttt{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 \texttt{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 \texttt{ssorder}, (4) the polynomial function is subtracted from the raw FID, and (5) the resulting FID is frequency-shifted by \texttt{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 \texttt{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 \texttt{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 \texttt{lsfrq} or \texttt{sslsfrq} to move the solvent peak to within $\pm 0.2$ 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 \texttt{ft1d}, \texttt{ft2d}, \texttt{wft1d}, and \texttt{wft2d} commands and with the \texttt{ft2da}, \texttt{ft1da}, \texttt{wft2da}, and \texttt{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: \textit{VnmrJ Liquids NMR}

Related:
- \texttt{addpar} Add selected parameters to the current experiment (M)
- \texttt{ft} Fourier transform 1D data (C)
- \texttt{ft1d} Fourier transform along $f_2$ dimension (C)
- \texttt{ft2d} Fourier transform 2D data (C)
- \texttt{ft3d} Perform a 3D Fourier transform on a 3D FID data set (M,U)
- \texttt{lsfrq} Frequency shift of the fn spectrum in Hz (P)
- \texttt{ntype3d} N-type peak selection in $f_1$ or $f_2$ (P)
- \texttt{ssfilter} Full bandwidth of digital filter to yield a filtered FID (P)
- \texttt{sslsfrq} Center of solvent-suppressed region of spectrum (P)
- \texttt{ssorder} Order of polynomial to fit digitally filtered FID (P)
- \texttt{ssntaps} Number of coefficients to be used in the digital filter (P)
- \texttt{wft} Weight and Fourier transform 1D data (C)

\textbf{parfix} \hspace{1cm} \textbf{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 \texttt{dgs} is updated. This is automatically done via the macro \texttt{fixpar} if the parameter \texttt{parversion} is less than 4.3. \texttt{parfix} is used by the macro \texttt{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., \texttt{rtp('pars')} \texttt{svp('pars')} update a parameter set named \texttt{pars}).

See also: \textit{VnmrJ Liquids NMR}

Related:
- \texttt{ap} “All” parameters display control (P)
- \texttt{dgs} Control \texttt{dgs} parameter group display (P)
- \texttt{fixpar} Correct parameter characteristics in experiment (M)
- \texttt{parversion} Version of parameter set (P)
- \texttt{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`, `nscans`, `ntrig`, and `savefile`. The `parlc` macro also creates `n1` and `sw1` (if they don’t exist) for use in isocratic runs. Finally, it creates a display parameter `dg1c`, so that the `dg('dg1c')` command (or the equivalent macro `dg1c`) 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: VnmrJ Liquids NMR

Related:  
- `curscan` Scan currently in progress (P)  
- `dg1c` 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)  
- `nscans` Number of scout/real scan repetitions (P)  
- `ntrig` Number of trigger signals to wait before acquisition (P)  
- `savefile` Base file name for saving FIDs or data sets (P)

**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: VnmrJ Liquids NMR

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`, `strtrlp`, `lptest`, `strtext`, `lptrace`, and `lpprint`.

- `parlp(1)` creates LP parameters `lpalg1`, `lpopt1`, `lpfilt1`, `lpnupts1`, `strtrlp1`, `lptest1`, `strtext1`, `lptrace1`, and `lpprint1`. `addpar('lp',1)` is functionally equivalent to `parlp(1)`.

- `parlp(2)` creates LP parameters `lpalg2`, `lpopt2`, `lpfilt2`, `lpnupts2`, `strtrlp2`, `lptest2`, `strtext2`, `lptrace2`, and `lpprint2`. `addpar('lp',2)` is functionally equivalent to `parlp(2)`.

Examples:  
- `parlp`
- `parlp(1)`

See also: VnmrJ Liquids NMR

Related:  
- `lpalg` LP algorithm for np dimension (P)  
- `lptest` 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)
**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 CONFIG window (opened from config) 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: VnmrJ Liquids NMR

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)
ossfb Digital filter bandwidth for oversampling (P)
ossfilt Oversampling filter for real-time DSP (P)
oslsfrq Bandpass filter offset for oversampling (P)
oversamp Oversampling factor for acquisition (P)
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 CONFIG window (from config) 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)

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: VnmrJ Liquids NMR
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)
Applicability: All systems; however, although available on MERCURYplus/Vx such systems can only process 3D data and cannot acquire such data.
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: VnmrJ Liquids NMR
Related: dplane  Display a 3D plane (M)
dproj  Display a 3D plane projection (M)
dplanes  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)

patlist  Active pulse template parameter list (P)
Applicability: Systems with imaging capabilities.
Description: Contains an array of strings, whose values define the rf pattern parameters used in conjunction with the length parameters defined in plist, for example, patlist='p1pat','p2pat','p3pat'. The nD, seqcon, plist, patlist, pwrlist, fliplist and sslist parameters configure a particular parameter set for an application sequence defined by the value of the seqfil parameter. The plist, patlist, pwrlist, fliplist and sslist parameters provide information concerning the rf pulse and conjugate gradients used by the sequence.
See also: VnmrJ Imaging NMR
Related: fliplist Standard flip angle list (P)
        nD Application dimension (P)
        plist Active pulse length parameter list (P)
        pwrlist Active pulse power level parameter list (P)
        seqcon Acquisition loop control (P)
        seqfil Application object code name (P)
        sslist Conjugate gradient list (P)

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: VnmrJ Liquids NMR

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 VnmrJ Liquids NMR 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.
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: VnmrJ Liquids NMR

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)
cpx 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)
pboxqet 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)
pbh 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)
Pxsps 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 the 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: *VnmrJ Liquids NMR*

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 the 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: *VnmrJ Liquids NMR*

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`

See also: *VnmrJ Liquids NMR*

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`

See also: *VnmrJ Liquids NMR*

Related: Pbox Pulse shaping software (U)
See also: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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 level 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: *VnmrJ Liquids NMR*

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 level 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_pwrf('myfile.DEC'):ypwrf`
- `pbox_pwrf:dpwrf2`

See also: *VnmrJ Liquids NMR*

Related: **Pbox** Pulse shaping software (U)
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: VnmrJ Liquids NMR
Related: Pbox Pulse shaping software (U)

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,dfm pboxget('selshape.RF'):pwpat,selpw,selpwr pboxget:<dseq2,r1,dpwr2,dpwrf2,dres2,dfm2

See also: VnmrJ Liquids NMR
Related: Pbox Pulse shaping software (U)

pboxpar Add parameter definition to the Pbox.inp file (M)

Syntax: pboxpar(param,value)

Description: Adds a parameter definition to the Pbox.inp file.

Arguments: param is the parameter name
value is the value of the parameter.

Examples: pboxpar('name','myfile.DEC')
pboxpar('bsim','y')
pboxpar('T1', 0.24)

See also: VnmrJ Liquids NMR
Related: Pbox Pulse shaping software (U)
**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:* [VnmrJ Liquids NMR](#)

*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:* [VnmrJ Liquids NMR](#)

*Related:* Pbox  Pulse shaping software (U)

---

**pcmapapply**  **Apply phase correction map to data in EPI experiments (C)**

*Applicability:* Systems with echo planar imaging (EPI) capabilities.

*Syntax:* pcmapapply(<file,>index)

*Description:* Applies a pixel-by-pixel phase shift to the current data file using the complex phase correction values from the phase correction map file, which must exist in $vnmruser/expN/datdir, where \( N \) is the current experiment number. pcmapapply opens and closes a phase map file unless it has been explicitly opened with pcmapopen.

*Arguments:* file specifies a phase correction map file name that must reside in the directory $vnmruser/expN/datdir. The default file is $vnmruser/expN/datdir/pcmap.

index specifies which phase correction map to use in the file. The value is usually 1, but can range up to the number of map blocks in the file.

*Examples:* pcmapapply(2)
  pcmapapply('mypcmap',1)

*See also:* [VnmrJ Imaging NMR](#)

*Related:* pcmapclose  Apply phase correction map to data in EPI experiments (C)
  pcmapgen  Generate phase correction map in EPI experiments (C)
  pcmapopen  Open phase correction map file in EPI experiments (C)

---

**pcmapclose**  **Close phase correction map in EPI experiments (C)**

*Applicability:* Systems with echo planar imaging (EPI) capabilities.

*Description:* Closes a phase correction map file that was explicitly opened with the pcmapopen command.

*See also:* [VnmrJ Imaging NMR](#)

*Related:* pcmapapply  Apply phase correction map to data in EPI experiments (C)
  pcmapgen  Generate phase correction map in EPI experiments (C)
  pcmapopen  Open phase correction map file in EPI experiments (C)

---

**pcmapgen**  **Generate phase correction map in EPI experiments (C)**

*Applicability:* Systems with echo planar imaging (EPI) capabilities.


pcmapgen

Syntax: `pcmapgen(<file,>index)`

Description: Generates pixel-by-pixel complex phase correction values from the current data file and stores them into the selected block in the phase correction map file. One or more phase correction maps can be generated. For multislice echo planar imaging experiments, there can be one phase correction map for each slice.

`pcmapgen` creates, opens, and closes a phase map file unless the file has been explicitly opened with the `pcmapopen` command.

Arguments:
- `file` specifies a phase correction map file name, which must reside in the directory `$vnmruser/expN/datdir`, where `N` is the current experiment number. The default file is `$vnmruser/expN/datdir/pcmap`.
- `index` specifies which phase correction map to use in the file. The value is usually 1, but can range up to the number of map blocks in the file.

Examples:
- `pcmapgen(2)`
- `pcmapgen('mypcmap',1)`

See also: VnmrJ Imaging NMR

Related:
- `pcmapapply` Apply phase correction map to data in EPI experiments (C)
- `pcmapclose` Close phase correction map file in EPI experiments (C)
- `pcmapopen` Open phase correction map file in EPI experiments (C)

pcmapopen

Open phase correction map in EPI experiments (C)

Applicability: Systems with echo planar imaging (EPI) capabilities.

Syntax: `pcmapopen(<file,>max_index)`

Description: Explicitly opens a phase correction map file, which can significantly speed up data processing. After the map file is open, use `pcmapgen` and `pcmapapply` to generate maps and correct data. Use `pcmapclose` to close the file when you are finished with it.

Arguments:
- `file` specifies the phase correction map file name residing in the directory `$vnmruser/expN/datdir`, where `N` is the current experiment number. The default is the file `pcmap`.
- `max_index` specifies the maximum number of phase correction maps in the file, which ensures that memory mapping extends to or past the end of the file.

Examples:
- `pcmapopen(2)`
- `pcmapopen('mypcmap',1)`

See also: VnmrJ Imaging NMR

Related:
- `pcmapapply` Apply phase correction map to data in EPI experiments (C)
- `pcmapclose` Close phase correction map file in EPI experiments (C)
- `pcmapgen` Generate phase correction map in EPI experiments (C)

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 pen 1. 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: *VnmrJ Liquids NMR*

### 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: *VnmrJ Liquids NMR*

**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 `f1` or `f2` traces are counted.
- `$point` returns the data point number of the maximum on that trace.

See also: `VnmrJ Liquids NMR`

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)  

---

**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.

color is the active color for the graphics screen: 'red', 'green', 'blue', 'cyan', 'magenta', 'yellow', 'black', or 'white'.

Examples: pen('pen2')
pen('graphics','red')

See also: VnmrJ Liquids NMR

Related: draw Draw line from current location to another location (C)
move Move to an absolute location (C)

peexpl

Plot exponential or polynomial curves (C)

Syntax: peexpl(<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.

Examples: peexpl
peexpl(1,3,6)

See also: VnmrJ Liquids NMR, User Programming

Related: expl Display exponential or polynomial curves (C)
sc Start of chart (P)
sc2 Start of chart in second direction (P)
wcc Width of chart (P)
wcc2 Width of chart in second direction (P)

peexpladd

Add another diffusion analysis to current plot (M)

Applicability: Systems with the diffusion option.

Syntax: peexpladd(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: peexpladd(1)
See also: *VnmrJ Liquids NMR*

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:** *VnmrJ Liquids NMR; Pulsed Field Gradient Modules Installation; VnmrJ Liquids NMR*

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:** *VnmrJ Liquids NMR*

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**Related:**
- **av** Set abs. value mode in directly detected dimension (C)
- **avl** 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)
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: VnmrJ Liquids NMR
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 (P)pal 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 f2 of 2D data (M)wft2d Weight and Fourier transform 2D data (M)

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: VnmrJ Liquids NMR

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 f2 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:  VnmrJ Liquids NMR
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:  VnmrJ Liquids NMR
Related:   addpar     Add selected parameters to the current experiment (M)  
cosyps     Set up parameters for phase-sensitive COSY (M)  
dqcqcosy   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)  
mqcqosy    Set up parameters for MQCOSY pulse sequence (M)  
noesy      Set up parameters for NOESY pulse sequence (M)  
roeay      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: User Guide: Solid-State NMR
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: VnmrJ Liquids NMR; 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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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 t2 dimension in a 2D experiment or as the t3 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: VnmrJ Liquids NMR

Related:

- dfid: Display a single FID (C)
- ds: Display a spectrum FID (C)
- ft: Fourier transform 1D data (C)
- ft1d: Fourier transform along f2 dimension (C)
- ft2d: Fourier transform 2D data (C)
- lsfid: Number of complex points to left-shift the np FID (P)
- lsfqr: 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 f2 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 t1/t2 interferograms and in a hypercomplex phase rotation for hypercomplex t1/t2 interferograms.

phfid1 (and related parameters lsfid1 and lsfqr1) operate on ni interferogram data, both hypercomplex and complex. ni interferogram data are referred to as the t1 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 lsfqr1, if selected, to the time-domain data prior to the Fourier transformation.

Values: −360.0 to +360.0, in degrees; 'n'.

See also: VnmrJ Liquids NMR

Related:

- lsfid1: Number of complex points to left-shift the ni interferogram (P)
- lsfqr1: 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 t1/t2 interferograms and in a hypercomplex phase rotation for hypercomplex t1/t2 interferograms.

phfid2 (and related parameters lsfid2 and lsfqr2) operate on ni2 interferogram data, both hypercomplex and complex. ni2 interferogram data
are referred to as the \( t_1 \) dimension in a 3D experiment. \( \text{phfid2} \) is in the processing group and is properly handled through the \( \text{wti} \) display.

Values: \(-360.0\) to \(+360.0\), in degrees; 'n'.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \( \text{lsfid2} \) Number of complex points to left-shift \( ni2 \) interferogram (P)
- \( \text{lsfrq2} \) Frequency shift of the \( fn2 \) spectrum in Hz (P)
- \( ni2 \) Number of increments in 2nd indirectly detected dimension (P)
- \( \text{phfid} \) Zero-order phasing constant for \( np \) FID (P)
- \( \text{phfid1} \) Zero-order phasing constant for \( ni \) interferogram (P)
- \( \text{rp2} \) Zero-order phase in 2nd indirectly detected dimension (P)
- \( \text{wti} \) Interactive weighting (C)

\( \text{phi} \) \hspace{1cm} \textbf{Euler angle \( \phi \) from magnet frame (P)}

Applicability: Systems with imaging capabilities.

Description: Euler angle \( \phi \) from magnet frame.

Values: \(-180\) to \(+180\), in degrees.

See also: \textit{VnmrJ Imaging NMR}

Related:
- \( \text{psi} \) Euler angle \( \psi \) from magnet frame (P)
- \( \text{theta} \) Euler angle \( \theta \) from magnet frame (P)

\textbf{Phosphorus} \hspace{1cm} \textbf{Set up parameters for \( ^{31}\text{P} \) experiment (M)}

Description: Set up parameters for \( ^{31}\text{P} \) experiment.

\( \text{pi} \) \hspace{1cm} \textbf{Inversion pulse length (P)}

Applicability: Systems with imaging capabilities.

Description: Pulse length for an inversion pulse, often used as an optional first pulse preceding the main sequence to provide contrast based on \( T_1 \) relaxation.

A \( \text{pi} \) pulse will often be programmed so that it may be toggled on or off by the operator with the inversion-recovery flag \( \text{ir} \).

See also: \textit{VnmrJ Imaging NMR}

Related:
- \( \text{ir} \) Inversion recovery mode (P)
- \( \text{pipat} \) Shape of an inversion pulse (P)
- \( \text{ti} \) Second delay in an inversion recovery sequence (P)
- \( \text{tpwri} \) Intensity of an inversion pulse in dB (P)

\( \text{pi3ssbsq} \) \hspace{1cm} \textbf{Set up pi/3 shifted sinebell-squared window function (M)}

Syntax:

\[
\text{pi3ssbsq}(<\text{t1}\_\text{inc}>,<\text{t2}\_\text{inc}>)
\]

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:
- \( \text{t1}\_\text{inc} \) is the number of \( t_1 \) increments. The default is \( ni \).
- \( \text{t2}\_\text{inc} \) is the number of \( t_2 \) increments. The default is \( ni2 \).

See also: \textit{VnmrJ Liquids NMR}

Related:
- \( \text{gaussian} \) 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)
- \( \text{pi4ssbsq} \) Set up pi/4 shifted 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: VnmrJ Liquids NMR

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)

pilot  Automatic sequence setup (P)

Applicability: Systems with imaging capabilities.

Description: Provides a degree of automatic setup of a sequence, where this capability is available. If pilot='y', access is provided to automatic setting for the gradients gssr and gror. These gradient levels are then adjusted to compensate for gradient slew rate. The adjustments are made at the time of go; however, the values used are not returned to the parameter set.

Values:  
  'y' means the automatic mode is on.
  'n' means the manual mode is set.

See also: VnmrJ Imaging NMR

Related:  
  go  Submit experiment to acquisition (C)
  gror  Readout compensation gradient (P)
  gssr  Slice selection refocusing gradient (P)

pintvast  Plots of integral regions (M)

Applicability: Systems with VAST accessory.

Syntax:  pintvast (last)

Description: pintvast plots 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: VnmrJ Liquids NMR

Related:  
  intvast  Builds text file the integral regions (M)

pipat  Shape of an inversion pulse (P)

Applicability: Systems with imaging capabilities.

Description: Specifies the shape of inversion pulse pi.

Values:  
  'hard', 'sinc', 'gauss', 'sech', 'sine', or any shape resident in the system pulse shape library or libraries.
pir

Plot integral amplitudes below spectrum (C)

Description: Plots integral amplitudes below the appropriate spectral regions.

See also: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related: dpirn  Display normalized integral amplitudes below spectrum (M)
ins  Integral normalization scale (P)
pir  Plot integral amplitudes below spectrum (C)

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 f1 or f2 domain by setting the parameter trace to 'f1' or 'f2', respectively. After the command ftld, interferograms 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: VnmrJ Liquids NMR

Related:
cutoff  Data truncation limit (P)
dssa  Display stacked spectra automatically (C)
dsww  Display spectra in whitewash mode (C)
ft1d  Fourier transform along f2 dimension (C)
ho  Horizontal offset (P)
intmod  Integral display mode (P)
plww  Plot spectra in whitewash mode (C)
sc  Start of chart (P)
sc2  Start of chart in second direction (P)
trace  Mode for 2D data display (P)
vo  Vertical offset (P)
vp  Vertical position of spectrum (P)
wcc  Width of chart (P)
wcc2  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: VnmrJ Liquids NMR

Related: dcon  Display noninteractive color intensity map (C)
dsa2d  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)
**plan**

**Display menu for planning a target scan (M)**

**Applicability:** Systems with imaging capabilities.

**Description:** Brings up a menu that provides access to the target scan planning utilities. The plan menu has three buttons: Slice, Voxel, and Exit.

The Slice button provides access to the slice planning menu. The user first clears the current experiment of any mark2d.out files using the Clear Marks button. The image display may then be made interactive using the Interactive View button. This activates the dconi program. The user should select and mark two points that lie on the edge of the desired target slice plane using the Mark button of the dconi menu. To write the mark data into the mark2d.out file, the user should exit dconi using the Return button. This exits to the slice planner menu.

The target slice selection can be shown graphically on the image display using the Show Target button of the slice planner menu. This button uses the drawslice macro. The slice parameters (pss, psi, phi, and theta) are calculated and set using the Calculate Target button of the slice planner menu. This button uses the ssplan macro. This program creates the string parameter planlock and assigns it the value 'ssplan'. This prevents a user inadvertently performing a second planning operation without applying the reset command to restore the original parameters for the scout data.

At this point, the current parameters of the scout experiment contain the data needed to acquire the desired slice. The user can use these directly or use the mp or transfer commands to move the information to another experiment.

The Voxel button of the plan menu provides access to the voxel planning menu. The user may enter the interactive mode using the Interactive View button. This activates the dconi program. The user should clear any previous unwanted planning information before starting.

The size and position of the voxel face parallel to the image plane can be selected by positioning the 2D box cursor. Once this is done, the user leaves the interactive mode using the Return button of the dconi menu. This returns the user to the voxel planning menu. The user can plan for more than one voxel. These target voxel selections can be shown graphically on the image display using the Show Target button of the planner menu. This button uses the drawvox macro. The parameter for the voxel can be calculated and set using the Calculate Target button, which uses the voxplan macro.

The voxplan macro requests the user to enter the voxel size in the direction parallel to the scout image slice select axis. Voxel parameters are computed from the 2D box cursor data and user entry. The voxel center is taken to lie in the scout image plane at the center of the 2D box. voxplan also creates the string parameter planlock and assigns it the value 'voxplan'. This provides an interlock against further planning operations. The reset command restores the original scout parameters and removes the planlock parameter.

The current parameters of the scout experiment contain the data needed to acquire the voxel. The user must use the transfer program to copy this data to the parameter set of a suitable voxel selective sequence.

**See also:** VnmrJ Imaging NMR

**Related:**
- drawslice: Display target slices (M)
- drawvox: Display target voxels (M)
- mp: Move parameters between experiments (C)
- phi: Euler angle phi from magnet frame (P)
- planlock: Planner lockout (P)
- psi: Euler angle psi from magnet frame (P)
- pss: Slice position (P)
- ssplan: Set slice parameters for target slice (M)
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.

Values: 'f1f3', 'f3f1', 'f2f3', 'f3f2', 'f1f2', or 'f2f1'

See also: VnmrJ Liquids NMR

planlock

Planner lock (P)

Description: Created by voxplan and assigned the value 'voxplan' to provide an interlock against further planning operations. This parameter is also created by the ssplan macro and assigned the value 'ssplan' to prevents a user inadvertently performing a second planning operation. In both cases, the reset command removes the value assigned to planlock.

See also: VnmrJ Imaging NMR

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: \texttt{13Cexp\_number} specifies the number, from 1 to 9, of an experiment with a standard $^{13}$C spectrum.

Examples:

\begin{verbatim}
plapt
plapt(2)
\end{verbatim}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{curexp} \hspace{1cm} Current experiment directory (P)

\textbf{plarray} \hspace{1cm} \textbf{Plotting macro for arrayed 1D spectra (M)}

Description: A generic macro for plotting arrayed 1D spectra. \texttt{plarray} is called by the \texttt{plot} macro, but can also be used directly. For the plot layout, \texttt{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 \texttt{stackmode} in the startup macro or before calling \texttt{procplot} or \texttt{procarray}. Possible values for \texttt{stackmode} are 'horizontal', 'vertical', or 'diagonal'. DEPT-type spectra can, in principle, also be processed with \texttt{procarray}, but no DEPT editing occurs, of course.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{aexppl} \hspace{1cm} Automatic expansion plot (M)
\texttt{plc} \hspace{1cm} Plot carbon spectrum (M)
\texttt{plh} \hspace{1cm} Plot proton spectrum (M)
\texttt{plot} \hspace{1cm} Automatically plot spectra (M)
\texttt{procarray} \hspace{1cm} Process arrayed 1D spectra (M)
\texttt{stackmode} \hspace{1cm} Stack control for processing arrayed 1D spectra (P)

\textbf{plate\_glue} \hspace{1cm} \textbf{Define a glue order for plotting and display (U)}

Applicability: Systems with VAST accessory

Description: In a Unix terminal or shell window type \texttt{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: \textit{VnmrJ Liquids NMR}

Related: \texttt{dsvast2d} \hspace{1cm} Display VAST data in a pseudo-2D format (M)
\texttt{plvast} \hspace{1cm} Plot VAST data in a stacked 1D-NMR matrix (M)
\texttt{plvast2d} \hspace{1cm} Plot VAST data in a pseudo-2D format (M)

\textbf{plc} \hspace{1cm} \textbf{Plot a carbon spectrum (M)}

Syntax: \texttt{plc<(pltmod)>}

Description: Plots a carbon spectrum based on the parameters \texttt{pltmod} (the options 'off', 'full', and 'fixed' are implemented) and \texttt{intmod} ('off', 'full', and 'partial' are implemented). Peak frequency labels, in ppm, are usually plotted.

Arguments: \texttt{pltmod} is an alternate value of \texttt{pltmod} for this macro only. The value of the \texttt{pltmod} parameter is not changed.
Examples:

\begin{verbatim}
plc
plc('full')
\end{verbatim}

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:}

\begin{itemize}
  \item \texttt{intmod} \quad Integral display mode (P)
  \item \texttt{pltmod} \quad Plotter display mode (P)
\end{itemize}

**plcosy**

\textbf{Plot COSY- and NOESY-type spectra automatically (M)}

\textbf{Syntax:} \texttt{plcosy(<'pos'|'neg'>,<,><levels<,spacing<,exp1D>>>)}

\textbf{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 \texttt{'pos'} or \texttt{'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 \texttt{exp1}). From then on, the 1D spectrum will be stored \textit{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 \texttt{exp1D} argument is not required for subsequent plots.

\textbf{Arguments:}

- \texttt{'pos'} is a keyword to plot only positive contours.
- \texttt{'neg'} is a keyword to plot only negative contours.
- \texttt{levels} is the number of contour levels. The default is 7.
- \texttt{spacing} is the spacing between the contours. The default is 2.
- \texttt{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.

\textbf{Examples:}

\begin{verbatim}
plcosy
plcosy(12,1.5)
plcosy('pos',7,2,3)
plcosy(7,2,-1)
plcosy('neg')
\end{verbatim}

See also: \textit{VnmrJ Liquids NMR}

**pldept**

\textbf{Plot DEPT data, edited or unedited (M)}

\textbf{Description:} Plots out DEPT data, either edited or not edited.
See also: *VnmrJ Liquids NMR*

**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)

### 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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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 f2 and f1 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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related: intmod Integral display mode (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 f2 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 multieperiment 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: VnmrJ Liquids NMR

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 f2 axis; such a 1D trace is taken from a full (or reduced) 1D spectrum in another 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 multieperiment 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: *VnmrJ Liquids NMR*

### 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')`
plist

Active pulse length parameter list (P)

Applicability: Systems with imaging capabilities.

Description: Contains an array of strings, whose values are the names of the rf pulse length parameters used by the sequence (e.g., \texttt{plist='p1','p2','p3'}). The \texttt{nD}, \texttt{seqcon}, \texttt{plist}, \texttt{patlist}, \texttt{pwrlist}, \texttt{fliplist}, and \texttt{sslist} parameters configure a particular parameter set for an application sequence defined by the value of the \texttt{seqfil} parameter. The \texttt{plist}, \texttt{patlist}, \texttt{pwrlist}, \texttt{fliplist}, and \texttt{sslist} parameters provide information concerning the rf pulse and conjugate gradients used by the sequence.

See also: \textit{VnmrJ Imaging NMR}

Related: \texttt{fliplist} Standard flip angle list (P)
\texttt{gcoil} Read data from gradient calibration tables (P)
\texttt{nD} Application dimension (P)
\texttt{patlist} Active pulse template parameter list (P)
\texttt{pwrlist} Active pulse power level parameter list (P)
\texttt{rfcoil} RF pulse calibration identity (P)
\texttt{seqcon} Acquisition loop control (P)
\texttt{seqfil} Application object code name (P)
\texttt{sslist} Conjugate gradient list (P)

pll

Plot a line list (M)

Syntax: \texttt{pll\langle x,y,minimum\_y\rangle}"

Description: Produces a columnar line list on a plotter, similar to what would appear on a printer. \texttt{pll} is quite different from the alternative method of plotting peak frequencies using \texttt{ppf}. The output of \texttt{pll} is automatically formatted into multiple columns, depending on the number of lines.

Arguments:
\texttt{x} is the \texttt{x} position of the upper left of the line list.
\texttt{y} is the \texttt{y} position of the upper left of the line list.
\texttt{minimum\_y} is the minimum \texttt{y} at which to reset back to top.

Examples:
\texttt{pll}
\texttt{pll(20,150)}
\texttt{pll(5,wc2max*.8,wc2max*.5)}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{ppf} Plot peak frequencies over spectrum (M)

pll2d

Plot results of 2D peak picking (C)

Syntax: \texttt{pll2d\langle options\rangle}"

Description: Plots the results of applying the \texttt{ll2d} command to pick 2D peaks in a 2D spectrum or a 2D plane of a 3D spectrum. Refer to the description of \texttt{ll2d} for a description of the process and the options available.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{ll2d} Automatic and interactive 2D peak picking (C)
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: *VnmrJ Liquids NMR*

Related:
- `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)
- `procplot` Automatically process FIDs (M)

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: *VnmrJ Liquids NMR*

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:**
- 'pos' is a keyword to plot positive contours.
- 'neg' is a keyword to plot negative contours.
- 'both' is a keyword to plot both positive and negative contours.
- levels is the number of levels to be plotted.
- spacing is the spacing between contour levels.
- 'top' is a keyword to plot a high-resolution spectrum on the top.
- 'notop' is a keyword to plot a non-high-resolution spectrum or projection.
- 'proj' is a keyword to plot a projection on top.
- 'side' is a keyword to plot a high-resolution spectrum on the side.
- 'noside' is a keyword to plot a non-high-resolution spectrum or projection.
- 'proj' is a keyword that plots a projection on the side.

**Examples:**
```plaintext
plot2D('pos',2,5,'top','side')
```

**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)

**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.

**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:** VnmrJ Liquids NMR

**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.

**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.

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: *VnmrJ Liquids NMR*

Related: `intmod`  Integral display mode (P)
`plh`  Plot proton spectrum (M)
`pltmod`  Plotter display mode (P)

plplanes  
**Plot a series of 3D planes (M)**

Applicability: All systems; however, although `plplanes` is available on MERCURYplus/Vx systems, such systems can only process 3D data and cannot acquire 3D data.

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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**Plttext**  
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: *VnmrJ Liquids NMR*

**Related:**
- `dplane` Display a 3D plane (M)
- `dproj` Display a 3D plane projection (M)
- `dplanes` 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)

**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)
plvast

Plot VAST data in a stacked 1D-NMR matrix format (M)

Applicability: Systems with the VAST accessory.

Syntax: plvast<(display order, number of columns plotted)>

Description: plvast arranges and plots the traces from a reconstructed 2D data set (see vastglue) as an array of 1D spectra in a convenient format (as a matrix of 1D spectra). If no arguments are provided, the number of rows and columns are determined by the periodicity of the display order. For example, if a block of 96 spectra, as is typical for a microtiter-plate, have been acquired using VAST automation, the spectra is plotted in a matrix 8 rows and 12 columns.

The default is to plot the spectra from 1 through arraydim (the number of spectra in the 2D data set). An optional argument (plvast(##)) allows one to specify that only spectra from 1 through ## should be plotted.

Arguments: display order is optional and its default value is the glue order as listed in glueorderarray.

number of columns plotted. The default value of is deduced by examining the periodicity of the requested display order. The number of columns plotted can entered as the second argument or as the first argument if the default display order is used.

Examples: plvast
plvast(12)
plvast('glue_file', 4)

See also: VnmrJ Liquids NMR

Related: dsaast2d Display VAST data in a pseudo-2D format (M)
dsvast Display VAST data in a stacked 1D-NMR matrix (M)
plvast2d Plot VAST data in a stacked pseudo-2D format (M)
plate_glue define a display order (U)

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).

See also: VnmrJ Liquids NMR

Related: dsaast2d 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 is the 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 is the index of the last spectra when plotting multiple spectra.

step is the increment for the spectral index when plotting multiple spectra. The default is 1.

'all' is a keyword to plot all spectra in the array. This is the default.

See also: VnmrJ Liquids NMR

Related: das 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 f2 or f1 display mode after the 2D FT. If the f2 display mode has been set to phased (dmg='ph'), each f2 spectrum is phase rotated using the phase constants rp and lp prior to the FT along the second dimension. If the f2 display mode has been set to power (dmg='pwr') or absolute-value (dmg='av'), however, the f2 spectrum is not processed any further after the first FT. The complex t1 interferograms are handled in a similar manner. If the f1 display mode has been set to phased (dmg1='ph1'), each f1 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'), however, the f2 spectrum is not processed any further after the first FT. Regardless of the requested f1 display mode, no further processing is performed by ft2d on the f2 spectra after the second FT. The calculations on 2D spectral data necessary to achieve the requested f1 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 change the f2 display mode after the 2D FT. It is possible, however, to select between the three f2 display modes without having to reprocess the 2D data. If the f2 display mode has been set to phased (dmg='ph'), each f2 spectrum is phase rotated using the phase constants rp and lp prior to FT along the second dimension. If the f2 display mode is set to power (dmg='pwr') or absolute value (dmg='av'), the f2 spectrum is not processed any further after the first FT. Regardless of the requested f1 display mode, no further processing is performed by ft2d on the f1 spectra after the second FT. The calculations on 2D spectral data necessary to achieve the requested f1 display mode are performed by dcon or dconi. If pmode does not exist, it is assigned a value of 'full' internal to VnmrJ.

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 t1 interferograms is:
\{\text{Re}(t_1)\text{Re}(F_2), \text{Re}(t_1)\text{Im}(F_2), \text{Im}(t_1)\text{Re}(F_2),  \\
\text{Im}(t_1)\text{Im}(F_2)\}\}

where \text{Re} represents the real part and \text{Im} represents the imaginary part. A hypercomplex FT along \(t_1\) yields a hypercomplex 2D spectrum with the following data structure per hypercomplex point:

\{\text{Re}(F_1)\text{Re}(F_2), \text{Re}(F_1)\text{Im}(F_2), \text{Im}(F_1)\text{Re}(F_2),  \\
\text{Im}(F_1)\text{Im}(F_2)\}\}

Note that if \text{ftmode} = 'full', the \text{ft2d} program will require an array index or coefficients for the construction of the \(t_1\) interferograms.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \textit{av}: Set abs. value mode in directly detected dimension (C)
- \textit{av1}: Set abs. value mode in 1st indirectly detected dimension (C)
- \textit{dcon}: Display noninteractive color intensity map (C)
- \textit{dconi}: Interactive 2D data display (C)
- \textit{dmg}: Data display mode in directly detected dimension (P)
- \textit{dmg1}: Data display mode in 1st indirectly detected dimension (P)
- \textit{ft1d}: Fourier transform along \(f_2\) dimension (C)
- \textit{ft2d}: Fourier transform 2D data (C)
- \textit{ph}: Set phased mode in directly detected dimension (C)
- \textit{ph1}: Set phased mode in indirectly detected dimension (C)
- \textit{pwr}: Set power mode in directly detected dimension (C)
- \textit{pwr1}: Set power mode in 1st indirectly detected dimension (C)
- \textit{wft1d}: Weight and Fourier transform 2D data (C)
- \textit{wft2d}: Weight and Fourier transform 2D data (C)

\textbf{poly0}  
\textbf{Display mean of the data in regression.inp file (M)}

\textbf{Description:} Calculates and displays the mean of data in the file \textit{regression.inp}.

See also: \textit{User Programming}

Related:
- \textit{averag}: Calculate average and standard deviation of input (C)
- \textit{expl}: Display exponential or polynomial curves (C)

\textbf{pos1 - pos3}  
\textbf{Position of voxel center (P)}

\textbf{Applicability:} Systems with imaging capabilities.

\textbf{Description:} Define the center position, in cm, of the desired voxel for localized spectroscopy experiments.

See also: \textit{VnmrJ Imaging NMR}

Related:
- \textit{transfer}: Move parameters to target experiment (M)
- \textit{vox1,vox2,vox3}: Voxel dimensions (P)

\textbf{pp}  
\textbf{Decoupler pulse length (P)}

\textbf{Description:} Sets the decoupler pulse length for use by pulse sequences such as DEPT, HET2DJ, and HETCOR.

See also: \textit{VnmrJ Liquids NMR}

Related:
- \textit{AC1-AC9}: Automatic calibration (M)
- \textit{dept}: Set up parameters for DEPT pulse sequence (M)
- \textit{dhp}: Decoupler high-power control with class C amplifier (P)
- \textit{dpwr}: Power level for first decoupler with linear amplifier (P)
- \textit{hetcor}: Set up parameters for HETCOR pulse sequence (M)
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:

```plaintext
ppa
ppa(10)
ppa(wcmax-80,wc2max*.9)
```

See also: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related:

- `AC1-AC9`: Automatic calibration (M)
- `d2pul`: Set up parameters for D2PUL pulse sequence (M)
- `dept`: Set up parameters for DEPT pulse sequence (M)
- `hetcor`: Set up parameters for HETCOR pulse sequence (M)
- `inept`: Set up parameters for INEPT pulse sequence (M)

ppe

Position of image center on 2D phase encode axis (P)

Applicability: Systems with imaging capabilities.

Description: Position of image center on 2D phase encode axis, in cm.

See also: *VnmrJ Imaging NMR*

Related:

- `pro`: Position of image center on the readout axis (P)

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: Pbox Pulse shaping software (U)

**pplvl**

*Proton pulse power level (P)*

Applicability: *MERCUReplus/VX*, broadband systems with the diode switching version of RF Control board and systems with amptype='a'.

Description: Sets the pulse power level. pplvl is only a relevant parameter in sequences that use decoupler pulses, such as DEPT, HET2DJ, and HETCOR.

Values: 0 to 63, in dB, steps of 1 dB.

When used with a 5-mm Gen. III switchable probe, typical value is 54 or 56.

See also: *VnmrJ Liquids NMR*

Related: amptype Amplifier type (P)
d2pul Set up parameters for D2PUL pulse sequence (M)
dep Set up parameters for DEPT pulse sequence (M)
het2dj Set up parameters for HET2DJ pulse sequence (M)
hetcor Set up parameters for HETCOR pulse sequence (M)
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
Related: dps               Display pulse sequence (C)
seqfil            Pulse sequence name (P)
wcmax             Maximum width of chart (P)
w2cmax            Maximum width of chart in second direction (P)
prep  prepare a scan (M)
Applicability: Imaging systems.
Description: Macro to prepare a scan. It uses execpars to select the prep method.
See also: apptype(P) execpars(M) execprep(P)

presat  Set up parameters for PRESAT pulse sequence (M)
Description: Sets up a 1D water suppression experiment.
See also: VnmrJ Liquids NMR

Presat  Set up parameters for presat $^1$H experiment (M)
Description: Set up parameters for presat $^1$H experiment.

presig  Preamplifier signal level selection (P)
Applicability: Systems with imaging capabilities.
Description: Allows the user to select either high or low signal handling on preamplifiers that support this capability:

- **UNITY** or **INOVA** imaging systems support this capability by using attenuation and a current increase. This allows larger signals and results in a lower overall signal level.
- **UNITY** or **INOVA** spectrometers with selectable large-signal mode preamplifiers support this capability by allowing a current increase in the preamplifier. This allows larger signals so that the overall signal level is slightly higher.

Using presig to control the hardware depends on the Magnet Leg Driver Board Configuration ID being set to 16 for imaging systems, or to 1 for **UNITY** or **INOVA** spectrometers with the selectable large-signal mode preamplifier.

Values: 'h' signifies high-signal mode at the preamplifier.
'1' signifies low-signal mode at the preamplifier. The default is this mode at the preamplifier if the hardware is present
'n' signifies not used.
See also: VnmrJ Imaging NMR
Related: gain Receiver gain (P)

dprevpl  Display the previous 3D plane (M)
Applicability: All systems; however, although dprevpl is available on MERCURYplus/Vx, such systems can only process 3D data and cannot acquire 3D data.
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, dprevpl results in the display of 3D plane 39 of that set. (If dprevpl immediately follows the command dproj, an error results because there is no 3D plane whose number is –1.) dprevpl 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: VnmrJ Liquids NMR
Related: dplane Display a 3D plane (M)
dproj Display a 3D plane projection (M)
dspplanes Display a series of 3D planes (M)
**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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
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: *VnmrJ Liquids NMR*

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'

**pro**  
**Position of image center on the readout axis (P)**

Applicability: Systems with imaging capabilities.

Description: Position of image center on readout axis, in cm.

See also: *VnmrJ Imaging NMR*

Related: `ppe` Position of image center on 2D phase encode axis (P)

**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: *VnmrJ Liquids NMR*

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)

**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:  VnmrJ Liquids NMR

proc  Type of processing on np FID (P)
Description:  Specifies the type of data processing to be performed upon the np (t2) FID. Similarly, parameters proc1 and proc2 specify the type of data processing on the ni (t1) 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:  VnmrJ Liquids NMR
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)
**proc1**

**Type of processing on ni interferogram (P)**

**Description:** Specifies the type of data processing to be performed upon the ni (t1) 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.

**See also:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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 T1 and T2 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: VnmrJ Liquids NMR

Related:
- deptproc: Process arrayed dept type spectra (M)
- plarray: Plot arrayed 1D spectra (M)
**process**  
**Generic automatic processing (M)**

**Description:** Processes a wide range of data types. 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:** *VnmrJ Liquids NMR*

**Related:**
- `c13p`  
  Processing of 1D carbon spectra (M)
- `deptproc`  
  Process array of DEPT spectra (M)
- `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:** *VnmrJ Liquids NMR*

**Related:***
- `deptproc`  
  Process arrayed dept type spectra (M)
- `plot`  
  Automatically plot 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. The gradaxis parameter is used by profile to select
the x, y, or z gradient axis.
See also: Performa I Pulsed Field Gradient Module Installation; Pulsed Field Gradient
Modules Installation; User Programming
Related: gcal  Gradient calibration constant (P)
gradaxis  Gradient axis (P)
setgcal  Calibrate gradient strength from measured data (M)

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 f1 or
f2, 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: VnmrJ Liquids NMR
Related: trace  Select mode for 2D data display (P)

Proton  Set up parameters for ¹H experiment (M)
Description: Set up parameters for ¹H experiment.
**prune**

**Prune extra parameters from current tree (C)**

**Syntax:**
```plaintext
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:**
```plaintext
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. `wp0` 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:** `VnmrJ Liquids NMR`

**Related:**
- `axis` Axis label for displays and plots (P)
- `axisf` Axis label for FID displays and plots (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 \( \text{l}b=\frac{-0.318}{(C1*\text{at})} \). The default value of C1 is 0.0625.
- C2 sets \( \text{g}f=C2*\text{at} \). The default value of C2 is 0.25.
- C3 sets \( \text{l}b1=\frac{-0.318}{(C3*(\text{n}i/\text{sw1})} \) but is used with 2D experiments only. The default value of C3 is 0.0625.
- C4 sets \( \text{g}f1=C4*(\text{n}i/\text{sw1}) \) but is used with 2D experiments only. The default value of C4 is 0.25.

Examples:
- pseudo
- pseudo(.1,.4,.2,.5)

See also: VnmrJ Liquids NMR

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,...,parN 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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related:
- `dshape` Display the last created pulse shape (M)  
- `Pbox` Pulse shaping software (U)

**psi** Euler angle psi from magnet frame (P)

Applicability: Systems with imaging capabilities.

Description: Euler angle psi from magnet frame.

Values: –90 to +90, in degrees

See also: *VnmrJ Imaging NMR*

Related:
- `phi` Euler angle phi from magnet frame (P)  
- `theta` Euler angle theta from magnet frame (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:*  
*VnmrJ Liquids NMR*  
*Related:* `seqfil`  
Pulse sequence name (P)

**pss**  
**Slice position (P)**

*Applicability:* Systems with imaging capabilities.

*Description:* Position of slice, in cm.

*See also:*  
*VnmrJ Imaging NMR*  
*Related:* `plan`  
Display menu for planning a target scan (M)

**pss0**  
**Stack center shift along z axis (P)**

*Applicability:* Systems with imaging capabilities.

*Description:* Shift of stack center along z, the axis perpendicular to the plane. Also used for `pos3`.

*See also:*  
*VnmrJ Imaging NMR*  
*Related:*  
`pos1-pos3`  
Position of voxel center (P)

**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:*  
*VnmrJ Liquids NMR*  
*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)**

*Applicability:* All systems; however, although `ptspec3d` is available on MERCURYplus/Vx, such systems can only process 3D data and cannot acquire 3D data.

*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 f₃ dimension. If `ptspec3d='ynn'`, only the currently displayed region of f₃ is retained as non-zero values after the f₃ transform in the 3D FT. A larger f₃ region may be kept to ensure that the number of hypercomplex f₃ points is a power of 2; but that portion of the f₃ 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 f₁ dimension (sw, np, fn); the second character, to the f₂ dimension (sw₁, ni, fn₁); and the third character, to the f₃ 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: VnmrJ Liquids NMR

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 f₁ or f₂ (P)
          par3d Create 3D acquisition, processing, display parameters (C)
          specdc3d 3D spectral dc 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 CONFIG window (opened from config).

**Values:** 0 (Not Present choice in CONFIG window); 160, 200, 250, 320, 500, 620, 1000 (PTS 160, PTS 200, PTS 250, PTS 320, PTS 500, PTS 620, PTS 1000 choices in CONFIG window, respectively). On MERCURYplus/Vx, ptsval has no meaning.

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)

pulsecal

**Update and display pulse calibration data file (M)**

**Applicability:** Systems with the imaging capabilities.

**Syntax:**
(1) pulsecal<name,pattern,length,flip,power>
(2) pulsecal(name,'remove')

**Description:** Creates and maintains a database file of rf coil calibration data. This database is accessed by the SEQUD command **setflip** in order to automatically enter power level settings for various types of rf pulses.

If entered without arguments, pulsecal displays the current contents of the database file. Using pulsecal with syntax 1 creates an entry in the file userdir+'/'+pulsecal'. Using syntax 2 removes the entire line associated with the calibration name.
Arguments:  
name is the name of the rf coil or calibration.

pattern is the rf pattern used in the calibration experiment.

length is the length of the rf pulse, in µs, used for calibration.

flip is the flip angle calibrated, in degrees.

power is the calibrated power level, in attenuator units.

'remove' is a keyword to remove the line associated with the calibration name.

Examples:  
pulsecal
pulsecal('small_coil','sinc',5000,180,88)
pulsecal('small_coil','remove')

See also:  
VnmrJ Imaging NMR

Related:  
setflip Set rf power levels for desired flip angle (M)
userdir User directory (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:  
VnmrJ Liquids NMR
**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: *VnmrJ Liquids NMR*

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 pulselength, 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: *VnmrJ Liquids NMR*

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 time, 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: *VnmrJ Liquids NMR*

**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:**
- On `MERCURYplus/Vx`: 0, 0.2 µs to 150,000 sec.
- On `INOA`: 0, 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also: *VnmrJ Liquids NMR*

**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:**
- On `MERCURYplus/Vx`: 0, 0.2 µs to 150,000 sec.
- On `INOA`: 0, 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

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:
- `pipat` 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: *VnmrJ Liquids NMR*

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)
- `f0d` Fourier transform along f0 dimension (C)
- `ft2d` Fourier transform 2D data (C)
- `pa` Set phase angle mode in directly detected dimension (C)
- `pal` 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 f1 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
The \texttt{pwr1} command is only needed if mixed-mode display is desired. If the parameter \texttt{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 \texttt{dmg}). For the contour display of multidimensional data, the result of \texttt{pwr1} is the same as for traces, provided that \texttt{pmode=’partial’} or \texttt{pmode=’’}.

See also: \textit{VnmrJ Liquids NMR}

\begin{verbatim}
Related: 
\texttt{dmg1}  Data display mode in 1st indirectly detected dimension (P) 
\texttt{pa}  Set phase angle mode in directly detected dimension (C) 
\texttt{pa1}  Set phase angle mode in 1st indirectly detected dimension (C) 
\texttt{pmode}  Processing mode for 2D data (P) 
\texttt{pwr}  Set power mode in directly detected dimension (C) 
\texttt{pwr2}  Set power mode in 2nd indirectly detected dimension (C) 
\end{verbatim}

\textbf{pwr2} \hspace{1cm} \textit{Set power mode in 2nd indirectly detected dimension (C)}

\textbf{Description:} Selects the power spectra display mode along the second indirectly detected dimension by setting \texttt{dmg2=’pwr2’}. If \texttt{dmg2} does not exist or is set to the null string, \texttt{pwr2} will create \texttt{dmg2} and set it equal to \texttt{’pwr2’}. In the \textit{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 \texttt{pwr2} command is only needed if mixed-mode display is desired. If the parameter \texttt{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 \texttt{dmg}). For the contour display of multidimensional data, the result of \texttt{pwr2} is the same as for traces, provided that \texttt{pmode=’partial’} or \texttt{pmode=’’}.

See also: \textit{VnmrJ Liquids NMR}

\begin{verbatim}
Related: 
\texttt{av2}  Set abs. value mode in 2nd indirectly detected dimension (C) 
\texttt{dmg2}  Data display mode in 2nd indirectly detected dimension (P) 
\texttt{ft1d}  Fourier transform along f2 dimension (C) 
\texttt{ft2d}  Fourier transform 2D data (C) 
\texttt{ph2}  Set phased mode in 2nd indirectly detected dimension (C) 
\texttt{pmode}  Processing mode for 2D data (P) 
\texttt{pwr}  Set power mode in directly detected dimension (C) 
\end{verbatim}

\textbf{pwrlist} \hspace{1cm} \textit{Active pulse power level parameter list (P)}

\textbf{Applicability:} Systems with imaging capabilities.

\textbf{Description:} Contains an array of strings that define the names of the power level parameters associated with \texttt{plist} and \texttt{patlist}. The \texttt{nd}, \texttt{seqcon}, \texttt{plist}, \texttt{patlist}, \texttt{pwrlist}, \texttt{fliplist} and \texttt{sslist} parameters configure a particular parameter set for an application sequence defined by the value of the \texttt{seqfil} parameter. The \texttt{plist}, \texttt{patlist}, \texttt{pwrlist}, \texttt{fliplist} and \texttt{sslist} parameters provide information concerning the rf pulse and conjugate gradients used by the sequence.

\textbf{Values:} String array such as \texttt{pwrlist=’tpwr1’,’tpwr2’,’tpwr3’}. 
See also: *VnmrJ Imaging NMR*

**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 (13C, 15N, or 31P) 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 13C-1H coupling constant, `jN15` is the 15N-1H coupling constant, `jP31` is the 31P-1H coupling constant, and `jname` is a selected calibration nucleus.

**See also:** *System Administration*

**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: *VnmrJ Liquids NMR*

Related: Pbox Pulse shaping software (U)
pboxget Extract Pbox calibration data (M)

**pxshape**

*Generates a single-band shape file (M)*

Syntax:

```plaintext
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:

```plaintext
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: *VnmrJ Liquids NMR*

Related: Pbox Pulse shaping software (U)

**Pxsim**

*Simulate Bloch profile for a shaped pulse (U)*

Syntax:

```plaintext
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
Related: Pbox Pulse shaping software (U)
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 VnmrJ Liquids NMR 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.
           On MERCURY, use qtune(0,20)
Examples:  qtune
           qtune(20)
           qtune(38,65)
See also:  VnmrJ Liquids NMR
Related:  tugain  Amount of receiver gain used by qtune (P)
su    Submit a setup experiment to acquisition (M)
tune  Assign frequencies on UNITYNOVA (C)

?  Display individual parameter value (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:**

```plaintext
lb?
sw?
pw[2]?
```

**See also:** *VnmrJ Liquids NMR*

**Related:**

- `display` Display parameters and their attributes (C)
- `getvalue` Get value of a parameter in a tree (C)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>Recall display parameter set (M)</td>
</tr>
<tr>
<td>r1-r7</td>
<td>Real-value storage for macros (P)</td>
</tr>
<tr>
<td>ra</td>
<td>Resume acquisition stopped with sa command (C)</td>
</tr>
<tr>
<td>radialAngles</td>
<td>Radial slice fan angle (P)</td>
</tr>
<tr>
<td>rcvrs</td>
<td>Which receivers to use (P)</td>
</tr>
<tr>
<td>rcvrszt</td>
<td>Weighting for different receivers (P)</td>
</tr>
<tr>
<td>rcvry</td>
<td>Pre-trigger delay (P)</td>
</tr>
<tr>
<td>react</td>
<td>Recover from error conditions during werr processing (M)</td>
</tr>
<tr>
<td>readallshims</td>
<td>Read all shims from hardware (M)</td>
</tr>
<tr>
<td>readbrutape</td>
<td>Read Bruker data files from 9-track tape (U)</td>
</tr>
<tr>
<td>readfile</td>
<td>Read the contents of a text file into two parameters (C)</td>
</tr>
<tr>
<td>readhw</td>
<td>Read current values of acquisition hardware (C)</td>
</tr>
<tr>
<td>readlk</td>
<td>Read current lock level (C)</td>
</tr>
<tr>
<td>readparam</td>
<td>Read one of more parameters from a file (C)</td>
</tr>
<tr>
<td>readultra</td>
<td>Read shim coil setting for Ultra•nmr shim system (M)</td>
</tr>
<tr>
<td>real</td>
<td>Create a real variable without a value (C)</td>
</tr>
<tr>
<td>recon_all</td>
<td>Reconstruct images from 2D MRI fid data (C)</td>
</tr>
<tr>
<td>record</td>
<td>Record keyboard entries as a macro (M)</td>
</tr>
<tr>
<td>redor1</td>
<td>Set up parameters for REDOR1 pulse sequence (M)</td>
</tr>
<tr>
<td>redosy</td>
<td>Restore 2D DOSY display from subexperiment (M)</td>
</tr>
<tr>
<td>refresh</td>
<td>Redraw, refresh overlay (C)</td>
</tr>
<tr>
<td>reffrq</td>
<td>Reference frequency of reference line (P)</td>
</tr>
<tr>
<td>reffrq1</td>
<td>Reference freq. of reference line in 1st indirect dimension (P)</td>
</tr>
<tr>
<td>reffrq2</td>
<td>Reference freq. of reference line in 2nd indirect dimension (P)</td>
</tr>
<tr>
<td>refpos</td>
<td>Position of reference frequency (P)</td>
</tr>
<tr>
<td>refpos1</td>
<td>Position of reference frequency in 1st indirect dimension (P)</td>
</tr>
<tr>
<td>refpos2</td>
<td>Position of reference frequency in 2nd indirect dimension (P)</td>
</tr>
<tr>
<td>refsourc1</td>
<td>Center frequency in 1st indirect dimension (P)</td>
</tr>
<tr>
<td>refsourc2</td>
<td>Center frequency in 2nd indirect dimension (P)</td>
</tr>
<tr>
<td>region</td>
<td>Divide spectrum into regions (C)</td>
</tr>
<tr>
<td>relayh</td>
<td>Set up parameters for RELAYH pulse sequence (M)</td>
</tr>
<tr>
<td>removeAstack</td>
<td>Remove stack (C)</td>
</tr>
<tr>
<td>rename</td>
<td>Move and/or rename a file (C)</td>
</tr>
<tr>
<td>rescal</td>
<td>Calculate pixel size and spatial resolution (M)</td>
</tr>
<tr>
<td>resetf3</td>
<td>Reset parameters after a partial 3D Fourier transform (M)</td>
</tr>
<tr>
<td>resetMovie</td>
<td>Reset movie to the beginning and restore original display (C)</td>
</tr>
<tr>
<td>resolv</td>
<td>Set resolution enhancement parameters (M)</td>
</tr>
<tr>
<td>resto</td>
<td>NMR resonance offset frequency (P)</td>
</tr>
<tr>
<td>restoreStack</td>
<td>Restore stack (C)</td>
</tr>
<tr>
<td>resume</td>
<td>Resume paused acquisition queue (C)</td>
</tr>
<tr>
<td>return</td>
<td>Terminate execution of a macro (C)</td>
</tr>
<tr>
<td>rev</td>
<td>System software revision level (P)</td>
</tr>
<tr>
<td>revdate</td>
<td>System software preparation date (P)</td>
</tr>
</tbody>
</table>
rfband
RF band in use (P)
rfblk
Reverse FID block (C)
rfchannel
Independent control of rf channel selection (P)
rfctype
Type of rf channel (P)
rfcoil
RF pulse calibration identity (P)
rfdata
Reverse FID data (C)
rf1
Reference peak position in directly detected dimension (P)
rf11
Reference peak position in 1st indirectly detected dimension (P)
rf12
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)
rftrace
Reverse FID trace (C)
rfctype
Type of rf generation (P)
rfwg
RF waveform generator (P)
right
Set display limits to right half of screen (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
Change parameters for ROESY experiment (M)
Roesy
Convert the parameter to a ROESY experiment (M)
roesy
Set up parameters for ROESY pulse sequence (M)
Roessyl1d
Convert the parameter set to a Roessyl1d experiment (M)
rof1
Receiver gating time preceding pulse (P)
rof2
Receiver gating time following pulse (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)
RQdisplay
Display images selected by aipDisplayMode (M)
rqfull
Review Queue table width (P)
rqselection
Select images in the Review Queue (P)
rqsort
Sort images in the Review Queue (P)
rqtype
Review Queue type (P)
rsliceplan
Generate absolute magnet frame data (M)
rt
Retrieve FIDs (M)
rtcmx
Return Spinsight data into current experiment (C)
rtphf
Return stored phasefile to current phasefile (C)
rts
Retrieve shim coil settings (C)
rttmp
Retrieve experiment data from experiment subfile (M)
Recall display parameter set (M)

Syntax:
1. \( r \text{set\_number} \)
2. \( r(\text{set\_number}) \)

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 undoing the new phase or drift correction.

Arguments: \( \text{set\_number} \) is the number, from 1 to 9, of a display parameter set.

Examples:
- \( r2 \)
- \( r(3) \)

See also: VnmrJ Liquids NMR

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)

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)

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: VnmrJ Liquids NMR

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)

**radialAngles** Radial slice fan angle (P)

Applicability: Systems with imaging capabilities.
Description: Fan angle of radial slices.
See also: VnmrJ Imaging NMR

**rcvrs** Which receivers to use (P)

Applicability: Systems with multiple receivers.
Description: A string of 'y's and 'n's that indicates which receivers should be used in a multiple receiver acquisition. Setting $rcvrs='y'$ uses only the first receiver, and is equivalent to the parameter being absent.
Examples: $rcvrs='ny'$ uses only the second receiver.
$rcvrs='yyyy'$ uses four receivers.
Related: $numrcvrs$ Number of receivers in the system (P)

**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: $rcvrwt=10,12,8$
Related: $addrcvrs$ Combine data from multiple receivers (M)

**rcvry** Pre-trigger delay (P)

Applicability: Systems with imaging capabilities.
Description: Delays the start of most Varian imaging sequences until after the external trigger (the parameter $ticks$) is received by the system. The delay is still active in the non-triggered mode ($ticks=0$). Setting $hold=0$ removes the delay in the sequence. The delays $rcvry$ and $hold$ are executed once per scan in Varian-provided sequences. In multislice imaging mode, this occurs at the beginning of the multislice pass, but not between the acquisition of individual slices.
Values: 0.1 μs to 8192 sec, in units of seconds.

See also: VnmrJ Imaging NMR

Related: hold | Post-trigger delay (P)
ticks | Number of trigger pulses (P)

**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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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:  *VnmrJ Liquids NMR*

Related:  *convertbru  Convert Bruker data (M,U)*

**readfile**  
Read the contents of a text file into two parameters (C)

**Syntax:**  
```
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 whitespace 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')
```

The `readfile` will return the number of lines added to the parameters.
This sets the `attr` and `vals` parameters to arrays of three strings.

```plaintext
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,

```plaintext
teststr(attr,'H1pwr'):$e
vals[$e] will be the value of 'H1Pwr'
```

### readhw
Read current values of acquisition hardware (C)

**Syntax:**

```plaintext
readhw(param1,param2,...)<:value1,value2,...>
```

**Description:**

Returns or displays the current values of the lock system parameters `lockpower`, `lockgain`, `lockphase`, 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.

`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:**

```plaintext
readhw('z1c','z2c','z1','z2')
readhw('z1c','z2c','z1','z2'):r1,r2,r3,r4
```

**See also:** *VnmrJ Liquids NMR*

**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:**

```plaintext
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:**

```plaintext
readlk
readlk:$levell
```

**See also:** *User Programming*

**Related:**

- `alock` Automatic lock status (P)

### readparam
Read one or more parameters from a file (C)

**Syntax:**

```plaintext
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: *VnmrJ Liquids NMR*

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: Systems with the imaging capabilities.

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.

Arguments: `acqstring`: Set to 'acq' to indicate concurrent reconstruction; performs no initializations. Any other value can be used for retrospective reconstruction or the first pass through concurrent reconstruction (initializations are 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**: for control of some features (see below), `recon_all` accesses parameters in the PROCESSED tree. 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 (see option below), the resulting fdf files are written to the raw subdirectory of the current experiment.

Supported features include:

- Compressed/Standard/Arrayed experiments supported (relevant 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`)
Multi-slice (interleaved) acquisitions (relevant parameter: ns)

Multi-shot/sorting (relevant parameters: petable, etl, and/or nseg)

Multiple receiver data (magnitude sum) (relevant parameter: rcvrs)

Multi-echo imaging support (mems, epi) (relevant parameter: ne)

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:

1. POINTWISE (the default; direct use of the phase of profile)
2. LINEAR (1st order fit of phase of profile)
3. QUADRATIC (2nd order fit of phase of profile)
4. CENTER_PAIR (even/odd pair at center of echo train used for all even/odd echoes)
5. 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 (faems, 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 (ie, 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

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

Examples: recon_all('', '/usr/home/myimages')
recon_all('', '/usr/home/myimages', 'CENTERPAIR')
recon_all('ignorethis', 'LINEAR')
recon_all('acq')

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 UNITY/INOVA systems with a triple-tuned MAS solids probe. This sequence is not supplied with MERCURYplus/Vx systems.

Description: Sets up a parameter set, obtained with XPOLAR1, 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 subexperiment (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: VnmrJ Liquids NMR
Related: dosy Process DOSY experiments (M)
undosy Restore original 1D NMR data from subexperiment (M)
**refresh**

*Redraw, refresh overlay (C)*

**Applicability:** Systems with imaging capabilities.

**Description:** Redraws/refreshes overlays.

See also: *VnmrJ Imaging NMR*

Related: *gplan* Start interactive image planning (C)

**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., `reffpos` is not set to 'n'), the `go, ga, and au` macros calculate values of `rfl` and `rfp` based on `reffrq` and `reffpos`. If referencing is off, `go, ga, and au` set `reffreq` to `sfrq`.

See also: *VnmrJ Liquids NMR*

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)

`reffpos` 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: *VnmrJ Liquids NMR*

Related: *crl1* Clear reference line in 1st indirectly detected dimension (M)

`reffrq` Reference frequency of reference line (P)

`reffpos1` 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: *VnmrJ Liquids NMR*

Related: *crl2* Clear reference line in 2nd indirectly detected dimension (M)

`reffrq` Reference frequency of reference line (P)

`reffpos2` 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: *VnmrJ Liquids NMR*

Related:
- `crl` Clear reference line in directly detected dimension (M)
- `reffrq` Reference frequency of reference line (P)
- `reffpos1` Position of reference frequency in 1st indirect dimension (P)
- `reffpos2` 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 an 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: *VnmrJ Liquids NMR*

Related:
- `crl1` Clear reference line in 1st indirectly detected dimension (M)
- `reffrq1` Ref. frequency of reference line in 1st indirect dimension (P)
- `reffpos` 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: *VnmrJ Liquids NMR*

Related:
- `crl2` Clear reference line in 2nd indirectly detected dimension (M)
- `reffrq2` Ref. frequency of reference line in 2nd indirect dimension (P)
- `reffpos` Position of reference frequency (P)
- `rl2` Set reference line in 2nd indirect dimension (M)
- `setref2` Set frequency referencing for 2nd indirectly detected dimension (M)

reffsource1  
**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 `reffsource1` 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: `VnmrJ Liquids NMR`

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: `VnmrJ Liquids NMR`

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 \( \text{sw} \), in Hz, that is added to the start and end of each calculated peak region; default value is \( \frac{\text{sw}}{10} \). 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 \( \text{fn}, \text{sw}, \text{weighting functions}, \) 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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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)

**removeAstack**  
Remove stack (C)

**Applicability:** Systems with imaging capabilities.

**Syntax:** `removeAstack(index)`

**Description:** Removes the stack with the given index.

**Arguments:** Stack indices begin with zero. If `index` is not given or `index=-1` the selected (active) stack is deleted.

See also: *VnmrJ Imaging NMR*

Related:  
- `gplan` - Start interactive image planning (C)

**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: *VnmrJ Liquids NMR*

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)

**rescal**  
Calculate pixel size and spatial resolution (M)

**Applicability:** Systems with imaging capabilities.

**Syntax:** `rescal<('silent')><:pixrc,pixrd,pixpc,pixpd`

**Description:** Calculates the pixel sizes for the acquisition (spatial resolution) and display (digital resolution). The results are displayed in the text window. As an option, the results can be returned to variables, which allows the user to call `rescal` from within other macros and use it to calculate this basic information. This
macro can be used before acquisition to check that the chosen conditions lead to the desired spatial resolution.

Arguments: 'silent' is a keyword to suppress the text window output.

Arguments:
- `pixrc` returns the readout pixel size (collected).
- `pixrd` returns the readout pixel size (displayed).
- `pixpc` returns the phase encode pixel size (collected).
- `pixpd` returns the phase encode pixel size (displayed).

Examples:
```
rescal
rescal('silent'):r1,r2,r3,r4
```

See also: *VnmrJ Imaging NMR*

**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 $s$ 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: *VnmrJ Liquids NMR*

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)
- `wp` Width of plot (P)

**resetMovie**

Reset movie to the beginning and restore original display (C)

Description: Like `stopMovie`, but rewinds movie to the beginning and restores the original image display.

See also: `startMovie`, `stopMovie`, `continueMovie`

**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: \texttt{resolv} \\
\texttt{resolv(.2,.4)}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{at} \quad \text{Acquisition time (P)}
\texttt{fn} \quad \text{Fourier number in directly detected dimension (P)}
\texttt{gf} \quad \text{Gaussian function in directly detected dimension (P)}
\texttt{lb} \quad \text{Line broadening in directly detected dimension (P)}
\texttt{np} \quad \text{Number of data points (P)}
\texttt{sw} \quad \text{Spectral width in directly detected dimension (P)}

\texttt{resto} \quad \textbf{NMR resonance offset frequency (P)}

Applicability: Systems with imaging capabilities.
Description: NMR resonance offset frequency, in Hz.
See also: \textit{VnmrJ Imaging NMR}
Related: \texttt{tn} \quad \text{Transmitter nucleus (P)}
\texttt{sfrq} \quad \text{Spectrometer frequency (P)}

\texttt{restoreStack} \quad \textbf{Restore stack (C)}

Applicability: Systems with imaging capabilities.
Description: Restores stack to its original spacing and number of slices.
See also: \textit{VnmrJ Imaging NMR}
Related: \texttt{gplan} \quad \text{Start interactive image planning (C)}

\texttt{resume} \quad \textbf{Resume paused acquisition queue (C)}

Description: Enables continuing submitting experiments to the acquisition system. For experiments initiated with the command \texttt{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. \texttt{resume} then allows the data processing macro to initiate another acquisition with \texttt{au('next')}, which is then performed immediately instead of at the end of the queue.
See also: \textit{VnmrJ Liquids NMR}
Related: \texttt{au} \quad \text{Submit experiment to acquisition and process data (C)}

\texttt{return} \quad \textbf{Terminate execution of a macro (C)}

Syntax: \texttt{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. \texttt{return} is used only in macros and not entered from the keyboard.
Arguments: \texttt{expression1,expression2,...} are return values to another calling macro.
See also: \textit{User Programming}
Related: \texttt{abort} \quad \text{Terminate action of calling macro and all higher macros (C)}

\texttt{rev} \quad \textbf{System software revision level (P)}

Description: Stores a string identifying the \textit{VnmrJ} software version for the system. This parameter is not be entered by the user, but can be examined by entering \texttt{rev?}. 

See also: *VnmrJ Installation and Administration*

**revdate**  
**System software preparation date (P)**

**Description:** Stores a string identifying the date the current VnmrJ software version was prepared. This parameter is not 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)**

**Applicability:** All systems except MERCURYplus/Vx.

**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:** *VnmrJ Liquids NMR*

**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`:

```bash
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: $\text{rfblk}(1, 2, 1)$ reverses and copies block 1 from the current experiment to block 1 of experiment 2.

See also: *User Programming*

Related:  
- $\text{mfblk}$: Move FID block (C)  
- $\text{mfclose}$: Memory map close FID file (C)  
- $\text{mfdata}$: Move FID data (C)  
- $\text{mfopen}$: Memory map open FID file (C)  
- $\text{mftrace}$: Move FID trace (C)  
- $\text{rfdata}$: Reverse FID data (C)  
- $\text{rftrace}$: Reverse FID trace (C)

**rfchannel**

Independent control of rf channel selection (P)

Applicability: *UNITY/INOVA* systems.

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')`.

On *UNITY/INOVA* systems, 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`, `dseq`, and `homo`. 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 are `dfrq2`, `dof2`, `dpwr2`, `dpwrf2`, `dmf2`, `dm2`, `dmm2`, `dres2`, `dseq2`, and `homo2`. The third decoupler would use parameters with a 3 appended: `dfrq3`, `dof3`, `dpwr3`, `dpwrf3`, `dmf3`, `dm3`, `dmm3`, `dres3`, `dseq3`, and `homo3`. 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 `dfrq3`, `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 CONFIG 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: *VnmrJ Liquids NMR; User Programming*

**rfchtype**

*Type of rf channel (P)*

**Applicability:** UNITY INOVA systems.

**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 CONFIG window (opened by entering `config`). Pulse sequence programs check `rfchtype` to determine if indirect detection should be used for some experiments. Indirect detection occurs automatically on a UNITY INOVA if the decoupler is set to U+ H1 Only in the CONFIG 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 `rfchtype` is \('d'\) on the U+ Direct Synthesis and U+ H1 Only entries.

- 'U+ Direct Synthesis' is the setting for a UNITY INOVA with direct synthesis (U+ Direct Synthesis in the CONFIG window).
- 'U+ H1 Only' is a fixed-frequency proton UNITY INOVA (U+ H1 Only in CONFIG window).
- 'Deuterium Decoupler' is the setting for a UNITY INOVA deuterium decoupler channel.
- 'Direct Synthesis' is the setting for direct synthesis (Direct Synthesis in the CONFIG window).
- 'Broadband' is the setting for broadband (Broadband in the CONFIG window).
- 'Fixed Frequency' is the setting for fixed frequency (Fixed Frequency in the CONFIG window).
- 'SIS Modulator' is the setting for imaging modulator (SIS Modulator in the CONFIG window).

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)
- `homo` Homodecoupling control 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)
rfcoil

**RF pulse calibration identity (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Contains a string identifying the rf pulse calibration.

See also: *VnmrJ Imaging NMR*

**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: mfb1k  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)
rfb1k  Reverse FID block (C)
rftrace  Reverse FID trace (C)

rf1  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, rf1 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: VnmrJ Liquids NMR
Related: rf1  Reference peak position in 1st indirectly detected dimension (P)
rf12  Reference peak position in 2nd indirectly detected dimension (P)
rfp  Reference peak frequency in directly detected dimension (P)

rf11  Reference peak position in 1st indirectly detected dimension (P)
Description: Analogous to the rf1 parameter except that rf11 applies to the first indirectly detected dimension of a multidimensional data set. rf11 can either be set manually or be adjusted automatically when the macro rli is used to assign a reference line.
Values: Number, in Hz.
See also: VnmrJ Liquids NMR
Related: rf1  Reference peak position in directly detected dimension (P)
rf12  Reference peak position in 2nd indirectly detected dimension (P)
rfp1  Reference peak frequency in 1st indirectly detected dimension (P)

rf12  Reference peak position in 2nd indirectly detected dimension (P)
Description: Analogous to the rf1 parameter except that rf12 applies to the second indirectly detected dimension of a multidimensional data set. rf12 can either be set manually or be adjusted automatically when the macro r12 is used to assign a reference line.
Values: Number, in Hz.
See also: VnmrJ Liquids NMR
Related: rf1  Reference peak position in directly detected position (P)
rf11  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.
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: VnmrJ Liquids NMR

Related: rf1, rfp1, rfp2, rl

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: VnmrJ Liquids NMR

Related: rf1, rfp, rf2, rl1

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:
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 the 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)
- `mfddata` Move FID data (C)
- `mfopen` Memory map open FID file (C)
- `mtrace` 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 MERCURYplus/Vx systems, the value is set using the System Type label in the CONFIG window (opened by entering `config`). On other systems, the value is set using the Type of RF label in the CONFIG window.

**Values:** The values of `rftype` parallel the `rfchtype` values. The only distinction is that on `UNITY INOVA`, the setting for `rftype` is 'd' on the entries U+ Direct Synthesis and U+ H1 Only. On the MERCURYplus/Vx, only 'ee' or 'fe' is used.

- 'd' is the setting for a `UNITY INOVA` with direct synthesis (U+ Direct Synthesis in the CONFIG window) or a fixed-frequency proton `UNITY INOVA` (U+ H1 Only in CONFIG window).
- 'l' is the setting for a `UNITY INOVA` deuterium decoupler channel.
- 'c' is the setting for direct synthesis (Direct Synthesis in the CONFIG window).
- 'b' is the setting for broadband (Broadband in the CONFIG window).
- 'a' is the setting for fixed frequency (Fixed Frequency in the CONFIG window).
- 'm' is the setting for imaging modulator (SIS Modulator in the CONFIG window).
- 'ee' is the setting for 4-nucleus, MERCURYplus/Vx 4-nucleus or 1H/13C systems (4 Nucleus or 1H/13C in the CONFIG window).
- 'fe' is the setting for MERCURYplus/Vx broadband systems (Broadband in the CONFIG window).
rfwg  RF waveform generator (P)
Applicability: Not available on MERCURYplus/Vx.
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 CONFIG window (opened by entering config).
Values: ‘n’ is setting for no waveform generator board on the channel (Not Present choice in CONFIG window).
‘y’ is setting for a waveform generation board on the channel (Present choice in CONFIG 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: VnmrJ Liquids NMR
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)
w

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: VnmrJ Liquids NMR; 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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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)

Applicability: All systems; however, although rl2 is available on MERCURYplus/Vx, such systems can only process 3D data and cannot acquire 3D data.

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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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.
ROESY  
**Change parameters for ROESY experiment (M)**
Description: Converts the current parameter set to a ROESY experiment.

Roesy  
**Convert the parameter to a ROESY experiment (M)**
Description: Convert the parameter to a ROESY experiment.

roesy  
**Set up parameters for ROESY pulse sequence (M)**
Syntax: roesy<(ratio)>
Description: Sets up a rotating frame Overhauser effect spectroscopy experiment.
Arguments: ratio is the value of the parameter ratio used in the sequence (ratio is not used in the ROESY sequence provided with MERCURYplus/-Vx).

Roesy1d  
**Convert the parameter set to a Roesy1d experiment (M)**
Description: Convert the parameter set to a Roesy1d experiment.
See also: Proton(M) selld(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/$^1$H amplifiers have a short turn-on time, usually 1 to 5 $\mu$s 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 $\mu$s.
Values: Typically 2-5 seconds.
See also: VnmrJ Liquids NMR
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 seconds.
See also: VnmrJ Liquids NMR
Related: rof1  
Receiver gating time preceding pulse (P)

rotate  
**Rotate 2D data (C)**
Syntax: rotate<(number_degrees)>
Description: Rotates a 2D spectrum. Both complex and hypercomplex 2D data will work.
Arguments: \texttt{number\_degrees} is the amount of counter-clockwise rotation, in degrees. The default is 45.

See also: \textit{VnmrJ Liquids NMR}

\textbf{rotorsync} \hspace{1cm} \textbf{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 \texttt{rotorsync} is set using the Rotor Synchronization label in the CONFIG window (opened by entering \texttt{config}). 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 CONFIG window).

0 is setting that system does not have solid rotor synchronization (Not Present choice in the CONFIG window).

See also: \textit{VnmrJ Installation and Administration}

Related: \texttt{config} \hspace{1cm} Display current configuration and possibly change it (M)

\textbf{rp} \hspace{1cm} \textbf{Zero-order phase in directly detected dimension (P)}

Description: Specifies the right phase-correction angles along the directly detected dimension according to

\[
\text{absorption spectrum(\omega)} = \\
\text{real channel(\omega) * sin } \theta + \text{imaginary channel(\omega) * cos } \theta
\]

where the phase angle $\theta$ is a function of frequency:

\[
\theta = \text{rp} + (\omega - \omega_o) * \text{lp}
\]

$\omega_0$ is defined as the right end of the spectrum. This dimension is referred to as the f2 dimension in 2D data sets, f3 dimension in 3D data sets, and so on.

Values: $-360$ to $+360$, in degrees.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{aph} \hspace{1cm} Automatic phase adjustment of spectra (C)

\texttt{aph0} \hspace{1cm} Automatic phase of zero-order term (C)

\texttt{lp} \hspace{1cm} First-order phase in directly detected dimension (P)

\texttt{rp1} \hspace{1cm} Zero-order phase in 1st indirectly detected dimension (P)

\texttt{rp2} \hspace{1cm} Zero-order phase in 2nd indirectly detected dimension (P)

\textbf{rp1} \hspace{1cm} \textbf{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 f1 dimension of a multidimensional data set during the process of phase-sensitive 2D transformation.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{lp1} \hspace{1cm} First-order phase in 1st indirectly detected dimension (P)

\texttt{rp} \hspace{1cm} Zero-order phase in directly detected dimension (P)

\texttt{rp2} \hspace{1cm} 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 f2 dimension.

**See also:** *VnmrJ Liquids NMR*

**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)

---

**RQdisplay**

Display images selected by *aipDisplayMode* (M)

**Syntax:** `RQdisplay(<'batch', n/'next'/prev'/refresh'/userselection>)`  

**Description:**
This macro displays images selected by *aipDisplayMode*:
- *aipDisplayMode*=1, display all loaded images.
- *aipDisplayMode*=3, display images selected in the Review Queue
- *aipDisplayMode*=4, display images in selected frames
- *aipDisplayMode*=5, display images selected by parameter 'userselection'.

3 and 5 works on for review viewport.

The layout depends on *aipDisplayMode*:
- *aipAutoLayout*=0, use current frame layout, if not enough frames, display images in batches.
- *aipAutoLayout*=1, automatically layout the frames for all images.

For the Review viewport, the images of multiple scans can be globally sorted according to *rqsort*:
- *rqsort*=0, display images in frames specified in RQ, or by user.
- *rqsort*=1, sort images by scan, ignore "frames" specification.
- *rqsort*=2, sort images by slice, ignore "frames" specification.

**Arguments:**
- 'batch', to display batch specified by next augment (an integer).
- 'next', to display next batch.
- 'prev', to display previous batch.
- 'refresh', to refresh the displayed.
- 'userselection', to display images selected by string userselection (works only for review viewport).

**Examples:**
- `RQdisplay('g1(1-9:2)[6-]')`
to display every other images in group 1, starts from frame 6.

---

**rqfull**

Review Queue table width (P)

**Description:**
An integer of value 1/0 to indicate the Review Queue table is full width, or fitted width.

---

**rqselection**

Select images in the Review Queue (P)

**Description:**
A string for selecting images and frames (selection syntax) Used to change selections in RQ table.

**Examples:**
- `g1-3, g1(1-4)[5-]`

---

**rqsort**

Sort images in the Review Queue (P)

**Description:**
Parameter to set global sorting of image display.
Values: 0, no sorting, use frames as specified in Review Queue
1, sort by scans
2, sort by slices.

rqtype  
Review Queue type (P)
Description: Review Queue type only 'imgstudy' is implemented.
Examples: rqtype='imgstudy' to review image studies.

rsliceplan  
Generate absolute magnet frame data (M)
Applicability: Systems with imaging capabilities.
Description: rsliceplan is a helper macro toiplan image planning. It combines the
iplan data with sequence parameters to generate the absolute magnet frame
data. Users without imaging capabilities should use sliceplan.
See also: VnmrJ Imaging NMR
Related:  
iplan  
Open interactive image planning tools (M)
sliceplan  
Set slice parameters for target slice (M)

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: VnmrJ Liquids NMR
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)

rtcmsgx  
Return Spinsight data into current experiment (C)
Syntax: rtcmsgx<(file)>
R

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: rtcnx
rcmcx('redor.data')
See also: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
Related: fixpar Correct parameter characteristics in experiment (M)
rt Retrieve FIDs (M)
rtv Retrieve individual parameters (C)
svp Save parameters from current experiment (M)

rtphf

Return stored phasefile to current phasefile (C)
Applicability: Systems with imaging capabilities.
Syntax: rtphf(file)
Description: Copies a stored phasefile (curexp+'/planes/file', where file is the file name given in the argument) into the phasefile of the current experiment (curexp+'/datdir/phasefile'). This allows the display and manipulation of previously transformed images, provided the parameter values in the current experiment are compatible with the parameter values present in the experiment that generated the stored phasefiles at the time they were stored.
Arguments: file is the file name of the stored phase file. Use only relative path names for file, not absolute path names (i.e., use path names beginning with "/").
Examples: rtphf('waldo')
See also: VnmrJ Imaging NMR
Related: curexp Current experiment directory (P)
imcalc Calculate 2D phasefiles (M,U)
makephf Transform and save images as phasefiles (M)
svphf Save current phasefile (C)

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 \texttt{su} (or related commands or macros). If the shim file is not found, \texttt{rts} displays the file names it tried.

The \texttt{rts} command returns shims from a .\texttt{fid} file or a .\texttt{par} file, selecting the shim parameters from the parameters stored there.

Arguments: \texttt{file} is the name of a file containing the shim coil settings to be retrieved. If the file name is an absolute path, \texttt{rts} uses it with no modifications. Otherwise, \texttt{rts} searches up to three different directories, as follows:

- First, \texttt{rts} looks for a 	exttt{shims} subdirectory in your user directory. If 	exttt{shims} exists, it looks for the requested file name there.
- Next, if 	exttt{shims} does not exist, \texttt{rts} then looks for the global parameter \texttt{shimspath}. If \texttt{shimspath} is present, it is expected to contain the name of a directory. If this directory exists, \texttt{rts} looks for the file in that directory.
- Finally, if this does not work, \texttt{rts} searches in the \texttt{shims} subdirectory of the system directory.

\texttt{status} is a return variable with one of the following values after \texttt{rts} finishes searching for the shim coil settings file:

- 0 indicates that \texttt{rts} failed to find requested file.
- 1 indicates that \texttt{rts} found the requested file, either as an absolute path or in the \texttt{shims} subdirectory of the user directory.
- 2 indicates that \texttt{rts} found the requested file using the global parameter \texttt{shimspath}.
- 3 indicates that \texttt{rts} found the requested file in \texttt{shims} subdirectory of the system directory.

Examples: \texttt{rts('acetone')}
\texttt{rts('bb10mm'):rl}

See also: \textit{VnmrJ Liquids NMR}

\begin{itemize}
\item \texttt{load} \quad Load status of displayed shims (P)
\item \texttt{shimspath} \quad Path to user's 	exttt{shims} directory (P)
\item \texttt{su} \quad Submit a setup experiment to acquisition (M)
\item \texttt{svs} \quad Save shim coil settings (C)
\end{itemize}

\textbf{rttmp}

\textbf{Retrieve experiment data from experiment subfile (M)}

\textbf{Syntax:} \texttt{rttmp(file)}

\textbf{Description:} Retrieves experiment data—parameters, FID, and transformed spectrum—from the file specified in a subdirectory inside \texttt{curexp'/subexp'}.

\textbf{Arguments:} \texttt{file} is the name of the subfile from which to retrieve the experiment data.

Examples: \texttt{rttmp('H1')}
\texttt{rttmp('cosy')}

See also: \textit{VnmrJ Liquids NMR}

\begin{itemize}
\item \texttt{captain} \quad Copy experiment data into experiment subfile (M)
\item \texttt{curexp} \quad Current experiment directory (P)
\item \texttt{svtmp} \quad Move experiment data into experiment subfile (M)
\end{itemize}

\textbf{rtv}

\textbf{Retrieve individual parameters (C)}

\textbf{Syntax:} \texttt{rtv<file,par1<,index1<,par2,index2...>>:<val>}

\textbf{Description:} Retrieves one or more parameters from a parameter file. The file might have been made with \texttt{svf} or \texttt{svp} or \texttt{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.

If a return argument is added, rtv returns values into the macro. This form of rtv command, in which values are passed only to macro variables, is useful if you do not want additional parameters created in the experiment’s current tree.

Arguments:
- **file** is the 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,...** are the 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** is a return argument for values to return to the macro.

If the requested parameter do not exist in the parameter file, rtv will abort. There is only one exception. If a single parameter is requested and it is being returned into a macro parameter and the 'noabort' option is given to the command, it will not abort if the parameter does not exist. An example is rtv('parmaster','parameter','noabort'):$pm

The noabort option must follow the 'parameter' keyword and precede the optional tree argument. If rtv is executed without macro return values, then the fixpar macro will automatically be run. If return values are requested, fixpar is not executed. If these commands are executed without an argument, they will ask for a filename. In that case, the filename can be given without 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')

See also: VnmrJ Liquids NMR

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)

**Description:** The rtx command retrieves parameters from filename, based on the setting of the P_LOCK protection bit and using the rules below.

- keyword1 may be "keep" or "rt". Default is keep. keyword2 may be "clear" or "noclear". Default is clear. keyword2 determines if the P_LOCK bit is cleared after it is rtx'ed.

<table>
<thead>
<tr>
<th>Status of P_LOCK bit in current exp.</th>
<th>Status of P_LOCK bit in filename.</th>
<th>keyword1</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>on</td>
<td>keep or rt</td>
<td>do not rt</td>
</tr>
<tr>
<td>on</td>
<td>off</td>
<td>keep or rt</td>
<td>do not rt</td>
</tr>
<tr>
<td>Status of P_LOCK bit in current exp.</td>
<td>Status of P_LOCK bit in filename</td>
<td>keyword1</td>
<td>result</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>keep or rt</td>
<td>do rt</td>
</tr>
<tr>
<td>off</td>
<td>off</td>
<td>keep</td>
<td>do not rt</td>
</tr>
<tr>
<td>off</td>
<td>off</td>
<td>rt</td>
<td>do rt</td>
</tr>
<tr>
<td>&lt;no parameter&gt;</td>
<td>on</td>
<td>keep or rt</td>
<td>do rt</td>
</tr>
<tr>
<td>&lt;no parameter&gt;</td>
<td>off</td>
<td>keep</td>
<td>do not rt</td>
</tr>
<tr>
<td>&lt;no parameter&gt;</td>
<td>off</td>
<td>rt</td>
<td>do rt</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>Save display parameters as a set (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s2pul</td>
<td>Set up parameters for standard two-pulse sequence (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sa</td>
<td>Stop acquisition (C)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sample</td>
<td>Submit change sample, Autoshim experiment to acquisition (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>save</td>
<td>Save data (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>savefile</td>
<td>Base file name for saving files (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>samplename</td>
<td>Sample name (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>saveglobal</td>
<td>Save selected parameters from global tree (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>saveMilestoneStacks</td>
<td>Save current planning as milestone (C)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>savePrescription</td>
<td>Save current planning to file (C)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sb</td>
<td>Sinebell constant in directly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sb1</td>
<td>Sinebell constant in 1st indirectly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sb2</td>
<td>Sinebell constant in 2nd indirectly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sb1</td>
<td>Sinebell shift in 1st indirectly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sb1</td>
<td>Sinebell shift in 2nd indirectly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sb2</td>
<td>Sinebell shift in 2nd indirectly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sc</td>
<td>Start of chart (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sc2</td>
<td>Start of chart in second direction (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scalelimits</td>
<td>Set limits for scales in regression (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scalesw</td>
<td>Set scaling factor for multipulse experiments (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scalesw</td>
<td>Scale spectral width in directly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scalesw1</td>
<td>Set f_1 scaling factor for 2D multipulse experiments (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scalesw1</td>
<td>Scale spectral width in 1st indirectly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scalesw2</td>
<td>Scale spectral width in 2nd indirectly detected dimension (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sd</td>
<td>Set first decoupler frequency to cursor position (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sd2</td>
<td>Set second decoupler frequency to cursor position (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sd3</td>
<td>Set third decoupler frequency to cursor position (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sda</td>
<td>Set first decoupler frequency array (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sda2</td>
<td>Set second decoupler frequency array (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sda3</td>
<td>Set third decoupler frequency array (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sdp</td>
<td>Show diffusion projection (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sediff</td>
<td>Set up spin-echo diffusion imaging sequence (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>selld</td>
<td>Execute protocol actions of apptype selld (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>select</td>
<td>Select spectrum, FID, trace, or 2D plane without display (C)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>selex</td>
<td>Defines excitation band (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>selexcit</td>
<td>Set up PFG selective excitation pulse sequence (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sems</td>
<td>Set up basic imaging sequence with oblique capability (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>send2vnmr</td>
<td>Send a command to VnmrJ (U)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seqcon</td>
<td>Acquisition loop control (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seqfil</td>
<td>Pulse sequence name (P)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seggen</td>
<td>Initiate compilation of user’s pulse sequence (M,U)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set2D</td>
<td>General setup for 2D experiments (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set2d</td>
<td>General setup for 2D experiments (M)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
set3dproc Set 3D processing (C)
setallshims Set all shims into hardware (M)
setarray Set up a parameter array (M)
setcenter Set up parameters for center sequence calibration (M)
setcolor Set colors for graphics window and for plotters (C)
setdecpars Set decoupler parameter values from probe file (M)
setdec2pars Set decoupler 2 parameter values from probe file (M)
setDefaultSize Set FOV to default size (C)
setDefaultSlices Set default number of slices (C)
setDefaultThk Set default slice thickness (C)
setDefaultType Set default type (C)
setDisplayStyle Show stripes or lines (C)
setDrawInterSection Show/hide intersection (C)
setDraw3D Show/hide 3D (C)
setDrawAxes Show/hide axes (C)
setDrawOrders Show/hide order of drawings (C)
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)
setFillPolygon Show/hide filled polygon (C)
setflip Set rf power levels to desired flip angle (M)
setfrq Set frequency of rf channels (C)
setGapMode Fix/Unfix slice gap (C)
setgauss Set a Gaussian fraction for lineshape (M)
setgcal Set the gradient calibration constant (M)
setgcoil Assign sysgcoil configuration parameter (M)
setgpe Set phase encode gradient levels (M)
setgrid Divide graphics window into rows and columns (C)
setgro Set readout gradient (M)
setgroup Set group of a parameter in a tree (C)
setgss Select slice or voxel selection gradient levels (M)
sethw 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)
setloop Control arrayed and real-time looping (M)
setLP1 Set F1 linear prediction parameters (M)
setMarkMode Remove/activate mark (C)
setnoether Disconnect host computer from Ethernet (U)
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 $\gamma B_1$ value (M)
setprotect Set protection mode of a parameter (C)
setref Set frequency referencing (M)
setref1  Set freq. referencing for 1st indirectly detected dimension (M)
setref2  Set freq. referencing for 2nd indirectly detected dimension (M)
setscout  Set up a scout run (M)
setssfilter  Set sslsfreq 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)
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)
setvalue  Set value of any parameter in a tree (C)
setValue  Set parameter values (C)
setwave  Write a wave definition string into Pbox.inp file (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)
ssfreq  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)
shimset  Type of shim set (P)
shimpath  Path to user’s shims directory (P)
showconsole  Show UNITYINOVA console configuration parameters (U)
showfit  Display numerical results of deconvolution (M)
showloginbox  Shows operator login dialog (M)
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)
showstat  Display information about status of acquisition (M,U)
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)
sliceorder  Reorder the slice position list (M)
sliceplan  Set slice parameters for target slice (M)
slp  Family of offset Frequencies of SLP shapes (P)
slw  Spin simulation linewidth (P)
smaxf  Maximum frequency of any transition (P)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sminf</td>
<td>Minimum frequency of any transition (P)</td>
</tr>
<tr>
<td>smsport</td>
<td>Sample Management System serial port connection (P)</td>
</tr>
<tr>
<td>sn</td>
<td>Signal-to-noise ratio (P)</td>
</tr>
<tr>
<td>solppm</td>
<td>Return ppm and peak width of solvent resonances (M)</td>
</tr>
<tr>
<td>solvent</td>
<td>Lock solvent (P)</td>
</tr>
<tr>
<td>solvinfo</td>
<td>Retrieve information from solvent table (C)</td>
</tr>
<tr>
<td>sort</td>
<td>Sort real values of a parameter (M)</td>
</tr>
<tr>
<td>sp</td>
<td>Start of plot in directly detected dimension (P)</td>
</tr>
<tr>
<td>spl</td>
<td>Start of plot in 1st indirectly detected dimension (P)</td>
</tr>
<tr>
<td>sp2</td>
<td>Start of plot in 2nd indirectly detected dimension (P)</td>
</tr>
<tr>
<td>spadd</td>
<td>Add current spectrum to add/subtract experiment (C)</td>
</tr>
<tr>
<td>spcfrq</td>
<td>Display frequencies of rf channels (M)</td>
</tr>
<tr>
<td>specdc3d</td>
<td>3D spectral dc correction (P)</td>
</tr>
<tr>
<td>spin</td>
<td>Submit a spin setup experiment to acquisition (C)</td>
</tr>
<tr>
<td>spinopt</td>
<td>Sample spin rate (P)</td>
</tr>
<tr>
<td>spincad</td>
<td>Run SpinCAD program (C)</td>
</tr>
<tr>
<td>spinll</td>
<td>Set up a slfreq array (M)</td>
</tr>
<tr>
<td>spinner</td>
<td>Open the Spinner Control window (C)</td>
</tr>
<tr>
<td>spinopt</td>
<td>Spin automation (P)</td>
</tr>
<tr>
<td>spins</td>
<td>Perform spin simulation calculation (C)</td>
</tr>
<tr>
<td>split</td>
<td>Split difference between two cursors (M)</td>
</tr>
<tr>
<td>spmax</td>
<td>Take the maximum of two spectra (C)</td>
</tr>
<tr>
<td>spmin</td>
<td>Take minimum of two spectra in add/subtract experiment (C)</td>
</tr>
<tr>
<td>spsm</td>
<td>Enter spin system (M)</td>
</tr>
<tr>
<td>spsub</td>
<td>Subtract current spectrum from add/subtract experiment (C)</td>
</tr>
<tr>
<td>sqcosine</td>
<td>Set up unshifted cosine-squared window function (M)</td>
</tr>
<tr>
<td>sqdir</td>
<td>Study queue directory (P)</td>
</tr>
<tr>
<td>sqname</td>
<td>Study queue parameter template (P)</td>
</tr>
<tr>
<td>sqrt</td>
<td>Return square root of a real number (O)</td>
</tr>
<tr>
<td>sqsinebell</td>
<td>Set up unshifted sinebell-squared window function (M)</td>
</tr>
<tr>
<td>srate</td>
<td>Spinning rate for magic angle spinning (P)</td>
</tr>
<tr>
<td>sread</td>
<td>Read converted data into VnmrJ (C)</td>
</tr>
<tr>
<td>ss</td>
<td>Steady-state transients (P)</td>
</tr>
<tr>
<td>ssecho</td>
<td>Set up solid-state echo pulse sequence (M)</td>
</tr>
<tr>
<td>ssechol</td>
<td>Set up parameters for SSECHO1 pulse sequence (M)</td>
</tr>
<tr>
<td>ssfilter</td>
<td>Full bandwidth of digital filter to yield a filtered FID (P)</td>
</tr>
<tr>
<td>sslsfrq</td>
<td>Center of solvent-suppressed region of spectrum (P)</td>
</tr>
<tr>
<td>ssntaps</td>
<td>Number of coefficients in digital filter (P)</td>
</tr>
<tr>
<td>ssorder</td>
<td>Order of polynomial to fit digitally filtered FID (P)</td>
</tr>
<tr>
<td>ssplan</td>
<td>Set slice parameters for target slice (M)</td>
</tr>
<tr>
<td>sslist</td>
<td>Conjugate gradient list (P)</td>
</tr>
<tr>
<td>ssprep</td>
<td>Calculate slice gradient and slice selection parameters (M)</td>
</tr>
<tr>
<td>stack</td>
<td>Stacking mode for processing and plotting arrayed spectra (M)</td>
</tr>
<tr>
<td>stackmode</td>
<td>Stacking control for processing arrayed 1D spectra (P)</td>
</tr>
<tr>
<td>startIplan</td>
<td>Start/restart image planning (C)</td>
</tr>
<tr>
<td>startMovie</td>
<td>Start running a movie (C)</td>
</tr>
</tbody>
</table>
status
std1d
stdshm
steam
stepMovie
sth
stopMovie
string
strtext
strtext1
strtext2
strtlp
strtlp1
strtlp2
studyid
su
sub
substr
uselfreq
svdat
svf
svfdf
svfdir
svfname
svib
svp
svphf
svs
svsp
svs
svtmp
sw
sw1
sw2
sw3
sysgcoil
system
systemdir

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: \texttt{s2}  
\texttt{s(3)}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{fr} Full recall of display parameter set (M)  
\texttt{r} Recall display parameter set (M)

\texttt{s2pul} \textbf{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: \textit{VnmrJ Liquids NMR}

\texttt{sa} \textbf{Stop acquisition (C)}

Applicability: All systems; however, the \texttt{option} and \texttt{number} arguments are unavailable on \textit{MERCUryplus/Vx} systems.

Syntax: \texttt{sa<\{option\|number\}>}

Description: Stops an experiment that has been submitted to acquisition. If experiment is active, it is stopped. Data is retained. \texttt{sa} applies to the experiment that you are joined to at the time the \texttt{sa} command is entered. Thus, if experiment 1 is active, you must be joined to experiment 1 for \texttt{sa} to stop that acquisition. If you are in experiment 2, entering \texttt{sa} has no effect on experiment 1.

When experiments are queued, the behavior of \texttt{sa} is more complex. If an experiment is active in \texttt{exp1} and queued in \texttt{exp2}, entering \texttt{sa} from \texttt{exp1} stops that experiment and immediately begins acquisition on \texttt{exp2}. Entering \texttt{sa} from \texttt{exp2}, on the other hand, removes \texttt{exp2} from the queue, without affecting the active experiment 1.

Entering \texttt{sa} from an experiment that is not active or queued has no effect.

Arguments: \texttt{option} is one of the following:
- \texttt{'eos'}, \texttt{'ct'}, \texttt{'scan'} are keywords to stop at the next \texttt{ct}.
- \texttt{'eob'}, \texttt{'bs'} are keywords to stop at the next block size.
- \texttt{'eof'}, \texttt{'nt'}, \texttt{'fid'} are keywords to stop at the next complete FID.
- \texttt{'eoc'}, \texttt{'il'} are keywords to stop at next complete \texttt{il} cycle (i.e., the latest block size that has been completed for all FIDs in interleave cycle).

\texttt{number} is an integer number to stop at the next \texttt{ct}, where the value of \texttt{ct} is a multiple of \texttt{number}. This is useful when you want to complete a phasecycle before stopping.

Examples: \texttt{sa}  
\texttt{sa('ct')}  
\texttt{sa(4)}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{bs} Block size (P)  
\texttt{ct} Completed transients (P)  
\texttt{il} Interleave arrayed and 2D experiments (P)  
\texttt{nt} Number of transients (P)  
\texttt{ra} Resume acquisition stopped with \texttt{sa} command (C)

\texttt{sample} \textbf{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: *VnmrJ Liquids NMR*

**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)

**save**

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.

**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: *VnmrJ Liquids NMR*

Related: `parlc` Create LC-NMR parameters (M)

**sample**

**Sample name (P)**

Description: Specifies the name of the sample. It is saved with a liquids study.

See also: *notebook (P) page (P)*

**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: *VnmrJ Liquids NMR*

Related: `go` Submit experiment to acquisition (C)
- `loc` Location of sample in tray (P)

**saveMilestoneStacks**

**Save current planning as milestone (C)**

Applicability: Systems with imaging capabilities.

Description: Saves current planning as a milestone prescription. Milestone is saved in both memory and to a file.

See also: *VnmrJ Liquids NMR*

Related: `gplan` Start interactive image planning (C)
**savePrescription**

Save current planning to a given file.

**Syntax:** `savePrescription(char* path)`

**Description:** Save current planning to a file.

**Applicability:** Systems with imaging capabilities.

**See also:** *VnmrJ Liquids NMR*

**Related:** `gplan` Start interactive image planning

---

**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 $\sin\left(\frac{t\cdot\pi}{2\cdot sb}\right)$
- A negative value applies a squared sinebell function of form $\sin^2\left(\frac{t\cdot\pi}{2\cdot sb}\right)$

$sb$ is given in seconds. Typical value is $sb='n'$.

**See also:** *VnmrJ Liquids NMR*

**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 $\sin\left(\frac{t\cdot\pi}{2\cdot sb1}\right)$
- A negative value applies a squared sinebell function of form $\sin^2\left(\frac{t\cdot\pi}{2\cdot sb1}\right)$

$sb1$ is given in seconds. Typical value is $sb1='n'$.

**See also:** *VnmrJ Liquids NMR*

**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 $\sin\left(\frac{t\cdot\pi}{2\cdot sb2}\right)$
- A negative value applies a squared sinebell function of form $\sin^2\left(\frac{t\cdot\pi}{2\cdot sb2}\right)$

$sb2$ is given in seconds. Typical value is $sb2='n'$.

**See also:** *VnmrJ Liquids NMR*

**Related:**
- `sb` Sinebell constant in directly detected dimension (P)
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
\[
\sin \left( \frac{(t-sbs) \cdot \pi}{2 \cdot sb} \right)
\]
The square of this function is applied if \( sb \) is negative. \( sbs \) is given in seconds. The typical value is \( sbs = 'n' \).

See also: VnmrJ Liquids NMR

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
\[
\sin \left( \frac{(t-sbs1) \cdot \pi}{2 \cdot sb1} \right)
\]
The square of this function is applied if \( sb1 \) is negative. \( sbs1 \) is given in seconds. The typical value is \( sbs1 = 'n' \).

See also: VnmrJ Liquids NMR

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
\[
\sin \left( \frac{(t-sbs2) \cdot \pi}{2 \cdot sb2} \right)
\]
The square of this function is applied if \( sb2 \) is negative. \( sbs2 \) is given in seconds. The typical value is \( sbs2 = 'n' \).

See also: VnmrJ Liquids NMR

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 \( w_{\text{cmax}} \), in mm
See also: *VnmrJ Liquids NMR*
Related: 
- \( sc \) Start of chart (P)
- \( wc \) Width of chart (P)
- \( wc_{\text{max}} \) 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 \( wc_{\text{c}} \) controls the width of the chart.
Values: 0 to \( wc_{\text{cmax}} \), in mm.
See also: *VnmrJ Liquids NMR*
Related: 
- \( sc \) Start of chart (P)
- \( wc \) Width of chart (P)
- \( wc_{\text{c}} \) Maximum width of chart (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: \( x_{\text{start}}, x_{\text{end}}, y_{\text{start}}, y_{\text{end}} \) are \( x \)-axis and \( y \)-axis starting and ending limits. The default is that `scalelimits` prompts for the limits.
See also: *VnmrJ Liquids NMR, 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*

**scalesw1**
Set f₁ scaling factor for 2D multipulse experiments (M)

Description: Sets the f₁ 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 BR24 multiple pulse experiment (M)
- `mrev8`: Set up MREV8 multiple pulse experiment (M)
- `r1`: 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)
- `etail1`: Set up solid-state echo pulse sequence (M)
- `xpolar1`: Set up parameters for XPOLAR1 pulse sequence (M)

**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:**
- `r11`: Set reference line in 1st indirectly detected dimension (M)
- `scalesw`: Scale spectral width in directly detected dimension (P)
- `scalesw1`: Set f₁ 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 `r12` macro.

Values: 'n', number greater than 0.0

See also: *User Guide: Solid-State NMR*

**Related:**
- `r12`: Set reference line in 2nd indirectly detected dimension (M)
- `scalesw`: Set scaling factor for multipulse experiments (M)
- `scalesw1`: Set f₁ scaling factor for 2D multipulse experiments (M)

**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: *VnmrJ Liquids NMR*

**Related:**
- `dof`: Frequency offset for first decoupler (P)
- `dn`: Nucleus of first decoupler (P)
- `sd2`: Set second decoupler frequency to cursor position (M)
sd2

**Set second decoupler frequency to cursor position (M)**

**Applicability:** Systems with a second decoupler.

**Description:** Sets the second decoupler 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:** *VnmrJ Liquids NMR*

**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)

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:** *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**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)

sd2a

**Set second decoupler frequency array (M)**

**Applicability:** Systems with a second decoupler.

**Description:** Sets up an array of offset values for the second decoupler, using `sd2` for the first position and `sd2a` for subsequent positions. This works only if the transmitter nucleus and second decoupler nucleus are the same (`tn=dn2`).

**See also:** *VnmrJ Liquids NMR*

**Related:**
- `dn2` Nucleus for second decoupler (P)
- `sd2` Set second decoupler frequency to cursor position (M)
**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:** *VnmrJ Liquids NMR*  
**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 m²/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:** *VnmrJ Liquids NMR*  
**Related:**  
- `dosy`  
  - Process DOSY experiments (M)

**sediff**  
*Set up spin-echo diffusion imaging sequence (M)*  
**Applicability:** Systems with imaging capabilities.  
**Description:** Sets up a standard spin-echo diffusion weighted experiment using the new user interface for imaging.  
**See also:** *VnmrJ Imaging NMR*

**sel1d**  
*Execute protocol actions of apptype sel1d (M)*  
**Description:** This macro is used to execute the protocol actions of the `sel1d` apptype.  
**Examples:**  
- `sel1d('setup')` — execute sel1d experimental setup  
- `sel1d('process')` — execute sel1d processing  
- `sel1d('plot')` — execute sel1d plotting

**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: VnmrJ Liquids NMR, 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: VnmrJ Liquids NMR

Related:

- Pbox: Pulse shaping software (U)
selexcit  Set up PFG selective excitation pulse sequence (M)
  Applicability: Systems with a pulsed field gradient module. Not available on MERCURYplus/Vx systems.
  Description: Prepares an experiment for PFG (pulsed field gradient) selective excitation, with presaturation option.
  See also: VnmrJ Liquids NMR

sems  Set up basic imaging sequence with oblique capability (M)
  Applicability: Systems with imaging capabilities.
  Description: Sets up a standard multislice spin-echo imaging sequence with oblique imaging capability.
  See also: VnmrJ Imaging NMR

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

seqcon  Acquisition loop control (P)
  Applicability: Systems with imaging capabilities.
  Description: Controls the status of various looping processes used during sequence acquisition. The `nD`, `seqcon`, `plist`, `patlist`, `pwrlist`, `fliplist`, and `sslist` parameters configure a particular parameter set for an application sequence defined by the value of the `seqfil` parameter.
  Values: String with five characters, consisting of the characters 'n', 's', and 'c', that control where and when the looping occurs:
  - 'n' (null loop) specifies a sequence that has no such loop function.
  - 's' (standard loop) sets the looping operation to occur during the execution of pulse sequence generation in the host computer. Each loop execution generates a new acode set for execution in the acquisition computer. Each acode set will ultimately give rise to its own data block in the FID file. A standard loop operation therefore lies outside the signal averaging (transient counter loop). Parameter arrays and use of the 2D implicit loop are standard loops. The multiecho loop cannot be a standard loop.
  - 'c' (compressed loop) sets the looping operation to occur dynamically in the acquisition computer, and each loop execution generates a new data "trace" within the current data "block". This requires space in the on-board HAL memory. Compressed loops lie inside the signal averaging loop.

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)
Each character position has place value and thus affects a different looping operation:

- First character: multiecho looping.
- Second character: multislice looping.
- Third character: 2D phase encode loop.
- Fourth character: 3D phase encode loop.
- Fifth character: 4D phase encode loop.

For example, `seqcon='ncsnn'` is 2D imaging with compressed multislice.

See also: *VnmrJ Imaging NMR*

**Related:**
- `fliplist` Standard flip angle list (P)
- `nD` Application dimension (P)
- `patlist` Active pulse template parameter list (P)
- `plist` Active pulse length parameter list (P)
- `pwrlist` Active pulse power level parameter list (P)
- `seqfil` Acquisition object code name (P)
- `sslist` Conjugate gradient list (P)

**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: *VnmrJ Liquids NMR*

**Related:**
- `pslabel` Pulse sequence label (P)

**seqgen**

**Initiate compilation of user’s pulse sequence (M,U)**

**Syntax:**
- (From VnmrJ) `seqgen([-static,]file<.c>)`
- (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 `psgen` 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

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, rf1, and rfp1 decisions based on tn=dn condition.

Arguments: F2_dig_res is the f2 digital resolution desired, in Hz/pt. Default is 6.
F1_dig_res is the f1 digital resolution desired, in Hz/pt. Default is 12.

Related: rf1 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 f2 digital resolution desired, in Hz/pt.
F1_dig_res is the f1 digital resolution desired, in Hz/pt.

Examples: set2d('cosyps')
set2d('hetcor',16)
set2d('het2dj',16,(2*sw1)/fn1)

See also: VnmrJ Liquids NMR

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('nocoef','curexp/info3d')`

See also: `VnmrJ Liquids NMR`

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: `VnmrJ Liquids NMR`

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)

### setarray

**Set up a parameter array (M)**

Applicability: Systems with imaging capabilities.

Syntax: `setarray<(name,start,step,elements)>`

Description: Sets up an array of a numeric acquisition parameter in single-arrayed experiments.

Arguments: `name` is the name of the parameter to be arrayed. The default (not entering any arguments) is the system prompts for the argument values.

- `start` is the starting value for the array.
- `step` is the step value for the array.
elements is the number of elements in the array.

Examples: setarray
setarray('d1',1,1,10)

See also: VnmrJ Imaging NMR

**setcenter**  
**Set up parameters for center sequence calibration (M)**

**Applicability:** Systems with imaging capabilities.

**Description:** Loads parameter sets for center sequence calibration during imaging installation.

See also: VnmrJ Imaging NMR

**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
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: VnmrJ Liquids NMR
Related: addi Start interactive add/subtract mode (C)
color Select plotting colors from a graphical interface (M)

setdecpars Set decoupler parameter values from probe file (M)
Syntax: setdecpars
Description: Reads from the probe file pwxlvl, pwx, pplvl, pp, dpwr, dmf, dmm, dres,
and dseq values, if they exist, and updates the current experiment parameters.
Related: setdec2pars Set decoupler 2 parameter values from probe file (M)

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)

setDefaultSize Set FOV to default size (C)
Applicability: Systems with imaging capabilities.
Syntax: setDefaultSize(float size)
Description: Sets default size (FOV) to size. All dimensions are set with the same size.
See also: VnmrJ Imaging NMR
Related: gplan Start interactive image planning (C)

setDefaultSlices Set default number of slices (C)
Applicability: Systems with imaging capabilities.
Syntax: setDefaultSlices(ns)
Description: Sets default number of slices to ns.
setDefaultThk  Set default slice thickness (C)
Applicability: Systems with imaging capabilities.
Syntax: setDefaultThk(float thk)
Description: Sets default thickness of slices to thk.
See also: VnmrJ Imaging NMR
Related: gplan  Start interactive image planning (C)

setDefaultType  Set default type (C)
Applicability: Systems with imaging capabilities.
Syntax: setDefaultType(type)
Description: Sets the default type to type.
See also: VnmrJ Imaging NMR
Related: gplan  Start interactive image planning (C)

setDisplayStyle  Show stripes or lines (C)
Applicability: Systems with imaging capabilities.
Syntax: setDisplayStyle(mode)
Description: Sets the display style to stripes or lines.
Arguments: If mode=0, shows stripes. If mode>0, shows lines. If mode=-1, toggles between the two styles.
See also: VnmrJ Imaging NMR
Related: gplan  Start interactive image planning (C)

setDrawIntersection  Show/hide intersection (C)
Applicability: Systems with imaging capabilities.
Syntax: setDrawIntersection(mode)
Description: mode=0, does not show intersection. mode>0, shows intersection. mode=-1, toggles between the two modes.
See also: VnmrJ Imaging NMR
Related: gplan  Start interactive image planning (C)

setDraw3D  Show/hide 3D (C)
Applicability: Systems with imaging capabilities.
Syntax: setDraw3D(mode)
Description: Shows or hides drawings in 3D.
Arguments: mode=0, does not show 3D. mode>0, shows 3D. mode=-1, toggles between the two modes.
See also: VnmrJ Imaging NMR
Related: gplan  Start interactive image planning (C)
**setDrawAxes**  
Show/hide axes (C)

Applicability: Systems with imaging capabilities.

Syntax: `setDrawAxes (mode)`

Description: Shows or hides axes in drawings.

Arguments: `mode=0`, does not show. `mode>0`, shows. `mode=-1`, toggles.

See also: *VnmrJ Imaging NMR*

Related: `gplan`  
Start interactive image planning (C)

**setDrawOrders**  
Show/hide order of drawings (C)

Applicability: Systems with imaging capabilities.

Description: Shows or hides order of drawings.

Arguments: `mode=0`, does not show. `mode>0`, shows. `mode=-1`, toggles.

See also: *VnmrJ Imaging NMR*

Related: `gplan`  
Start interactive image planning (C)

**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 enumerated 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)

**setFillPolygon**  
**Show/hide filled polygon (C)**

Applicability: Systems with imaging capabilities.

Syntax: setFillPolygon(mode)

Description: Shows or hides a filled polygon.

Arguments: 
- **mode=0**, does not show.
- **mode>0**, shows.
- **mode=-1**, toggles.

See also: *VnmrJ Imaging NMR*

Related: *gplan*  
Start interactive image planning (C)

**setflip**  
**Set rf power levels to desired flip angle (M)**

Applicability: Systems with imaging capabilities.

Syntax: setflip(name, patname, pwrname, flip)

Description: Sets up the rf power levels for a given pulse to obtain a desired flip angle. Power levels are calculated from the calibration data for a square pulse. The calibration data should be located in the file pulsecal, which should reside in the vnmrsys directory. The macro setflip also looks for the pulsecal file in the system directory.

Arguments: 
- **name** is the name of the pulse parameter.
- **patname** is the name of the pattern parameter.
- **pwrname** is the name of the power parameter.
- **flip** is the flip angle, in degrees.

Examples: setflip('pw', 'pwpat', 'tpwr', 90)

See also: *VnmrJ Imaging NMR*

Related: *pulsecal*  
Update and display pulse calibration data file (M)

**setfrq**  
**Set frequency of rf channels (C)**

Syntax: setfrq<(channel)><('nucleus')>

Description: Calculates frequencies based on the nucleus (tn, dn, dn2, etc.), referencing (lockfrq), 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: VnmrJ Liquids NMR

Related: spcfrq Display frequencies of rf channels (M)

---

**setGapMode** Fix/Unfix slice gap (C)

Applicability: Systems with imaging capabilities.

Syntax: `setGapMode(mode)`

Description: Fixes or unfixes gap between slices.

Arguments: mode=0, gap is not fixed. mode>0, gap is fixed.

See also: VnmrJ Imaging NMR

Related: gplan Start interactive image planning (C)

---

**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: VnmrJ Liquids NMR

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 $gro$, or $gzlvl1$ 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)
- $gro$: Readout gradient strength in DAC units (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 system global parameter $sysgcoil$ to the named table and updates the assignment values for the hardware-specific gradient calibration parameters $gcoil$, $gxcal$, $gycal$, $gzcal$, $griserate$, and $boresize$ to their corresponding values, described 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:**
- $boresize$: Magnet bore size (P)
- $config$: Display current configuration and possible change it (M)
- $gcoil$: Read data from gradient calibration tables (P)
- $griserate$: Gradient rise rate (P)
- $gxcal$, $gycal$, $gzcal$: Gradient strength for X, Y, Z gradients (P)
- $sysgcoil$: System value for $gcoil$ parameter (P)

### setgpe

**Set phase encode gradient levels (M)**

**Applicability:** Systems with imaging capabilities.

**Description:** Provides for selection of the phase encode gradient step size levels ($gpe$, $gpe2$, $gpe3$) and gradient pulse timing ($tpe$, $tpe2$, $tpe3$) from the FOV parameters ($lpe$, $lpe2$, $lpe3$).

The program requires no inputs and automatically calculates the values of $gpe$ and $tpe$ (2D, 3D, 4D), $gpe2$ and $tpe2$ (3D and 4D), and $gpe3$ and $tpe3$ (4D) from the corresponding FOV parameters and requested acquisition matrix sizes ($nv1$, $nv2$, $nv3$). Defaults are supplied for 2D, 3D, and 4D matrix sizes if these have not been set by the user.

The result of the $setgpe$ calculations results in setting the phase encode gradient levels so as to give the shortest possible phase encode timing. This prepares the sequence to collect data at the minimum $te$. Sequence applications, however, are free to rescale the values of the gradient level and timing parameters to meet their own requirements. Rescaling requires that:

\[ gpe \times tpe = gpe' \times tpe' \]
The product of the gradient set size and phase encode pulse remain constant.

See also: *VnmrJ Imaging NMR*

Related:

- `gpe` Phase encoding gradient increment (P)
- `lpe` Field of view for phase encode axis (P)
- `tpe` Duration of phase encoding gradient pulse (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: *VnmrJ Liquids NMR*

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)

### setgro

**Set readout gradient (M)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** `setgro<('min'|level)>`

**Description:** Sets the readout gradient by adjusting the values of `gro`, `sw`, and `at`. If entered without arguments, `setgro` operates in the automatic mode and uses a novel algorithm to estimate the maximum usable readout gradient. The algorithm is designed to provide a compromise between chemical shift artifact and S/N ratio in the image.

**Arguments:**
- `'min'` is a keyword to operate `setgro` in the automatic mode, to use simple algorithms to estimate the maximum usable readout gradient, and to set `gro`, `sw`, and `at` based on the estimate. Typical usage would be when operating at the shortest practical echo time.
- `levels` is a real number that is interpreted as a gradient level in gauss/cm. Provided that the number is in the range 0 to `gmax`, `setgro` then calculates `sw` and sets `gro` and `at`.

**Examples:**
- `setgro`
- `setgro('min')`
- `setgro(1.0)`

See also: *VnmrJ Imaging NMR*

Related:

- `at` Acquisition time (P)
- `gmax` Maximum gradient strength (P)
- `gro` Readout gradient strength (P)
- `sw` Spectral width (P)
**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`: 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)

**setgss**

*Select slice or voxel selection gradient levels (M)*

Applicability: Systems with imaging capabilities.

Syntax: `setgss(<gradient_name>, <thickness_name>)`

Description: Sets slice or voxel selection gradient levels, given the gradient level parameter and the thickness parameter. `setgss` searches the configuration list `sslist` (conjugate gradients) for the desired gradient level name.

If the gradient name is found (possibly multiple times), `setgss` calculates the bandwidth, in Hz, “cut” by each corresponding rf template on the list (patlist), at the length pointed to by the list (plist), and for the flip angle on the list (fliplist). The minimum bandwidth is assumed to define the “thickness” of the “cut.” The gradient level is then calculated from the minimum bandwidth selected by the rf pulses.

If `setgss` fails to find the supplied gradient_name, it returns the message “All RF templates used with gradient_name are nonselective.”

Arguments:
- `gradient_name`: is the name of the gradient level parameter whose value is to be set. The default is the user is prompted for the parameter name.
- `thickness_name`: is the name of the thickness parameter from which to compute the gradient level. The default is the user is prompted for the parameter name.

Examples:
- `setgss`
- `setgss('gss', 'thk')`

See also: *VnmrJ Imaging NMR*

Related:
- `fliplist` Standard flip angle list (P)
- `patlist` Active pulse template parameter list (P)
- `sslist` Conjugate gradient list (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 9 applies only to MERCURYplus/Vx.

Syntax: (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: Sets acquisition system hardware values. `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, on the `UNITY/NOVA`, `MERCURYplus/Vx`.

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 pairs 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.
'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.

tune' is a keyword that identifies the next argument, mode, as the tune mode to perform probe tuning on MERCURY. Mode is 1 for high band, 2 for low band, and 3 for off.

'lockfreq' is a keyword that the next argument is the lock frequency.

lockfreq_value is the lockfreq value, in MHz, for the lock frequency.

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related: dli Display list of integrals (C)
**setlimit**  
**Set limits of a parameter in a tree (C)**

**Syntax:**
1. `setlimit(parameter,max,min,step_size<,tree>)`
2. `setlimit(parameter,index<,tree>)`

**Description:**
If syntax 1 is used, when a parameter value is changed, the new value is checked against the limits set by `max` and `min`. The new value must also be a multiple of `step_size + min` (e.g., `setlimit('r1',80,10,20)` allows the values 10, 30, 50, and 70). The value of the parameter can be further modified by a macro called `_parameter` if the proper protection bit is set (see the `setprotect` command).

If syntax 2 is used, the `max`, `min`, and `step_size` for a parameter are obtained from the `index`-th entry of a table set for the parameter by `parmax`, `parmin`, and `parstep` in `conpar`.

**Arguments:**
- `parameter` is the name of the parameter.
- `max` and `min` are the maximum and minimum limits on a parameter value.
- `step_size` is the size of the steps allowed for a parameter within the limits `max` and `min`.
- `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.
- `index` is an index into a lookup table. When a single `index` argument is given, the parameter's protection bits (see the `setprotect` command) are set so that the table lookup is turned on.

**Examples:**
- `setlimit('a',80,10,20)`
- `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)
- `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 `a_lock='s'` is required in the parameter sets where used.
Arguments: solvent is the solvent to be used.

See also: VnmrJ Liquids NMR

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)

**setloop**  Control arrayed and real-time looping (M)

Applicability: Systems with imaging capabilities.

Description: Set the values for nf and ni to control arrayed and real-time looping. 

Loop control in imaging experiments, such as multislice, multiecho, and phase encoding, is set through a series of parameters (ne, ns, nv, nv2, nv3) directly set by the user. Underlying these parameters are two lower level parameters, nf and ni, used during pulse sequence execution to determine the mode of data acquisition. setloop manages the values of nf and ni as required to be consistent with the experiment parameters ne, nv, etc.

Two modes of data acquisition are supported in VnmrJ: arrayed and compressed. The difference between the modes is mainly in the data flow timing between host and acquisition computers:

- Arrayed data acquisition involves continuous communications between host and acquisition computers as pulse sequence instructions are sent to the acquisition CPU and data is returned to the host Sun for each element in the arrayed experiment. All explicitly arrayed experiments (e.g., pw=10, 20, 30) run in this manner. 2D experiments, including most high-resolution liquids and many imaging experiments, also run as “implicit” arrays, with the array size set by the parameter ni. Although communications between acquisition and host computers are quite fast, a small delay (typically a few milliseconds) is required to accommodate the communications and reinitialization between array elements. Certain fast imaging experiments, such as turboflash, Echo Planar Imaging (EPI), or even conventional multislice, often require loop timing similar to this interelement delay. These experiments use a second mode of data acquisition: the compressed mode.

- In compressed data acquisition, a single pulse sequence instruction set is sent to the acquisition computer, which then manages the entire experiment through real-time loops and pulse sequence elements. All data accumulated in the real-time loops is retained in the acquisition data memory until the experiment or array element is complete, at which time the data is sent back to the host. No timing overhead is associated with a real-time loop, and extremely short timing intervals may therefore be achieved with the compressed mode. Compressed data acquisition is controlled by the parameter nf, which requires that the number of points acquired must be nf*np. Experiments may be run completely in arrayed acquisition mode, or completely in compressed acquisition mode, or in a combination of the two.
setloop uses the `seqcon` parameter to determine which acquisition loops, if present, are arrayed and which are compressed. It then computes nf as the product of all compressed loop counts, and sets ni appropriately as either nv in the case of uncompressed phase-encode, or zero in the case of compressed phase-encode.

Each of the parameters `ne`, `ns`, `nv`, `nv2`, and `nv3` have corresponding underscore macros that execute setloop. Therefore, setloop is a lower level “management” macro that is run automatically each time one of these parameters is entered, and will not normally be run explicitly by the user. The comprehensive setup macro `imprep` also performs the setloop function. If imprep has been executed, there is no need to run setloop.

See also: *VnmrJ Imaging NMR*

Related:
- `d0` Overhead delay between FIDs (P)
- `flashc` Convert compressed 2D data to standard 2D format (C)
- `ne` Number of echoes to be acquired (P)
- `nf` Number of FIDs (P)
- `ns` Number of slices to be acquired (P)
- `nv` Number of 2D phase encode steps to be acquired (P)
- `seqcon` Acquisition loop control (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: *VnmrJ Liquids NMR*

Related: `ni` Number of increments in 1st indirectly detected dimension (P)

`setMarkMode` **Remove/activate mark (C)**

Applicability: Systems with imaging capabilities.

Syntax: `setMarkMode(mode)`

Description: Removes or activates one mark each time it is executed. If all marks (maximum number is 3) are removed or activated, it does nothing. When a mark is activated (mode>0), cursor changes into a pencil. Clicking a graphic area places a mark on the area, then the cursor turns back into an arrow.

mode=0, removes mark. mode>0, activates mark.

See also: *VnmrJ Imaging NMR*

Related: `gplan` Start interactive image planning (C)

`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)

**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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`.

max_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: *VnmrJ Liquids NMR*

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 parameters 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:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Cannot array the parameter</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Cannot change active/not active status</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Cannot change the parameter value</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>Causes _parameter macro to be executed (e.g., if parameter is named sw, macro _sw is executed when sw is changed)</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>Avoids automatic redisplay</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>Cannot delete parameter</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>System ID for spectrometer or data station</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>Cannot copy parameter from tree to tree</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>Will not set array parameter</td>
</tr>
<tr>
<td>9</td>
<td>512</td>
<td>Cannot set parameter enumeral values</td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
<td>Cannot change the parameter's group</td>
</tr>
<tr>
<td>11</td>
<td>2048</td>
<td>Cannot change protection bits</td>
</tr>
<tr>
<td>12</td>
<td>4096</td>
<td>Cannot change the display group</td>
</tr>
<tr>
<td>13</td>
<td>8192</td>
<td>Look up minimum, maximum, step values in table</td>
</tr>
<tr>
<td>14</td>
<td>16384</td>
<td>Parameter marked for locking (P_LOCK; see rtx)</td>
</tr>
<tr>
<td>15</td>
<td>32768</td>
<td>Global parameter not shared in multiple VJ viewports</td>
</tr>
<tr>
<td>16</td>
<td>65536</td>
<td>Force automatic redisplay in VJ templates</td>
</tr>
</tbody>
</table>

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)
- 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)
**setref**  
Set frequency referencing (M)

Syntax:  
```plaintext
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` (²H 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.

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:  
```plaintext
setref
setref('C13'): $rfl, $rfp
```

See also: *VnmrJ Liquids NMR*

### Related:
- **reffrq**: Reference frequency of reference line (P)
- **refpos**: Position of reference frequency (P)
- **rf1**: 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)

**setref1**  
Set freq. referencing for 1st indirectly detected dimension (M)

Syntax:  
```plaintext
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.

Arguments: `nucleus` is the frequency-relevant nucleus in f₁.

Examples:  
```plaintext
setref1(tn)
setref1('C13')
```

See also: *VnmrJ Liquids NMR*

### Related:
- **reffrq₁**: Reference frequency of reference line in 1st indirect dimension (P)
- **refpos₁**: Position of reference frequency in 1st indirect dimension (P)
- **rf1**: Reference peak position (P)
- **rfl₁**: Reference peak position in 1st indirectly detected dimension (P)
setref2  Set freq. referencing for 2nd indirect detected dimension (M)

Syntax:  setref1(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.

Arguments: nucleus is the frequency-relevant nucleus in f2.

Examples:  setref2(tn)
            setref2('C13')

See also:  VnmrJ Liquids NMR

Related:  reffrq2  Reference frequency of reference line in 2nd indirect dimension (P)
          refl2  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)

setsccout  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:  VnmrJ Liquids NMR

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:  VnmrJ Liquids NMR

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:  
VnmrJ Liquids NMR

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:  
VnmrJ Liquids NMR

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:  
VnmrJ Liquids NMR

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 (UNITY/INOVA). Otherwise, setselinv selects a “rectangular” pulse.
Related: suselfrq  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')
ssettcldefault('default2d','HMQC8')
See also: User Programming
Related: seqfil  Pulse sequence name (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('H1')`
- `setup('C13','DMSO')`

See also: *VnmrJ Liquids NMR*

---

**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: *VnmrJ Liquids NMR*

**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)

---

**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` is the name of the parameter.
- `value` is the value to set to the parameter.
- `index` is the number of a single element in an arrayed parameter. The default is 1.
- `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:**
- `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)
**setValue**  
Set parameter values (C)

Applicability: Systems with imaging capabilities.

Syntax: `setValue(char* paramName, float value, int index)`

Description: Sets imaging parameter values.

See also: *VnmrJ Imaging NMR*

Related: `gplan`  
Start interactive image planning (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` 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 excitation, 1 for de-excitation, or 0.5 for refocusing.
- `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.

Examples:
- `setwave('eburp1')`
- `setwave('GARP 12000.0')`
- `setwave('esnob 600 -1248.2 1 90.0 n n 0.001')`

See also: *VnmrJ Liquids NMR*

Related: `Pbox`  
Pulse shaping software (U)

**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: *VnmrJ Liquids NMR*

Related:
- `curwin`  
Current window (P)
- `fontselect`  
Open FontSelect window (C)
- `jwin`  
Activate current 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: *VnmrJ Liquids NMR*  
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 $(2 \times ni)/sw1$, in seconds.  
See also: *VnmrJ Liquids NMR*  
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 $(2 \times ni2)/sw2$, in seconds.  
See also: *VnmrJ Liquids NMR*  
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: *VnmrJ Liquids NMR*  
Related:  
- dfreq: Transmitter frequency of first decoupler (P)  
- dfreq2: Transmitter frequency of second decoupler (P)  
- dfreq3: Transmitter frequency of third decoupler (P)  
- tn: Nucleus for observe transmitter (P)
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)
        pipat  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: VnmrJ Liquids NMR
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: *VnmrJ Liquids NMR, User Programming*

Related: shelli  Start an interactive UNIX shell (C)

**shelli**  \[\text{Start an interactive UNIX shell (C)}\]

**Syntax:**  \[
\text{shelli(command)}
\]

**Description:**  On a terminal, runs interactively the UNIX command line given as the argument. No return or output variables are allowed.

**Arguments:**  \[
\text{command} \quad \text{is a UNIX command line to be executed.}
\]

**Examples:**  \[
\text{shelli('vi myfile')}
\]

See also: *VnmrJ Liquids NMR, User Programming*

Related: shell  Start a UNIX shell (C)

**shim**  \[\text{Submit an Autoshim experiment to acquisition (C)}\]

**Description:**  Performs validity checks on the acquisition parameters and then submits an Autoshim experiment to acquisition.

See also: *VnmrJ Liquids NMR*

Related: au  Submit experiment to acquisition and process data (C)

type=\text{change}  Submit a change sample experiment to acquisition (M)

type=\text{ga}  Submit experiment to acquisition and FT the result (C)

type=\text{go}  Submit experiment to acquisition (C)

type=\text{lock}  Submit an Autolock experiment to acquisition (C)

type=\text{sample}  Submit change sample, autoshim experiment to acquisition (M)

type=\text{spin}  Submit a spin setup experiment to acquisition (C)

type=\text{su}  Submit a setup experiment to acquisition (M)

**shimset**  \[\text{Type of shim set (P)}\]

**Description:**  Configuration parameter for the type of shims on the system. The value of shimset is set using the Shimset label in the CONFIG window (opened from config).

**Values:**  \[
1 \text{ to } 14, \quad \text{where the value identifies one of the following shim sets:}
\]

1 is a shim set in a Varian 13-shim supply with computer-controlled axial shims \(z_1, z_1c, z_2, z_2c, z_3, z_4,\) and radial shims \(x_1, y_1, xz, yz, xy, x^2y_2, x_3, y_3.\) Shims can be adjusted from \(-2047\) to \(+2047.\) This value is used with the Ultra•nmr shim system when operated from the HIM box (Varian 13 Shims choice in CONFIG window).

2 is a shim set in a Oxford 18-shim supply with computer-controlled axial shims \(z_1, z_1c, z_2, z_2c, z_3, z_4, z_5,\) and radial shims \(x_1, y_1, xz, yz, xy, x^2y_2, x_3, y_3, xz_2, yz_2, yz_3, x^2y_2.\) Shims can be adjusted from \(-2047\) to \(+2047\) (Oxford 18 Shims choice in CONFIG window).

3 is a shim set in a Varian 23-shim supply with computer-controlled axial shims \(z_1, z_2, z_3, z_4, z_5, z_6,\) and radial shims \(x_1, y_1, xz, yz, xy, x^2y_2, x_3, y_3, xz_2, yz_2, zxy, x^2y_2.\) Shims can be adjusted from \(-32767\) to \(+32767\) (Varian 23 Shims choice in CONFIG window).

4 is a shim set in a Varian 28-shim supply with computer-controlled axial shims \(z_1, z_2, z_3, z_4, z_5, z_6, z_7,\) and radial shims \(x_1, y_1, xz, yz, x^2y_2, x_3, y_3, xz_2, yz_2, zxy, x^2y_2.\) Shims can be adjusted from \(-32767\) to \(+32767\) (Varian 23 Shims choice in CONFIG window).
adjusted from –32767 to +32767 (Varian 28 Shims choice in CONFIG 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, xz2y2, yz2x2, z3x, z3y, z4x, z4y, z3x2y2, z3xy, z2x3, z2y3, z3x3, z3y3, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Ultra Shims choice in CONFIG window).

6 is a shim set in a Varian 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, xz2y2. Shims can be adjusted from –32767 to +32767 (Varian 18 Shims choice in CONFIG window).

7 is a shim set in a Varian 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, xz2y2, z3x, z3y, z2x2y2, z2xy, z3x2y2, z3xy, z3y2, z3x2y2, z3xy, z2x3, z2y3, z3x3, z3y3, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Varian 20 Shims choice in CONFIG window).

8 is a shim set in an Oxford 15-shim supply with computer-controlled axial shims z1, z2, z3, z4, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y2, yz, xz2y2. Shims can be adjusted from –2047 to +2047 (Oxford 15 Shims choice in CONFIG window).

9 is a shim set in a Varian 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, xz2y2, z3x, z3y, z2x2y2, z2xy, z3x2y2, z3xy, z3y2, z3x2y2, z3xy, z2x3, z2y3, z3x3, z3y3, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Varian 40 Shims choice in CONFIG window).

10 is a shim set in a Varian 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 (Varian 14 Shims choice in CONFIG window).

11 is a shim set in a Varian 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 CONFIG window).

12 is a shim set in a Varian 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, xz2y2, z3x, z3y, z2x2y2, z2xy, z3x2y2, z3xy, z3y2, z3x2y2, z3xy, z2x3, z2y3, z3x3, z3y3, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Varian 26 Shims choice in CONFIG window).

13 is a shim set in an Varian 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, xz2y2, z3x, z3y, z2x2y2, z2xy, z3x2y2, z3xy, z3y2, z3x2y2, z3xy, z2x3, z2y3, z3x3, z3y3, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Varian 29 Shims choice in CONFIG window).

14 is a shim set in a Varian 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, xz2y2, z3x, z3y, z2x2y2, z2xy, x3, zy3, x4y, z4y, z3x2y2, z3xy, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Varian 35 Shims choice in CONFIG window).

15 is the Varian 15 Shim.

16 is the Ultra 18 Shims.

See also: VnmrJ Installation and Administration

Related: config Display current configuration and possibly change it (M
shimpath  Path to user's shims directory (P)
Description: Contains an absolute path to a user's shims directory, which has files of shim settings. If shimpath exists for a user, it must be defined in the user's global parameter file. To create shimpath, enter:
create('shimpath','string','global').
See also: VnmrJ Liquids NMR
Related: rts Retrieve shim coil settings (C)
         svs Save shim coil settings (C)

showconsole Show UNITY/INOVA console configuration parameters (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: VnmrJ Liquids NMR
Related: ihwinfo Hardware status of UNITY/INOVA console (C)

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: VnmrJ Liquids NMR
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.

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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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.
**showprintq**  
**Display print jobs in print queue (M)**  
Description: Displays current print jobs in the print queue for the active printer.  
See also: VnmrJ Liquids NMR  
Related: killprint Stop print jobs and remove from print queue (C)  
Related: showplotq Display plot jobs in plot queue (M)

**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: VnmrJ Liquids NMR  
Related: Acqstat Bring up the acquisition status display (U)

**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: acos Find arc cosine of number (C)  
arccos Calculate arc cosine of real number (M)  
arcsin Calculate arc sine of real number (M)  
arctan Calculate arc tangent of real number (M)  
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 \( \text{shift} \) is greater than 1, the \( \text{sbs} \) parameter is calculated as \( 2 \times \frac{\text{sb}}{\text{shift}} \) (\( \text{sbs1} \) is calculated as \( 2 \times \frac{\text{sb1}}{\text{shift}} \)). \( \text{sine} \) (2) gives a “\( \pi /2 \)-shifted” sine window, i.e., cosine weighting. \( \text{sine} \) (3) gives a “\( \pi /3 \)” shifted sine window, etc. If \( \text{shift} \) is less than or equal to 1, an unshifted sine window is used (\( \text{sbs} = 'n' \) or \( \text{sbs1} = 'n' \)).

\( \text{number_points} \) specifies the number of real points that the window function spans. The value of the window function for subsequent points is 0. \( \text{number_points} \) must be greater than 0 and a multiple of 2. The default is \( \text{ni} \times 2 \) if \( \text{trace} = 'f1' \), or \( \text{np} \) if \( \text{trace} = 'f2' \).

domain is ‘f1’ or ‘f2’. The default is the current setting of \( \text{trace} \).

See also: \( \text{VnmrJ Liquids NMR} \)

Related:
- \( \text{np} \): Number of data points (P)
- \( \text{sb} \): Sinebell const. in directly detected dimension (P)
- \( \text{sb1} \): Sinebell const. in 1st indirectly detected dimension (P)
- \( \text{sbs} \): Sinebell shift const. in directly detected dimension (P)
- \( \text{sbs1} \): Sinebell shift const. in 1st indirectly detected dimension (P)
- \( \text{sinesq} \): Find values for a sine squared window function (M)
- \( \text{trace} \): Mode for \( n \)-dimensional data display (P)

\( \text{sinebell} \)  
Select default parameters for sinebell weighting (M)

Description: Generates initial guess at good sinebell weighting parameters by setting the \( \text{sb} \) and \( \text{sb1} \) parameters to one-half the acquisition time and turning off all other weighting. Use \( \text{sinebell} \) in absolute-value 2D experiments only.

See also: \( \text{VnmrJ Liquids NMR} \)

Related:
- \( \text{pseudo} \): Set default parameters for pseudo-echo weighting (M)
- \( \text{sb} \): Sinebell const. in directly detected dimension (P)
- \( \text{sb1} \): Sinebell const. in 1st indirectly detected dimension (P)

\( \text{sinesq} \)  
Find values for a sine-squared window function (M)

Syntax:

\[
\text{sinesq}(\text{shift},\text{number_points},\text{domain})
\]

Description: Calculates appropriate values for parameters \( \text{sb} \) and \( \text{sbs} \) (if the domain argument is ‘f2’) or for parameters \( \text{sb1} \) and \( \text{sbs1} \) (if the domain argument is ‘f1’) in order to achieve a sine-squared window function. The value of parameter \( \text{trace} \) is used if the domain argument is not entered.

Arguments: \( \text{shift} \) sets the starting value for the window function. If \( \text{shift} \) is greater than 0, the starting value is given by \( \sin \left(\frac{\pi}{\text{shift}}\right) \); otherwise, if \( \text{shift} \) is less than or equal to 0, the starting value is 0. The default value is 0.

\( \text{number_points} \) specifies the number of real points that the window function spans. The value of the window function for subsequent points is 0. The \( \text{number_points} \) argument must be greater than 0 and a multiple of 2. The default is \( \text{ni} \times 2 \) if \( \text{trace} = 'f1' \), or \( \text{np} \) if \( \text{trace} = 'f2' \).

domain is ‘f1’ or ‘f2’. The default is the current setting of \( \text{trace} \).

See also: \( \text{VnmrJ Liquids NMR} \)

Related:
- \( \text{ni} \): Number of increments in 1st indirectly detected dimension (P)
- \( \text{np} \): Number of data points (P)
- \( \text{sb} \): Sinebell const. in directly detected dimension (P)
- \( \text{sb1} \): Sinebell const. in 1st indirectly detected dimension (P)
- \( \text{sbs} \): Sinebell shift const. in directly detected dimension (P)
- \( \text{sine} \): Find values for a sine window function (M)
- \( \text{trace} \): 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:**
```
ri = 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:** *VnmrJ Liquids NMR*

**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)

---

**sliceorder**

**Reorder the slice position list (M)**

**Applicability:** Systems with the imaging capabilities.

**Syntax:** `sliceorder<('a'|'d'|'i')>`

**Description:** Reorders the slice position list, `pss`, in ascending, descending, or alternating odd/even order.

Alternating order is often used for multislice excitation to separate physically adjacent slices in time to reduce saturation effects. For example, if `pss=-3,-2,-1,0,1,2,3` is reordered by alternating odd/even order, the result is `pss=-3,-1,1,3,-2,0,2` so that the adjacent slices `-1` and `-2`, for example, are separated by three time intervals instead of just one.

**Arguments:**
- `'a'` is a keyword to reorder the list in alternating odd/even order. This is the default.
- `'d'` is a keyword to reorder the list in descending order.
- `'i'` is a keyword to reorder the list in ascending order.

**Examples:**
```
sliceorder('d')
```

**See also:** *VnmrJ Imaging NMR*

**Related:**
- `pss` Slice position (P)

---

**sliceplan**

**Set slice parameters for target slice (M)**

**Applicability:** Systems with imaging capabilities.

**Description:** Calculates and sets the slice parameters for the target slice defined in the file `curexp+ '/mark2d.out'`. The slice parameters (i.e., `pss, psi, phi, theta`) are calculated and set by using `sliceplan`. The Calculate Target button of the slice planner menu also uses `sliceplan`. See the `plan` macro for further details.
s

See also: *VnmrJ Imaging NMR*

Related: `curexp` Current experiment directory (P)
`drawslice` Display target slices (M)
`drawvox` Display target voxels (M)
`plan` Display menu for planning a target scan (M)
`voxplan` Set voxel parameters for voxel defined by 2D box cursor (M)

**slp**

Family of offset Frequencies of SLP shapes (P)

Applicability: Systems with LC-NMR or VAST accessory.

Syntax: `slp(frequency offset from the trans transmitter)`

Description: Specifies frequencies, in Hz, of Shifted Laminar Pulses (SLP) shapes used for suppression of solvent peaks. There are 6 members of the slp family, slp0 (solvent 1), slp1 (solvent 2), slp2 (solvent 3), slp3 (solvent 4), slp4 (solvent 5), slp6 (solvent 6), and slp6 (solvent 7). There is no slp1 parameter.

**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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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)**

**Applicability:** *UNITY*/*NOVA* systems only.

**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 CONFIG window (opened from **config**).

**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; VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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 ¹H or ¹³C, 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 ²H 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 (D2O)
- Cyclohexane
- Acetone (C6D12)
- Toluene (CD3OD)
- Benzene (C6D5CH3)
- C6D6
- Acetic_Acid
- DMSO (CD3COOD)

See also: 

- VnmrJ Liquids NMR

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: 

- VnmrJ Liquids NMR

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 sort 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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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 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: *VnmrJ Liquids NMR*

**Related:**
- **add**: Add current FID to add/subtract experiment (C)
- **addi**: Start interactive add/subtract mode (C)
- **ciradd**: 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)

---

**spcfreq**

**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, `spcfreq` set the frequency of the observe channel. The parameter `sfrq` now sets the frequency instead of `spcfreq`.

See also: *VnmrJ Liquids NMR*

**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 dc 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 f3 dimension (`sw`, `np`, `fn`), the second character refers to the f1 dimension (`sw1`, `ni`, `fn1`), and the third character refers to the f2 dimension (`sw2`, `ni2`, `fn2`). The default is `'nnn'`.

See also: *VnmrJ Liquids NMR*

**Related:**
- **dc**: Calculate spectral drift correction (C)
- **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)
- **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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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)

```plaintext
spingen
spingen(pulsesquence)
spingen<option,pulsesquence,pulsesquence2>
spingen'-psg',pulsesquence
spingen'-all',pulsesquence
spingen'-dps',pulsesquence
```

(From UNIX)

```plaintext
spingen pulsesquence < pulsesquence2,,>
spingen <option> pulsesquence < pulsesquence2,, >
spingen -psg pulsesquence
```
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 user'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: VnmrJ Liquids NMR

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:** *VnmrJ Liquids NMR*

**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)

**spinopt**  
**Spin automation (P)**

**Applicability:** *MERCURYplus/Vx* systems.

**Description:** Specifies whether spin hardware is installed. The hardware is always present and `spinopt='y'` is the default.

**Values:** 'y' is the default.  
'n' disables spin hardware.
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 sifreq, 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 sifreq. 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:

```plaintext
spins
spins('calculate','energy')
spins('iterate')
```

See also: VnmrJ Liquids NMR

Related:

- `assign` Assign transitions to experimental lines (M)
- `clindex` Index of experimental frequency of a transition (P)
- `dga` Display parameter groups (spin simulation) (C)
- `dia` 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: VnmrJ Liquids NMR, UNITYINOVA Solids Hardware Installation

Related:

- `delta` Difference of two frequency cursors (P)

### 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 \texttt{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 \texttt{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 \texttt{spmin} spectrum will contain only baseline in these regions, eliminating the spinning sidebands.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:}
- \texttt{addi} Start interactive add/subtract mode (C)
- \texttt{spadd} Add current spectrum to add/subtract experiment (C)
- \texttt{spsub} Subtract current spectrum from add/subtract experiment (C)

\textbf{spsm}

\textbf{Enter spin system (M)}

\textbf{Syntax:} \texttt{spsm(spin\_system)}

\textbf{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.

\textbf{Arguments:} \texttt{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., \texttt{A=78.5 JAC=5.6}). Entry of chemical shifts in ppm is entered by using \texttt{sfrq} (e.g., \texttt{B=7.5*sfrq}).

\textbf{Examples:}
- \texttt{spsm('AB')}
- \texttt{spsm('A3B2')}
- \texttt{spsm('AB2CMXY')}

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:}
- \texttt{sfrq} Transmitter frequency of observe nucleus (P)
- \texttt{spins} Perform spin simulation calculation (C)

\textbf{spsub}

\textbf{Subtract current spectrum from add/subtract experiment (C)}

\textbf{Syntax:} (1) \texttt{spsub< (multiplier,,shift>)>}
(2) \texttt{spsub('new')}
(3) \texttt{spsub('trace',index)}

\textbf{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.

\textbf{Arguments:} \texttt{multiplier} is a value to multiply each spectrum being subtracted from the add/subtract experiment (exp5). The normal range of \texttt{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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related:
gaussian Set up unshifted Gaussian window function (M)
n1 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 fullpath directory where a study is stored. It is set when a new study is created.

See also: autodir(P), globalauto(P), studyid(P), sqname(P)

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. Note that
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'
slname='s_$loc$_' studyid='s_7_01'
slname='r$vrack$z$vzone$/well$loc$%R0%'
studyid='r1z3/well16'

See also: autodir(P), autoname(P), globalauto(P), sqdir(P), studyid(P)

**sqrt**

Return square root of a real number (O)

Description: In MAGICAL programming, an operator that returns the square root of a real
number. If the argument is negative, sqrt evaluates to 0.0.

Examples: \( a = \sqrt{b} \)

See also: User Programming

Related: acos Find arc cosine of number (C)
arccos Calculate arc cosine of real number (M)
arcsin Calculate arc sine of real number (M)
arctan Calculate arc tangent of real number (M)
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)

**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: VnmrJ Liquids NMR

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)
sgcosine 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 \texttt{hsrotor='y'}, the measured spinning speed is reported in \texttt{srate} for systems that have rotor synchronization.

Values: \(0 \text{ to } 10^7\), in Hz.

See also: \textit{User Guide: Solid-State NMR}

Related: \texttt{hsrotor} Display rotor speed for solids operation (P)

\texttt{xpolar1} Set up parameters for XPOLAR1 pulse sequence (M)

\texttt{sread} \hspace{1cm} \textit{Read converted data into VnmrJ (C)}

\textbf{Syntax}: \texttt{sread(file<,template>)}

\textbf{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 \texttt{convertbru} command before \texttt{sread} can read the data in the file into VnmrJ.

\textbf{Arguments}:

- \texttt{file} is the name of a file containing data converted using \texttt{convertbru}.
- \texttt{template} is the full path of a parameter template file, but without appending the \texttt{.par} extension on the file name. The default is \texttt{bruker.par}. If no parameter template is specified and \texttt{bruker.par} cannot be found in the user or system \texttt{parlib} directory, \texttt{sread} aborts with an error message.

\textbf{Examples}: \texttt{sread('brudata.cv','/vnmr/parlib/bruker')}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{convertbru} Convert Bruker data (M,U)

\texttt{ss} \hspace{1cm} \textit{Steady-state transients (P)}

\textbf{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 \textit{dummy scans}). If \texttt{ss} is positive, \texttt{ss} steady-state transients are applied on the first increment only, and if \texttt{ss} is negative, \(-\texttt{ss}\) steady-state transients are applied at the start of each increment.

Values: \texttt{‘n’}, \(-32768\) to \(32767\)

See also: \textit{VnmrJ Liquids NMR}; \textit{User Programming}

\texttt{ssecho} \hspace{1cm} \textit{Set up solid-state echo pulse sequence (M)}

\textbf{Applicability}: Systems with a solids module. Not supplied with \textit{MERCURYplus/Vx}.

\textbf{Syntax}: \texttt{ssecho}

\textbf{Description}: Converts a standard two-pulse experiment to a ready-to-run solid-state NMR echo (SSECHO) pulse sequence.

See also: \textit{User Guide: Solid-State NMR}

\texttt{ssecho1} \hspace{1cm} \textit{Set up parameters for SSECHO1 pulse sequence (M)}

\textbf{Applicability}: \textit{UNITY/INOVa} system with a wideline solids module. Not supplied with \textit{MERCURYplus/Vx}.

\textbf{Description}: Sets up a parameter set for the quadrupole echo pulse sequence SSECHO1.

See also: \textit{User Guide: Solid-State NMR}
**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:** VnmrJ Liquids NMR

**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 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:** VnmrJ Liquids NMR

**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.
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, sslsfreq, 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: VnmrJ Liquids NMR

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)
        sslsfreq 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)

ssplan

Set slice parameters for target slice (M)

Applicability: Systems with imaging capabilities.

Description: Used by the Calculate Target button of the slice planner menu to calculate and set the slice parameters pss, psi, phi, and theta. ssplan creates the string parameter planlock and assigns it the value 'ssplan'. This prevents a user inadvertently performing a second planning operation without applying the reset command to restore the original parameters for the scout data.

See also: VnmrJ Imaging NMR

Related: drawslice Display target slices (M)
        plan Display menu for planning a target scan (M)
        phi Euler angle phi from magnet frame (P)
        psi Euler angle psi from magnet frame (P)
        pss Slice position (P)
        theta Euler angle theta from magnet frame (P)

sslst

Conjugate gradient list (P)

Applicability: Systems with imaging capabilities.
Description: Sets an array of strings that defines the names of gradient parameters used for slice or voxel selection. If the pulse performs no slice selection operation, the user may enter ' ' or 'n' for the value of sslist (e.g., sslist='n', 'gss', 'gss'). The nD, seqcon, plist, patlist, pwrlist, fliplist, and sslist parameters configure a particular parameter set for an application sequence defined by the value of the seqfil parameter. The plist, patlist, pwrlist, fliplist, and sslist parameters provide information concerning the rf pulse and conjugate gradients used by the sequence.

See also: VnmrJ Imaging NMR

Related:
- fliplist Standard flip angle list (P)
- nD Application dimension (P)
- patlist Active pulse template parameter list (P)
- plist Active pulse length parameter list (P)
- pwrlist Active pulse power level parameter list (P)
- seqcon Acquisition loop control (P)
- seqfil Application object code name (P)

ssprep Calculate slice gradient and slice selection parameters (M)

Applicability: Systems with echo planar imaging (EPI) capabilities.

Description: Calculates the slice gradient parameter, gss, and the slice selection parameters, tpwr1 and tpwr2, for use in the EPI experiment. Unlike imprep, readout and phase encode related parameters are not modified by ssprep.

See also: VnmrJ Imaging NMR

Related:
- gss Slice selection gradient strength (P)
- imprep Calculate gradient and rf parameters for imaging (M)
- tpwr1 Intensity of an excitation pulse (P)
- tpwr2 Intensity of an inversion pulse (P)

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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)

startIplan  **Start/restart image planning (C)**

Applicability: Systems with imaging capabilities.

Syntax: `startIplan<(type)>`

Description: Starts/restarts image planning with the Active prescription, and sets the default type as `type`.

Values: `type` is an integer. If `type` is not given, `type=0`; if `type=-1`, `type` is determined by current parameters.

See also: VnmrJ Imaging User Guide: Image Processing

Related: gplan  Start interactive image planning (C)

startMovie  **Start running a movie (C)**

Applicability: Systems with imaging capabilities.

Description: Start running the current movie. It is run in the first selected frame, or if none is selected, in the first frame.

See also: VnmrJ Imaging User Guide: Image Processing

Related: stopMovie  (C)

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`
S

See also: *VnmrJ Walkup NMR: 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**  
**Execute protocol actions of apptype std1d (M)**

Applicability: Liquids systems.

Description: This macro is used to execute the protocol actions of the std1d apptype.

Examples:  
- `std1d('setup')` – execute std1d experimental setup  
- `std1d('process')` – execute std1d processing  
- `std1d('plot')` – execute std1d plotting

**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: *VnmrJ Liquids NMR*

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)

**steam**  
**Set up volume localized spectroscopy sequence (M)**

Applicability: Systems with optional Imaging Pulse Sequences installed.

Description: Sets up a sequence for volume localized spectroscopy that uses the stimulated echo technique.

See also: *VnmrJ Imaging NMR*

**stepMovie**  
**Step one frame in a movie (C)**

Syntax: `stepMovie('+')` | `stepMovie('-')`
Description: Shows either the next frame (with the '+' argument) or the previous frame (with the '-' argument) of the current movie.

Examples: `stepMovie('+')`

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `startMovie`

### 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: *VnmrJ Liquids NMR*

Related: `spins` Perform spin simulation calculation (C)

Related: `spsm` Enter spin system (M)

### stopMovie

**Stop running a movie (C)**

Description: Stops the current movie, if it is running.

See also: *VnmrJ Imaging User Guide: Image Processing*

Related: `startMovie`

### 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*

### 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: *VnmrJ Liquids NMR*

Related: `addpar` Add selected parameters to the current experiment (M)

Related: `lpalg` LP algorithm in `np` dimension (P)

Related: `np` Number of data points (P)

Related: `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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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 strtlp. Enter addpar('lp',1) to create strtlp1 and other ni dimension LP parameters in the current experiment.

See also: VnmrJ Liquids NMR

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)
lpnuptsl LP number of data points in ni dimension (P)
strlp2  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. strlp2 functions analogously to strlp. Enter addpar('lp',2) to create strlp2 and other ni2 dimension LP parameters in the current experiment.

See also: VnmrJ Liquids NMR

Related: addpar, lpalg2, lpfilt2, lpnupts2, lpopt2, strtext2

studyid  Study identification (P)

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: autodir(P), globalauto(P), sqdir(P), sqname(P)

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.

On Unity/Inova systems, 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: VnmrJ Liquids NMR

Related: acqi, au, change, ga, go, load, lock, lockgain, lockphase, lockpower, qtune, sample, shim
**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 (0.75)`
- `sub('new')`
- `sub('trace',2)`

See also: _VnmrJ Liquids NMR_

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)**

Syntax:
1. `substr(string,word_number):substring`
2. `substr(string,index,length):substring`

Description: Returns a substring from a string based on the number of a word in the string (syntax 1) or on the starting character and length of the substring (syntax 2).

Arguments:
- `string` is the string or a string variable.
- `word_number` is the number of the word to be selected. A word is defined here as any string of characters separated by spaces or tabs. For example, if `string` is 'There are 10 samples to run' and `word_number` is 4, the substring 'samples' is returned (see first example below).
- `substring` returns the substring from string.
- `index` is the character to start from, with the first character considered 1.
- `length` is the length of substring in characters or spaces. For example, if `string` is 'abcdefg', `index` is 2, and `length` is 3, the substring 'bcd' is returned (see second example below).
Examples:

```plaintext
substr('There are 10 samples to run',4): sa
substr('abcdefg',2,3): sa
```

See also: User Programming

Related:

<table>
<thead>
<tr>
<th>length</th>
<th>Determine length of a string (C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Create a string variable (C)</td>
</tr>
</tbody>
</table>

**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.

Related:

<table>
<thead>
<tr>
<th>NOESY1D</th>
<th>Change parameters for NOESY1D experiment (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>setselinv</td>
<td>Set up selective inversion (M)</td>
</tr>
<tr>
<td>setselfrqc</td>
<td>Select selective frequency and width (M)</td>
</tr>
<tr>
<td>TOCSY1D</td>
<td>Change parameters for TOCSY1D experiment (M)</td>
</tr>
</tbody>
</table>

**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*graysl*numgray)/64.0`
- `b = numgray*(0.5-(graysl*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 by the macros `svib` and `svsis` to save data files for ImageBrowser.

Examples:

```plaintext
svdat(rathead,'b')
```

See also: VnmrJ Imaging NMR

Related:

<table>
<thead>
<tr>
<th>browser</th>
<th>Start ImageBrowser (U)</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>Create new parameter in parameter tree (C)</td>
</tr>
<tr>
<td>fdfgluer</td>
<td>Make FDF file from header and data parts (C)</td>
</tr>
<tr>
<td>grayctr</td>
<td>Gray level window adjustment (P)</td>
</tr>
<tr>
<td>graysl</td>
<td>Gray level slope (contrast) adjustment (P)</td>
</tr>
<tr>
<td>svib</td>
<td>Generate and save images as ImageBrowser FDF files,(M)</td>
</tr>
<tr>
<td>svsis</td>
<td>Generate and save images as FDF files (M)</td>
</tr>
</tbody>
</table>
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 preventsvp 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: VnmrJ Liquids NMR

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 $2^{\text{scale}}$ 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)
svib Save image data in FDF format (M)
svfdir  Directory for non-study data (P)
Description: Specifies the directory where data is saved when not using a study in VnmrJ.
See also: fidsave(M), svfname(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 delimiters, dollar signs ($) and percent signs (%), as autoname.
Examples: If svfdir=userdir+'/data', the result from fidsave is:
svfname='$pslabel$_$tn$_' -> userdir+'/data/Proton_H1_01.fid'
svfname='%DATE%t%TIME%R0%' -> userdir+'/data/20040501/t113005.fid'

svib  Generate and save images as ImageBrowser FDF files (M)
Applicability: Systems with imaging capabilities.
Syntax: svib(directory<,'f'|'m'|'i'|'o'>)
Description: Generates images from the current experiment and saves them into the specified directory as FDF (Flexible Data Format) files. svib can save a single image, or a number of images in the case of multislice experiments. The resulting FDF image files are composed of two parts: a text header, followed by the binary image data.
svib uses a the command svdat 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. svdat dumps the data so that the (0,0) coordinates are the first data point in the file.
Note that modifications to svib should be made in the user’s maclib and that the output values of the direction cosines may not be correct.
Arguments: directory is the name of a directory that is made in the current working directory. The .dat extension is appended to the name. Image files are created in this directory as image0001.fdf, image0002.fdf, and so on. A procpar file is also saved into this directory.
'f', 'm', 'i', and 'o' are keywords that define the type of image data:
- 'f' outputs the data in floating point format. This is the default.
- 'm' or 'i' outputs the data as 12-bit integer values in 16-bit words.
- 'b' outputs the data in 8-bit integer bytes.
Examples: svib('rat.images')
See also: VnmrJ Imaging NMR
Related: dmi Display multiple images (M)
fdfgluer Make FDF file from header and data parts (U)
svdat  Save data (C)

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: VnmrJ Liquids NMR

Related:
rt Retrieve FID (M)
rtp Retrieve parameters (M)
svf Save FIDs in current experiment (M)

svphf Save current phasefile (C)

Applicability: Systems with imaging capabilities.

Syntax: svphf(file)

Description: Copies current experiment phasefile (curexp+ '/datdir/phasefile') to planes directory of current experiment (curexp+ '/planes/file', where file is the file name given in the argument). The current phasefile is the current processed data set after apodization, Fourier transformation, vertical scaling, and phasing or absolute-value calculation, but before the contrast windowing controlled by the grayctr and graysl parameters. No parameters of any kind are stored with the phasefile. svphf creates the planes directory if it does not already exist.

Arguments: file is the name to be given to the phasefile when copied to the planes directory. Use only a relative path for file, not an absolute path.

Examples: svphf ('elsa')

See also: VnmrJ Imaging NMR

Related:
curexp Current experiment directory (P)
grayctr Gray level window adjustment (P)
graysl Gray level slope (contrast) adjustment (P)
imcalc Calculate 2D phasefiles (M,U)
makephf Transform and save images as phasefiles (M)
rtphf Return stored phasefile to the current phasefile (C)

svs Save shim coil settings (C)

Syntax: svs(file)::status

Description: Saves all shim coil settings except Z0 to a file. If svs cannot store the shim file, it displays the directories it tried to use.

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 tries to go into up to three different directories, as follows:

- First, it looks for a shims subdirectory in your user directory. If that exists, the settings are stored there.
- Next, if the shims subdirectory does not exist, it then looks for the global parameter shimspath. If shimspath is present, it is expected to contain a directory name. If this directory exists and a new file entry can be created in the directory, the file is saved there.
Finally, if this does not work, the file is saved in the shims subdirectory of the system directory.

`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 subdirectory of the user directory.
- 2 indicates `svs` stored the file using the global parameter `shimspath`.
- 3 indicates `svs` stored the file in shims subdirectory of the system directory.

Examples:

```
svs('acetone')
svs('bb10mm'):r1
```

See also: *VnmrJ Liquids NMR*

**svs**

**Spin simulation vertical scale (P)**

Description: Vertical scale for simulated spectrum.

Values: 0 to 1e10. A typical value is 200.

See also: *VnmrJ Liquids NMR*

Related: 
- `rts` Retrieve shim coil settings (C)
- `shimspath` Path to user’s shims directory (P)

**svsis**

**Generate and save images as FDF files (M)**

Applicability: Systems with imaging capabilities.

Syntax: `svsis(directory<,'f'|'m'>)`

Description: Generates images from the current experiment and saves them into the specified directory as Flexible Data Format (FDF) files. `svsis` saves one image, or a number of images in the case of multislice experiments.

`svsis` only saves images from the standard SISCO imaging sequences: image, shorte, multiecho, csi2D, and ssfp. However, `svsis` can be easily modified to produce images from user 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 modify `svsis` for a user sequence, 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`. The string characters for `seqcon` are defined as follows:

- First character: multiecho looping
- Second character: multislice looping
- Third character: 2D phase encode loop
- Fourth character: 3D phase encode loop
- Fifth character: 4D phase encode loop

The values of each character are ‘n’ for a null loop, ‘s’ for a standard loop, or ‘c’ for a compressed loop.

Related: 
- `rts` Retrieve shim coil settings (C)
- `shimspath` Path to user’s shims directory (P)

**svsis**

Perform spin simulation calculation (C)

**spsm**

Enter spin system (M)
In this case, 'ncsn' is a standard 2D image with compressed multislice. The $thk$ value is the slice thickness type, as defined by the type of acquisition, which in this case is the standard image sequence.

svsis uses the command svsdfd to dump the transformed data out to the data file. After dumping the headers out, the UNIX shell command fdfgluer is called to glue the headers to the data. svsdfd dumps the data in such a way that the (0,0) coordinates are the first data point in the file.

More detailed modifications can be made to svsis but it is left to the user to make these adjustments. Modifications to the macro should be made in the user's maclib.

Arguments: directory is the directory name desired. The specified directory is made in the user’s data directory and is appended with the suffix .dat. Image files are created under this directory as image0001.fdf, image0002.fdf, etc. A procpar file is also saved into this directory.

'f' | 'm' defines the type of image data. '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'. ImageBrowser currently only accepts data in floating point values.

See also: VnmrJ Imaging NMR

Related: seqcon Acquisition loop control (P)

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: VnmrJ Liquids NMR

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)

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).

The sampling rate is:

- 12.5 ns on INOVA.
- 100 ns on MERCURY.
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.

A value of \(sw\) greater than the value of the \(maxsw_loband\) parameter forces \(dp='y'\).

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:

- On \(UNITY\) or \(INOVA\): 100 Hz to 500 kHz.
- On \(MERCURYplus/Vx\), 100 Hz to 100 kHz.
- On \(UNITY\) or \(INOVA\) with solids: up to 5 MHz.

See also: \(VnmrJ\) Liquids NMR

**Related:**
- \(dp\) Double precision (P)
- \(dsp\) Type of DSP for data acquisition (P)
- \(maxsw_loband\) Maximum spectral width of input board (P)
- \(oversamp\) Oversampling factor for acquisition (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)

**\(sw1\)**

**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: \(VnmrJ\) Liquids NMR

**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: \(VnmrJ\) Liquids NMR

**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)
**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: *VnmrJ Liquids NMR*

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 CONFIG window (opened by entering config) 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:
- boresize: Magnet bore size (P)
- config: Display current configuration and possibly change it (M)
- creategtable: Generate new gradient calibration file (M)
- gcoil: Current gradient coil (P)
- gmax: Maximum gradient strength (P)
- setgcoil: Assign sysgcoil configuration parameter (M)
- trise: Gradient rise time (P)

**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 CONFIG window (opened from config).

Values: 'spectrometer' is a spectrometer system (Spectrometer choice in CONFIG window).
        'datastation' is a system used as a data station (Data Station choice in CONFIG 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 \texttt{vnmrsystem} initializes \texttt{systemdir} at bootup.
  See also: \textit{VnmrJ Liquids NMR}
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>$T_1$ exponential analysis (M)</td>
</tr>
<tr>
<td>timage</td>
<td>Fit arrayed imaging data to $T_1$ exponential data (M)</td>
</tr>
<tr>
<td>t1s</td>
<td>$T_1$ exponential analysis with short output table (M)</td>
</tr>
<tr>
<td>t2</td>
<td>$T_2$ exponential analysis (M)</td>
</tr>
<tr>
<td>t2image</td>
<td>Fit arrayed imaging data to $T_2$ exponential data (M)</td>
</tr>
<tr>
<td>t2s</td>
<td>$T_2$ exponential analysis with short output table (M)</td>
</tr>
<tr>
<td>tabc</td>
<td>Convert data in table order to linear order (M)</td>
</tr>
<tr>
<td>tan</td>
<td>Find tangent value of an angle (C)</td>
</tr>
<tr>
<td>tape</td>
<td>Read tapes from VXR-style system (M,U)</td>
</tr>
<tr>
<td>tape</td>
<td>Control tape options of files program (P)</td>
</tr>
<tr>
<td>tbox</td>
<td>Draw a tilted box (C)</td>
</tr>
<tr>
<td>tcapply</td>
<td>Apply table conversion reformatting to data (C)</td>
</tr>
<tr>
<td>tcclose</td>
<td>Close table conversion file (C)</td>
</tr>
<tr>
<td>tcl</td>
<td>Send Tcl script to Tcl version of dg window (C)</td>
</tr>
<tr>
<td>tcopen</td>
<td>Open table conversion file (C)</td>
</tr>
<tr>
<td>te</td>
<td>Echo time (P)</td>
</tr>
<tr>
<td>techron</td>
<td>Set up parameters for gradient amplifier tests (M)</td>
</tr>
<tr>
<td>temp</td>
<td>Open the Temperature Control window (C)</td>
</tr>
<tr>
<td>temp</td>
<td>Sample temperature (P)</td>
</tr>
<tr>
<td>tempcal</td>
<td>Temperature calculation (C)</td>
</tr>
<tr>
<td>tep</td>
<td>Post-acquisition delay in EPI experiments (P)</td>
</tr>
<tr>
<td>testct</td>
<td>Check ct for resuming signal-to-noise testing (M)</td>
</tr>
<tr>
<td>testsn</td>
<td>Test signal-to-noise of a spectrum (M)</td>
</tr>
<tr>
<td>teststr</td>
<td>Find which array matches a string (M)</td>
</tr>
<tr>
<td>text</td>
<td>Display text or set new text for current experiment (C)</td>
</tr>
<tr>
<td>textis</td>
<td>Return the current text display status (C)</td>
</tr>
<tr>
<td>textvi</td>
<td>Edit text file of current experiment (M)</td>
</tr>
<tr>
<td>th</td>
<td>Threshold (P)</td>
</tr>
<tr>
<td>th2d</td>
<td>Threshold for integrating peaks in 2D spectra (P)</td>
</tr>
<tr>
<td>thadj</td>
<td>Adjust threshold for peak printout (M)</td>
</tr>
<tr>
<td>theta</td>
<td>Euler angle theta from magnet frame (P)</td>
</tr>
<tr>
<td>thk</td>
<td>Slice thickness (P)</td>
</tr>
<tr>
<td>ti</td>
<td>Inversion recovery time (P)</td>
</tr>
<tr>
<td>ticks</td>
<td>Number of trigger pulses (P)</td>
</tr>
<tr>
<td>time</td>
<td>Display experiment time or recalculate number of transients (M)</td>
</tr>
<tr>
<td>tin</td>
<td>Temperature interlock (P)</td>
</tr>
<tr>
<td>title</td>
<td>Plot a title on a plotter (M)</td>
</tr>
<tr>
<td>tilt</td>
<td>First-order baseline correction (P)</td>
</tr>
<tr>
<td>tmove</td>
<td>Left-shift FID to time-domain cursor (M)</td>
</tr>
<tr>
<td>tmsref</td>
<td>Reference 1D proton or carbon spectrum to TMS (M)</td>
</tr>
<tr>
<td>tn</td>
<td>Nucleus for observe transmitter (P)</td>
</tr>
<tr>
<td>tncosyps</td>
<td>Set up parameters for TNCOSYPS pulse sequence (M)</td>
</tr>
<tr>
<td>tndqcosy</td>
<td>Set up parameters for TNDQ COSY pulse sequence (M)</td>
</tr>
</tbody>
</table>
t1 exponential analysis (M)

Description: Processes data obtained using an array of values of the parameter \(d2\) for a \(T_1\) experiment. It runs \texttt{expfit}, which does an exponential curve fitting that determines the value of \(T_1\). The output is matched to the equation:

\[
M(t) = (M(0) - M0) \times \exp(-t/T1) + M0
\]

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 \texttt{fp(index1,index2,...)} before running the analysis. The output file is the \texttt{analyze.list} in the current experiment. The file \texttt{analyze.out} is used by \texttt{exp1} to display the results. The output of the analysis program shows \( T_1 \) and its standard deviation, but does not explicitly show \( M(0), M_0 \), or their standard deviations. The \( M(0) \) and \( M_0 \) values can be found in “raw” form in \texttt{analyze.out} in the current experiment, but their standard deviations are not part of the program output.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{d2} \hspace{1em} Incremented delay in 1st indirectly detected dimension (P)  
\texttt{expfit} \hspace{1em} Make least squares fit to polynomial or exponential curve (C)  
\texttt{fp} \hspace{1em} Find peak heights (C)  
\texttt{t1s} \hspace{1em} \( T_1 \) exponential analysis with short output table (M)  
\texttt{t2} \hspace{1em} \( T_2 \) exponential analysis (M)  
\texttt{t2s} \hspace{1em} \( T_2 \) exponential analysis with short output table (M)

\texttt{t1image} \hspace{1em} \textbf{Fit arrayed imaging data to} \( T_1 \) \textbf{exponential data (M)}

Applicability: Systems with imaging capabilities.

Description: Does preprocessing required for fitting arrayed imaging data to \( T_1 \) data using the \texttt{imfit} program. The user is prompted for the base phasefile names and the lower limit noise threshold. \texttt{t1image} then transforms and saves all of the images, and calls \texttt{imfit} to complete the fitting process.

See also: \textit{VnmrJ Imaging NMR}

Related: \texttt{imfit} \hspace{1em} Fit arrayed imaging data to \( T_1 \) or \( T_2 \) exponential data (M,U)  
\texttt{t2image} \hspace{1em} Fit arrayed imaging data to \( T_2 \) exponential data (M)

\texttt{t1s} \hspace{1em} \textbf{\( T_1 \) exponential analysis with short output table (M)}

Description: Performs the same analysis as \texttt{t1} but produces a short output table showing only a summary of the measured relaxation times.

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{t1} \hspace{1em} \( T_1 \) exponential analysis (M)

\texttt{t2} \hspace{1em} \textbf{\( T_2 \) 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 \texttt{expfit}, which does an exponential curve fitting that determines the value of \( T_2 \). The output is matched to the equation:

\[
M(t) = (M(0) - M(\text{inf}))*\exp(-t/T2) + M(\text{inf})
\]

where \( M(0) \) is the magnetization at time zero (i.e., the full magnetization excited by the observe pulse) and \( M(\text{inf}) \) is the xy-magnetization at infinite time (zero unless the peak is sitting on an offset baseline).

The required input is the file \texttt{fp.out} from \texttt{fp} and the values of the arrayed parameter. The \( T_2 \) analysis is done for all the peaks listed in \texttt{fp.out}. Peaks are selected for analysis by entering \texttt{fp(index1,index2,...)} before running the analysis. The output file is the file \texttt{analyze.list} in the current experiment. The file \texttt{analyze.out} is used by \texttt{exp1} to display the results. The output of the analysis program shows \( T_2 \) and its standard deviation, but does not explicitly show \( M(0), M(\text{inf}), \) or their standard deviations. The \( M(0) \) and \( M(\text{inf}) \) values can be found in “raw” form in \texttt{analyze.out} in the current experiment, but their standard deviations are not part of the program output.
t2image  
**Fit arrayed imaging data to \( T_2 \) exponential data (M)**

**Applicability:** Systems with imaging capabilities.

**Description:** Does preprocessing required for fitting arrayed imaging data to \( T_2 \) data using the `imfit` program. The user is prompted for the base phasefile names and the lower limit noise threshold. `t2image` then transforms and saves all of the images, and calls `imfit` to complete the fitting process.

See also: *VnmrJ Imaging NMR*

Related: 
- `imfit` Fit arrayed imaging data to \( T_1 \) or \( T_2 \) exponential data (M,U)
- `t1image` Fit arrayed imaging data to \( T_1 \) exponential data (M)

\( T_2 \) 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: *VnmrJ Liquids NMR*

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., \(-64, -63, -62, \ldots, -1, 0, 1, \ldots, 62, 63\)). 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 \( n_1 \) standard 2D planes of data (the third dimension), each consisting of \( n_2 \) 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: `arccos` Calculate arc cosine of real number (M)
- `arcsin` Calculate arc sine of real number (M)
- `arctan` Calculate arc tangent of real number (M)
- `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:
- **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)

---

**tbox**

Draw a tilted box (C)

Applicability: Systems with imaging capabilities.

Syntax:
1. `tbox('<keywords'>angle,xcenter,ycenter, hlen,vlen)`
2. `tbox('<keywords'>angle,xcenter,ycenter, hlen,vlen, vspace,nboxes)`

Description: Draws a tilted box centered at xcenter, ycenter (as indicated by a small diamond) (syntax 1) or produces an aligned array of nboxes tilted boxes centered at xcenter, ycenter (syntax 2) and separated by vspace.

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

angle is the tilt angle, in radians, of a box.

xc, yc are coordinates on the screen, in mm, specifying the point at which a box is centered.

hlen is the horizontal coordinate on the screen, in mm.

vlen is the vertical coordinate, on the screen, in mm.

vspace controls the separation or overlap of boxes.

nboxes is the number of boxes.

Examples: tbox('plotter', 20, 100, 40, 150)

See also: VnmrJ Imaging NMR

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

tcapply Apply table conversion reformatting to data (C)

Applicability: Systems with imaging capabilities.

Syntax: tcapply<(file)>

Description: Rearranges the spectra in a 2D data set that resides in the current data file. You must apply ft1d to the data before you can use tcapply. Using values from an AP table, tcapply arranges the spectra corresponding to the value in the AP table from low value to high value. The values might have already been read in by the tcopen command.

Arguments: file specifies the name of the file containing the AP table to be read. The file must be in $vnmruser/tablib.

Examples: tcapply('petable')

See also: VnmrJ Imaging NMR

Related: tabc Close table conversion file (C)

ft1d Fourier transform along f2 dimension (C)

ft2d Fourier transform along f1 dimension (C)

tcclose Close table conversion file (C)

tcopen Open table convert file (C)

tcclose Close table conversion file (C)

Applicability: Systems with imaging capabilities.

Description: Removes a table conversion file and frees the memory used to store the sorted table indices read in with the tcopen command.
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

Open table conversion file (C)

**Applicability:** Systems with imaging capabilities.

**Syntax:**
tcopen<(file)>

**Description:** Explicitly reads, sorts, and stores in memory, a table conversion file. tcopen uses the file when tcapply is called.

**Arguments:**
- `file` specifies the file to be read; it must be in $vnmruser/tablib.

**Examples:**
tcopen('petable')

See also: VnmrJ Imaging NMR

Echo time (P)

**Applicability:** Systems with imaging capabilities.

**Description:**
- Echo time for imaging and some localized spectroscopy experiments.
- In gradient and spin echo imaging sequences, `te` is usually defined as the time measured from the middle of the initial rf excitation pulse to the center of the resulting echo.
- In multiecho sequences, `te` may also define the time duration between successive echoes, normally a constant interval. Multiecho sequences with variable echo times are also possible, in which case the `te` period between successive echoes may take on a range of values represented by a `te` array.
- Some more unusual pulse sequences, such as stimulated echo, RARE and Fast Spin Echo, may use `te` in ways somewhat different from the normal standards.

See also: VnmrJ Imaging NMR

Set up parameters for gradient amplifier tests (M)

**Applicability:** Systems with imaging capabilities.

**Description:**
Recalls parameters sets for gradient amplifier tests during microimaging installation.

See also: Microimaging Module Installation
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: VnmrJ Liquids NMR

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)
- `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)

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: VnmrJ Liquids NMR

Related:
- `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: VnmrJ Liquids NMR

tep Post-acquisition delay in EPI experiments (P)
Applicability: Systems with echo planar imaging (EPI) capabilities.
Description: Delay used in the EPI sequence to adjust the beginning of data acquisition. This
correction is necessary to allow for the finite (propagation) delay of gradient
pulses. This allows the user to center the EPI echoes in the acquisition window.
Values: Number, in µs. Typically 0 to 50 µs, depending on the gradient hardware.
See also: VnmrJ Imaging NMR
Related: episet Set up parameters for EPI experiment (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-
oise. See the description of testsn for details.
See also: VnmrJ Liquids NMR
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 r? (using a conservative estimate), and then sets
  the processing at future blocks to be only testct, which simply tests if
  ct is greater than r?, and, if so, resumes testing of signal-to-noise with
  testsn.
See also:  VnmrJ Liquids NMR

Related:  
- **cli3**: 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)
- **rl-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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: `thadj` Adjust threshold for peak printout (M)
th2d

**Threshold for integrating peaks in 2D spectra (P)**

**Description:** Used by 112d 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('112d').

**Values:** From 0.0 to 1.0. If th2d=1.0, 112d 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 112d to integrate a larger area when determining the volume of a peak. If th2d=0.5, for example, 112d integrates all points in a peak that are above 0.5 times the current threshold.

**See also:** VnmrJ Liquids NMR

**Related:**
- addpar: Add selected parameters to the current experiment (M)
- 112d: 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:** VnmrJ Liquids NMR

**Related:**
- nll: Find line frequencies and intensities (C)
- ppf: Plot peak 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)

theta

**Euler angle theta from magnet frame (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Euler angle theta from magnet frame.

**Values:** –90 to +90, in degrees.
See also: *VnmrJ Imaging NMR*

**thk**

**Slice thickness (P)**

*Applicability:* Systems with imaging capabilities.

*Description:* Returns the slice thickness, in mm.

*See also:* *VnmrJ Imaging NMR*

**ti**

**Inversion recovery time (P)**

*Applicability:* Systems with imaging capabilities.

*Description:* Specifies the recovery time following an inversion prepulse in inversion recovery experiments. The value of \( ti \) generally has a strong impact on image contrast, which depends on the \( T1 \) relaxation time of the sample in different regions of the image.

*See also:* *VnmrJ Imaging NMR*

**ticks**

**Number of trigger pulses (P)**

*Applicability:* Systems with imaging capabilities.

*Description:* Sets the number of trigger pulses the system waits before acquisition begins. This parameter is found in some Varian pulse sequences that feature gating. \( ticks \) controls an external gating signal received through an external TTL input. If \( ticks=0 \), the system ignores trigger pulses and runs in the nontriggered mode. The pre- and post-trigger delays \( rcvry \) and \( hold \) remain active in the nontriggered mode.

*Values:* Integers from 0 to 100.

*See also:* *VnmrJ Imaging NMR*

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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)

**title**

**Plot a title on a plotter (M)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** `title(string)`

**Description:** Plots a string provided by the user on the plotter.

**Arguments:**
- `string` is a string of characters.

**Examples:**
- `title('15 June Image')`

See also: *VnmrJ Imaging NMR*

**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: *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**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 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-CH₃ signal. Large signals within 0.6 ppm for ¹H (or 6 ppm for ¹³C) 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:** *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**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)

**tncoyps**  
*Set up parameters for TNCOSYPS pulse sequence (M)*

**Applicability:** Sequence is not supplied with *MERCURYplus/Vx*.

**Description:** Sets up a homonuclear correlation experiment (phase-sensitive version) with water suppression.

**See also:** *VnmrJ Liquids NMR*

**tndqcosy**  
*Set up parameters for TNDQCOSY pulse sequence (M)*

**Applicability:** Systems with a linear amplifier on the observe channel and a T/R switch. Sequence is not supplied with *MERCURYplus/Vx*.
Description: Sets up a 2D J-correlation experiment with water suppression.
See also: VnmrJ Liquids NMR

**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. Sequence not supplied with MERCURYplus/Vx.

Description: Sets up a multiple-quantum filtered COSY experiment with water suppression.
See also: VnmrJ Liquids NMR

**tnnoesy**  Set up parameters for TNNOESY pulse sequence (M)

Applicability: Systems with a linear amplifier on the observe channel and a T/R switch. Sequence is not supplied with MERCURYplus/Vx.

Description: Sets up a 2D cross-relaxation experiment with water suppression.
See also: VnmrJ Liquids NMR

**tnroesy**  Set up parameters for TNROESY pulse sequence (M)

Applicability: Sequence is not supplied with MERCURYplus/Vx.

Description: Sets up a rotating-frame NOE experiment with water suppression.
See also: VnmrJ Liquids NMR

**tntocsy** Set up parameters for TNTOCSY pulse sequence (M)

Applicability: Systems with T/R switch, computer-controlled attenuators, and linear amplifiers on observe channel. Sequence not supplied with MERCURYplus/Vx.

Description: Sets up a total-correlation spectroscopy experiment (HOHAHA) with water suppression.
See also: VnmrJ Liquids NMR

**TOCSY**  Change parameters for TOCSY experiment (M)

Description: Converts the current parameter set to a TOCSY experiment.

**Tocsy**  Convert the parameters to a TOCSY experiment (M)

Description: Convert parameters to a TOCSY experiment.

**tocsy** Set up parameters for TOCSY pulse sequence (M)

Applicability: Any system with linear amplifiers on the observe channel.

Description: Sets up a total-correlation (TOCSY) experiment, also known as the Homonuclear Hartmann-Hahn (HOHAHA) experiment.
See also: VnmrJ Liquids NMR

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: Proton(M) sel1d(M)

TOCSY1D

Change parameters for TOCSY1D experiment (M)

Description: Converts the current parameter set to a TOCSY1D (also known as DPFGSE-noe) experiment. A 1D proton spectrum is displayed to do peak selection.

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 CONFIG window (opened from config, implicitly set for MERCURYplus/Vx systems). 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: VnmrJ Liquids NMR

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)

tpe

Duration of the phase encoding gradient pulse (P)

Applicability: Systems with imaging capabilities.

Description: Sets the length of the phase encoding gradient period in imaging and CSI experiments. The spectral width in the indirect dimension (sw1) is determined from tpe as sw1=1/tpe. tpe may be recomputed within the pulse sequence to provide optimum performance, such as minimum echo time, or scaled to match the required timing for slice refocusing and readout dephasing.

See also: VnmrJ Imaging NMR

Related:
gpe Phase encoding gradient increment in DAC units (P)
nv Number of 2D phase encode steps to be acquired (P)
sw1 Spectral width in 1st indirectly detected dimension (P)
tpe2, tpe3 Duration of second and third phase encoding gradient periods (P)

tpe2, tpe3

Duration of second and third phase encoding gradient periods (P)

Applicability: Systems with imaging capabilities.

Description: Sets the lengths of the phase encoding gradient periods that control second spatial and third spatial dimensions in nD imaging and CSI experiments.

For example, 3D volume imaging sequence have two independent phase encode axes, controlled by tpe and tpe2. It is common to have a single phase encoding time block, in which two independent phase encode gradients share the same time period. In this case, tpe and tpe2 would be equal.

See also: VnmrJ Imaging NMR

Related:
sw2 Spectral width in 2nd indirectly detected dimension (P)
tpe Duration of the phase encoding gradient pulse (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 CONFIG window (opened from `config`). 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.
- On MERCURYplus/Vx systems, the range is 0 to 63, in dB, 1-dB steps.

**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: `VnmrJ Liquids NMR`

**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)

**tpwr1**  
**Intensity of an excitation pulse (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Specifies the peak power, in dB, of transmitter pulses corresponding to `p1`.

See also: `VnmrJ Imaging NMR`

**Related:**
- `p1` First pulse width (P)
- `tpwr` Observe transmitter power level with linear amplifiers (P)

**tpwr2**  
**Intensity of an excitation pulse (P)**

**Applicability:** Systems with imaging capabilities.

**Description:** Specifies the peak power, in dB, of transmitter pulses corresponding to `p2`.

See also: `VnmrJ Imaging NMR`

**Related:**
- `p2` Second pulse width (P)
- `tpwr` Observe transmitter power level with linear amplifiers (P)

**tpwrcal**  
**Calibrate power levels of 90° and 180° pulse (M)**

**Applicability:** Systems with imaging capabilities.

**Syntax:** `tpwrcal(start_tpwr,end_tpwr)`
Description: Sets up paired arrays of form \( tprw1, tprw2 \). The parameter array is set as array='(tprw1, tprw2)'. This macro is especially useful for calibrating the 90° and 180° power levels for a slice.

Arguments: start_tprw is the starting value for the tprw part of the arrayed pairs. The starting value for tprw1 is 6 less than start_tprw.

end_tprw is the ending value for the tprw part of the arrayed pairs. The ending value for tprw1 is 6 less than end_tprw.

Examples: \( tprwcal(30, 45) \)

See also: *VnmrJ Imaging NMR*

Related: array Parameter order and precedence (P)
tprw Observe transmitter power level with linear amplifiers (P)
tprw1 Intensity of excitation pulse (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 CONFIG window (opened from config). The fine attenuator is linear and spans 60 dB (\( 10^6 \) INOVA) or 6 dB (other systems). If tprwrf is not present, enter

\[
\text{create('tpwrf', 'integer') setlimit('tpwrf', 4095, 0, 1)}
\]

to create it.

On MERCURYplus/Vx systems, controls the transmitter by simulating a fine attenuator. The fine power control is linear and spans 0 to tprw.

Values: 0 to 4095, where 4095 is maximum power. If tprwrf does not exist in the parameter table, a value of 4095 is assumed.

On MERCURYplus/Vx systems, 0 to 255 (where 255 is maximum power). If tprwrf or tprwrm do not exist in the parameter table, a value of 255 is assumed. If both exist, tprwrm is used.

See also: *VnmrJ Liquids NMR; User Guide: Solids; MERCURYplus/Vx CP/MAS Installation, Testing, and Operation*

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)
tprw Observe transmitter power level with linear amplifier (P)
tprwrm Observe transmitter linear modulator power (P)

**tpwri**

Intensity of inversion pulse (P)

Applicability: Systems with imaging capabilities.

Description: Specifies the peak power of transmitter pulses corresponding to \( pi \).

Values: Number, in dB.

See also: *VnmrJ Imaging NMR*

Related: ir Inversion recovery mode (P)
pi Width of an inversion pulse in microseconds (P)
tprw Observe transmitter power level with linear amplifiers (P)
tprw1 Intensity of an excitation pulse (P)
tpwrm  Observe transmitter linear modulator power (P)

Description: Controls the power level on the observe transmitter linear modulator. On MERCURYplus/-Vx systems, tpwrm controls the transmitter by simulating a fine attenuator. 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.

On MERCURYplus/Vx systems, 0 to 255 (where 255 is maximum power). If tpwrm does not exist in the parameter table, a value of 255 is assumed.

See also: VnmrJ Liquids NMR; User Guide: Solids; MERCURYplus/-Vx CP/MAS Installation, Testing, and Operation

Related: config  Determine current configuration and possibly change it (M)
dpwrf  First decoupler fine power (P)
fattn  Fine attenuator (P)

tr  Repetition time in imaging and localized spectroscopy (P)

Applicability: Systems with imaging capabilities.

Description: Sets the repetition time of an experiment. The definition of repetition time can vary somewhat from pulse sequence to pulse sequence. In general, for imaging experiments, tr is the time required to complete one transient of one phase encode step, including relaxation delay, excitation, data acquisition, and any post-acquire events, such as rf spoiling, phase encode rewinding, and gradient turn-off.

For multislice and/or multiecho imaging sequences, tr includes the complete multislice/multiecho train (for standard arrayed slice acquisitions, where the second character in seqcon is s, the complete train is not included, and tr is the repetition time for each slice position).

Some 1D experiments, such as STEAM and ISIS are also written using tr, with the similar definition that tr is the repetition time per transient.

tr describes the total duration of all events in a pulse sequence, and will never be directly found as an argument to “delay.” Instead, tr will generally be used in precalculations to determine the time required to pad the sum of programmed events up to the desired repetition time. This padding delay will often be found in the pulse sequence as “predelay.”

See also: VnmrJ Imaging NMR

Related: seqcon  Acquisition loop control (P)

trace  Mode for n-dimensional data display (P)

Applicability: All systems; however, MERCURYplus/Vx systems can only process 3D data and cannot acquire such data.

Description: Sets the multidimensional data display mode.

Values: 'f1' displays the f1 axis horizontally and allows f1 traces to be displayed.

'f2' displays the f2 axis horizontally and allows f2 traces to be displayed.

'f3' displays the f3 axis horizontally and allows f3 traces to be displayed if the data set is 3D.

See also: VnmrJ Liquids NMR

transfer  Move parameters to target experiment (M)

Applicability: Systems with imaging capabilities.
Syntax: \texttt{transfer(data	extunderscore type,\textless scout	extunderscore exp,\textgreater target	extunderscore exp)}

Description: Transfers selectively parameter data from a scout data set to the target experiment in preparation for the next or future scanning operation. The following series of actions are carried out: (1) \texttt{transfer} joins the scout experiment and saves the current parameters in the userdir+’/parlib' directory, under the file name TRANSFER.par. Any previous parameter sets with this file name are removed. (2) \texttt{transfer} then joins the target experiment and displays the transfer menu. The user may then use the menu to selectively copy groups of parameters from TRANSFER.par to the target experiment. The groups that may be transferred include:

- **Nucleus**: \texttt{tn, resto}
- **Voxel**: \texttt{pos1-pos3, voxel-vox3, psi1, thetal, mopos, scpos}
- **Slice**: \texttt{pss, psi, phi, theta, mopos, scpos}
- **FOV**: \texttt{lro, lpe}
- **Coil**: \texttt{rfcoil, gcoil}
- **Sample**: \texttt{mopos, scpos}

If any of the parameters \texttt{pos1, pos2, pos3, thetal, psi, phi, or theta} are arrayed in the scout experiment, in addition to copying the voxel or slice list, \texttt{transfer} sets the \texttt{array} parameter in the target experiment. Other parameters copied by \texttt{transfer} cannot legally be arrayed, except \texttt{pss}.

Parameters \texttt{tn, gcoil}, and \texttt{pss} are special cases that trigger \texttt{_macros} execution. \texttt{transfer} executes the \texttt{_tn, _gcoil, and _pss (setloop)} programs once if these parameters are copied to the target. This execution ensures that all the normal side effects of setting these parameters are properly executed.

Arguments: \texttt{data	extunderscore type} is a keyword defining the type of data for transfer as 'slice' or 'voxel', which can be abbreviated to 's' or 'v', respectively.

\texttt{scout	extunderscore exp} is the number of the scout experiment. The default is the current experiment is the source of the scout parameter data.

\texttt{target	extunderscore exp} is the number of the target experiment.

Examples: \texttt{transfer('s',5)}
\texttt{transfer('v',5,6)}

See also: \textit{VnmrJ Imaging NMR}

Related: \texttt{gcoil} Read data from gradient calibration tables (P)
\texttt{lpe} Field of view size for phase encode axis (P)
\texttt{lro} Field of view size for readout axis (P)
\texttt{phi} Euler angle from magnet frame (P)
\texttt{psi} Euler angle from magnet frame (P)
\texttt{pss} Slice position (P)
\texttt{resto} NMR resonance offset frequency (P)
\texttt{rfcoil} RF pulse calibration identity (P)
\texttt{theta} Euler angle from magnet frame (P)
\texttt{tn} Nucleus for observe transmitter (P)
\texttt{userdir} VnmrJ user directory (P)

\textbf{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 CONFIG window (opened from \texttt{config}).
Values: 0 is setting for no sample changer present or, if a sample changer is attached, to disable the changer (None choice in the CONFIG 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 NMR*

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

**trfunc**

*Translate screen coordinates (M)*

**Applicability:** Systems with imaging capabilities.

**Syntax:** `trfunc($x,$y):$xincm,$yincm`

**Description:** Translates screen coordinates to hertz or centimeters depending upon the `axis` parameter.

**Arguments:**
- `$x` is a coordinate . . .
- `$y` is a coordinate . . .
- `$xincm` is a coordinate . . .
- `$yincm` is a coordinate . . .

See also: *VnmrJ Imaging NMR*

Related: `axis` Axis label for displays and plots (P)

**trfuncd**

*Translate screen distance (M)*

**Applicability:** Systems with imaging capabilities.

**Syntax:** `trfuncd($screenlength):$imagelength`

**Description:** Translates a screen distance into centimeters in a real image. It is only useful in `axis='cc'` (aspect ratio constrained) images.

**Arguments:**
- `$screenlength` is the length of the display screen.
- `$imagelength` is the length of the image.

See also: *VnmrJ Imaging NMR*

Related: `axis` Axis label for displays and plots (P)

**trise**

*Gradient rise time (P)*

**Applicability:** Systems with imaging capabilities.

**Description:** Stores the time required for an x, y, or z magnetic field gradient to change from zero to maximum gradient (gmax). Because the gradient system is adjusted by Varian at installation time so that all three gradients have the same rise time, only one parameter is used to describe the rise time for all three gradients.

This parameter accurately describes the time required for gradient changes only in systems that use slew-rate-limited gradient amplifiers, such as the Oxford GPS 2239 gradient amplifier supplied with most imaging systems. Do not confuse this gradient rise time with the amount of time required by a pulse sequence to transmit the DAC value that initiates a gradient value change (see the `gradient` and `vgradient` statements in the manual *User Programming* for a discussion of that timing).
trise is used in some sequences to control various aspects of gradient timing, including the automatic setup of gradient refocusing. This parameter does not need to be declared and initialized in pulse sequence source code files, because it is a standard PSG parameter and is therefore already declared and initialized by the Varian-supplied PSG library. See the source file `sems.c` for an example.

`trise` is defined in the system gradient table files found in the directory `$vnmrsystem/gradtables`, and is automatically set from one of those files when a value is entered for the parameter `gcoil`.

Values: 0.005 seconds (nominal).

See also: VnmrJ Imaging NMR

Related: boresize Magnet bore size (P)  
gcoil Read data from gradient calibration tables (P)  
gmax Maximum gradient strength (P)

troesy

Set up parameters for TROESY pulse sequence (M)

Applicability: Not available on MERCURYplus/Vx systems.

Description: Sets up parameters for the transverse cross-relaxation experiment in a rotating frame.

See also: VnmrJ Liquids NMR

trunc

Truncate real numbers (O)

Description: In MAGICAL programming, an operator that truncates real numbers.

Examples: $3 = \text{trunc}(3.6)$

See also: User Programming

Related: acos Find arc cosine of number (C)  
arccos Calculate arc cosine of real number (M)  
arcsin Calculate arc sine of real number (M)  
arctan Calculate arc tangent of real number (M)  
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)

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

tspoil

Gradient spoiling time (P)

Applicability: Systems with imaging capabilities.
Description: Delay parameter for use in controlling a spoiling gradient. Many imaging sequences use tspoil to set the additional time that the slice-select gradient is on, symmetrically bracketing the 180° refocusing pulse, to spoil any magnetization excited by the 180 itself.

See also: VnmrJ Imaging NMR

Related: gcrush  
Crusher gradient level (P)

gspoil  
Spoiler gradient level (P)

tugain  
Amount of receiver gain used by qtune (P)

Description: Sets the amount of receiver gain used by the interactive probe tuning program qtune. On some systems, the default receiver gain of 50 causes the signal to saturate, which qtune displays as a mostly flat line. To adjust the receiver gain to avoid saturation, set tugain to an appropriate value for the system before qtune is started.

Values: 0 to 60, in steps of 2 dB (60 represents the highest possible receiver gain and 0 the lowest). On UNITY/INOVA (500-MHz and higher), low-band gain is limited 18 to 60. On MERCURY, typically 0, 0-38, 2 dB steps.

See also: VnmrJ Liquids NMR

Related: qtune  
Tune probe using swept-tune graphical tool (C)

tune  
Assign a frequency to a channel for probe tuning (C)

Applicability: UNITY/INOVA systems.

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 1H, the second channel at 13C, and the third channel at 15N. 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: VnmrJ Liquids NMR

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)
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)

tuneoff Turn off probe tuning mode on MERCURYplus/-Vx (M)

Applicability: MERCURYplus/Vx systems.

Description: Takes a MERCURYplus/Vx broadband system out of tuning mode by turning off the transmitter directing rf to the probe. After entering tuneoff, be sure to change the cables on the probe and magnet leg back to the normal BNC connectors (as they were before they were moved for tuning purposes).

See also: VnmrJ Liquids NMR; Autoswitchable NMR Probes Installation

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:
on Make a parameter active or test its state (C)
size Return number of elements in an arrayed parameter (O)
### Undospins

**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 `spin1.inpar` and the transitions from the file `spin1.savela` in the current experiment.

**See also:** VnmrJ Liquids NMR

**Related:** `spins` Perform spin simulation calculation (C)

### Undosy

**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:** VnmrJ Liquids NMR

**Related:** `dosy` Process DOSY experiments (M)  
`redosy` Restore 2D DOSY display from subexperiment (M)

### Unit

**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 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)
ri=10k will set ri to 10000

unit('p','ppm','reffrq','processed')
ri=10p will set ri to 10*reffrq, where reffrq from processed tree

unit('p','','0')
ri=10p will set ri to 10 and give an error "unknown unit p"

unit('F','degF',5/9,-32*5/9)
ri=212F will set ri to 100 (degrees C)

unit('C','degC',9/5,32)
ri=100C will set ri to 212 (degrees F)
```

**See also:** *VnmrJ Liquids NMR, User Programming*

**Related:**
- `axis` Axis label for displays and plots (P)
- `bootup` Macro executed automatically when VnmrJ is activated (M)

---

**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 \texttt{wexp} processing). This lock can be left behind if the program or the computer crashes.

The \texttt{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:** \texttt{exp\_number} is the number of the experiment from 1 to 9 to be unlocked.

\texttt{force} unlocks an experiment under all circumstances and joins the unlocked experiment.

**Examples:**

\texttt{unlock(3)}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{jexp} Join existing experiment (C)

**updatepars** Update all parameter sets saved in a directory (M)

**Syntax:** \texttt{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. \texttt{updatepars} searches a directory for parameter and FID files and corrects the \texttt{procpar} files found. This macro overwrites parameters in the current experiment. The corrections applied to the parameter sets are defined by the \texttt{parfix} macro. Because \texttt{updatepars} uses the current experiment to process the parameter sets, the experiment chosen for running \texttt{updatepars} should not contain a valuable data set.

**Arguments:** \texttt{directory} is the name of the directory to be searched.

**Examples:**

\texttt{updatepars('myparlib')}

\texttt{updatepars('mydata')}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{parfix} Update parameter sets (M)

\texttt{parversion} Version of parameter set (P)

**updateprobe** Update probe file (M)

**Syntax:** \texttt{updateprobe(<probe|'tmplt'>,<,'system'>)}

**Description:** Updates the current existing probe file or probe template.

**Arguments:**

\texttt{probe} is the probe parameter to update. The default is the current probe parameter value.

\texttt{'tmplt'} is a keyword to update the local probe template. The default is the current probe file.

\texttt{'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:**

\texttt{updateprobe}

\texttt{updateprobe('autosw')}

\texttt{updateprobe('autosw','system')}

\texttt{updateprobe('tmplt')}

See also: \textit{VnmrJ Liquids NMR}

Related: \texttt{adddparams} Add 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: *VnmrJ Liquids NMR; User Programming; VnmrJ Imaging NMR*

**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: *VnmrJ Liquids NMR*

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:**  
*VnmrJ Liquids NMR*

**Related:**  
*curexp*  
*systemdir*  

**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:**  
*VnmrJ Liquids NMR*

**Related:**  
*au*  
*ga*  
*go*  
*go_*

**userfixpar**  
*Macro called by fixpar (M)*

**Description:** Called by the macro *fixpar* to provide an easy mechanism to customize parameter sets.

**See also:**  
*VnmrJ Liquids NMR*

**Related:**  
*fixpar*  

**userselection**  
*Selection for images and frames (P)*

**Description:** A string for selecting images and frames (selection syntax). Used by display commands.

**Examples:**  
g1\-3, \ g1(1\-4) [5\-]
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vast1d</td>
<td>Set up initial parameters for VAST experiments (M)</td>
</tr>
<tr>
<td>vastget</td>
<td>Selects and displays VAST spectra (M)</td>
</tr>
<tr>
<td>vastglue</td>
<td>Assemble 1D datasets into a 2D (or pseudo-2D) dataset (M)</td>
</tr>
<tr>
<td>vastglue2</td>
<td>Assemble 1D datasets into a 2D (or pseudo-2D) dataset (M)</td>
</tr>
<tr>
<td>vastgo</td>
<td>Turn off LC stop flow automation, start VAST automation (M)</td>
</tr>
<tr>
<td>vbg</td>
<td>Run VNMR processing in background (U)</td>
</tr>
<tr>
<td>vf</td>
<td>Vertical scale of FID (P)</td>
</tr>
<tr>
<td>vi</td>
<td>Edit text file with vi text editor (M)</td>
</tr>
<tr>
<td>vjhelp</td>
<td>Display VnmrJ help (U)</td>
</tr>
<tr>
<td>vn</td>
<td>Start VNMR directly (U)</td>
</tr>
<tr>
<td>vnmr</td>
<td>Start VNMR in current windowing system (U)</td>
</tr>
<tr>
<td>vnmr2sc</td>
<td>VNMR to SpinCAD pulse sequence translator (M)</td>
</tr>
<tr>
<td>vnmr_accounting</td>
<td>Open Accounting window (U)</td>
</tr>
<tr>
<td>vnmrexit</td>
<td>Exit from the VNMR system (C)</td>
</tr>
<tr>
<td>vnmrj</td>
<td>Start Vnmrj (U)</td>
</tr>
<tr>
<td>vnmrplot</td>
<td>Plot files (U)</td>
</tr>
<tr>
<td>vnmrprint</td>
<td>Print text files (U)</td>
</tr>
<tr>
<td>vo</td>
<td>Vertical offset (P)</td>
</tr>
<tr>
<td>vorient</td>
<td>Voxel orientation</td>
</tr>
<tr>
<td>vox1 - vox3</td>
<td>Voxel dimensions (P)</td>
</tr>
<tr>
<td>voxplan</td>
<td>Set voxel parameters for voxel defined by 2D box cursor (M)</td>
</tr>
<tr>
<td>vp</td>
<td>Vertical position of spectrum (P)</td>
</tr>
<tr>
<td>vpf</td>
<td>Current vertical position of FID (P)</td>
</tr>
<tr>
<td>vpfi</td>
<td>Current vertical position of imaginary FID (P)</td>
</tr>
<tr>
<td>vphi,vpsi,vtheta</td>
<td>Euler angles for voxel orientation</td>
</tr>
<tr>
<td>vs</td>
<td>Vertical scale (P)</td>
</tr>
<tr>
<td>vs2d</td>
<td>Vertical scale for 2D displays (P)</td>
</tr>
<tr>
<td>vsadj</td>
<td>Automatic vertical scale adjustment (M)</td>
</tr>
<tr>
<td>vsadj2</td>
<td>Automatic vertical scale adjustment by powers of 2 (M)</td>
</tr>
<tr>
<td>vsadjc</td>
<td>Automatic vertical scale adjustment for $^{13}$C spectra (M)</td>
</tr>
<tr>
<td>vsadjh</td>
<td>Automatic vertical scale adjustment for $^1$H spectra (M)</td>
</tr>
<tr>
<td>vsproj</td>
<td>Vertical scale for projections and traces (P)</td>
</tr>
<tr>
<td>vtc</td>
<td>Variable temperature cutoff point (P)</td>
</tr>
<tr>
<td>vtttype</td>
<td>Variable temperature controller present (P)</td>
</tr>
<tr>
<td>vtwait</td>
<td>Variable temperature wait time (P)</td>
</tr>
<tr>
<td>vxr_unix</td>
<td>Convert VXR-style text files to UNIX format (M,U)</td>
</tr>
</tbody>
</table>

**vast1d**

**Set up initial parameters for VAST experiments (M)**

**Applicability:** Systems with VAST accessory.

**Description:** Sets up initial VAST parameters from the `/vmnr/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: VnmrJ Liquids NMR

**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: VnmrJ Liquids NMR

**vastglue**  
**Assemble 1D datasets into a 2D (or pseudo-2D) dataset (M)**

Applicability: Systems with the VAST accessory.

Syntax: vastglue(<rack,<zone>)

Description: Used to artificially reconstruct a 2D dataset 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 dataset 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: VnmrJ Liquids NMR

Related: autoname Prefix for automation data file (P)

vastglue2 Assemble related 1D datasets into a 2D (or pseudo-2D) dataset (M)

**vastglue2**  
**Assemble 1D datasets into a 2D (or pseudo-2D) dataset (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: `VnmrJ Liquids NMR`

Related: `autoname` Prexix for automation data file (P)  
`vastglue` Assemble related 1D datasets into a 2D (or pseudo-2D) dataset (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: VnmrJ Liquids NMR

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)
3dd 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
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)

**vjhelp**

Display VnmrJ help (U)

Syntax: `vjhelp file:///vnmr/jhelp/jhelp.html`

Description: Displays the VnmrJ help in a Netscape 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 &
v -display hostname:0.0 &
v -font 8x13 &
```

See also: VnmrJ Liquids NMR

Related:
- `vnmr` Start VNMR (U)

**vnmr**

Start VNMR in current windowing system (U)

Description: Starts the VNMR application using the current windowing system. `vnmr` can also be used to start VNMR from terminals. In this case, the `vnmr` command is equivalent to the `vn` command.

See also: VnmrJ Liquids NMR

Related:
- `vn` Start VNMR in window environment (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
V

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

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)
Description: Starts the VnmrJ application using the current windowing system.
Arguments: -display, don't add fonts, for example vnmrj -display hostname adm, opens the VnmrJ Administration interface (vnmrj adm).

See also: VnmrJ Liquids NMR; VnmrJ Walkup NMR

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: VnmrJ Liquids NMR
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
```
V

See also: *VnmrJ Imaging NMR*

Related: *transfer* Move parameters to target experiment (M)

**voxplan**

*Set voxel parameters for voxel defined by 2D box cursor (M)*

**Applicability:** Systems with imaging capabilities.

**Description:** Calculates and sets the voxel parameters for the voxel defined by the position of the 2D box cursor. The parameter for the voxel can be calculated and set using the Calculate Target button of the voxel planning menu. This uses the `voxplan` macro. See the `plan` macro for details.

See also: *VnmrJ Imaging NMR*

Related: *drawsliwx* Display target slices (M)

*drawvox* Display target voxels (M)

*plan* Display menu for planning a target scan (M)

*ssplan* Set slice parameters for target slice (M)

**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: *VnmrJ Liquids NMR*

Related: *vpf* Current vertical position of FID (P)

*vpfi* Current vertical position of imaginary FID (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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

**Related:**
- *addpar* Add selected parameters to the current experiment (M)
- *axiaf* Axis label for FID displays and plots (P)
- *crf* Current time-domain cursor position (P)
- *delaf* Difference of two time-domain cursors (P)
- *dotflag* Display FID as connected dots (P)
- *vp* Vertical position of spectrum (P)
- *vps* Vertical scale (P)
- *vps2d* Vertical scale for 2D displays (P)
- *vpsi* Euler angle for voxel orientation (P)
- *vtheta* Euler angle for voxel orientation (P)

**Euler angles for voxel orientation**

**Applicability:** Systems with imaging capabilities.

**Description:** Euler angles used to define voxel orientation. Definitions are similar to the imaging plane orientation definition parameters *phi*, *psi*, and *theta*.

Generally, voxel Euler angles are not directly set by the user, but instead are set either by entering a string value into *vorient* or through interactive graphical planning of a voxel plane from an existing scout image.

**Related:**
- *phi* Euler angle for defining imaging plane orientation (P)
- *psi* Euler angle for defining imaging plane orientation (P)
- *theta* Euler angle for defining imaging plane orientation (P)
- *plan* Interactive slice and voxel selection (M)
- *vorient* Voxel orientation (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:** *VnmrJ Liquids NMR*

**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*.

**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 \(0.9 \times (wc2max - vp - sc2)\).

*Examples:* vsadj
vsadj(100)

*See also:* VnmrJ Liquids NMR

*Related:* ai Select absolute intensity mode (C)
isdj 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 \(0.9 \times (wc2max - vp - sc2)\).

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:* VnmrJ Liquids NMR

*Related:* aexppl Automatic expansions plot (M)
isdj 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 \(^1\)H spectra (M)
wc2max Maximum width of chart in second direction (P)

vsadjc  
**Automatic vertical scale adjustment for \(^{13}\)C 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 \(0.9 \times (wc2max - vp - sc2)\).

*Examples:* vsadjc
vsadjc(wc2max-sc2-wc2-5)
**vsadjh**  
**Automatic vertical scale adjustment for ¹H 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:**  
*VnmrJ Liquids NMR*

**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 ¹H 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.

**Related:**
- `par2d` Create 2D acquisition, processing, and display parameters (M)
- `vs` Select vertical scale (C)
- `vs2d` Adjust vertical scale for 2D displays (M)

**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: VnmrJ Liquids NMR

Related: temp Sample temperature (P)
         tin Temperature interlock (P)

**vttype**

**Variable temperature controller present (P)**

Description: In the CONFIG 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 CONFIG window (opened from config).

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 CONFIG is not affected by entering vttype on the command line.

Values: 2 is setting for VT controller (Present choice in CONFIG window).
        0 is setting for no VT controller (Not Present choice in CONFIG 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; VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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:  *VnmrJ Liquids NMR*

Related:  
- **convert**  
  Convert data set from a VXR-style system (C,U)
- **decomp**  
  Decompose a VXR-style directory (C)
W

Who is using system (C)
walkup  Walkup automation (M)
waltz  WALTZ decoupling present (P)
wbs  Specify action when bs transients accumulate (C)
wbs  When block size (P)
wc  Width of chart (P)
wc2  Width of chart in second direction (P)
wcmax  Maximum width of chart (P)
w2cmax  Maximum width of chart in second direction (P)
werr  Specify action when error occurs (C)
werr  When error (P)
wt  flag to turn on or off wet solvent suppression (P)
wetld  Set up parameters for a WET1D pulse sequence (M)
Wetld  Set up parameters for a WET1H experiment (M)
wetdqcqosy  Set up parameters for a WETDQ COSY pulse sequence (M)
wetgcosy  Set up parameters for a WETGCOSY pulse sequence (M)
wetghmqcps  Set up parameters for a WETGHMQCPS pulse sequence (M)
wetghsqt  Set up parameters for a WETGHQC pulse sequence (M)
wetgmqcosy  Set up parameters for a WETGHQC pulse sequence (M)
wetnoesy  Set up parameters for a WETNOESY pulse sequence (M)
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)
xexp  Specify action when experiment completes (C)
wexp  When experiment completes (P)
wft  Weight and Fourier transform 1D data (C)
wftld  Weight and Fourier transform f2 for 2D data (C)
wftlda  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)
wfft3  Process f3 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)
wpt  Width of plot in directly detected dimension (P)
wpt1  Width of plot in 1st indirectly detected dimension (P)
wpt2  Width of plot in 2nd indirectly detected dimension (P)
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 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 Liquids NMR

Related: enter Enter sample information for automation run (M,U)
globalauto Automation directory name (P)

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 CONFIG window.

Values: 'n' sets WALTZ decoupling not present.
'y' sets WALTZ decoupling present.

See also: VnmrJ Installation and Administration

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.

Examples: wbs('dg wft')
wbs('mf(3)')
wbs('"

See also: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related: bs Block size (P)
wbs Specify action when bs transients accumulate (C)

wc Width of chart (P)
Description: Specifies the width of the chart (plotting or printing area).

Values: 5 to wcmax, in mm.

See also: VnmrJ Liquids NMR

Related: wc2 Width of chart in second direction (P)
wcmax Maximum width of chart (P)

cutoff Data truncation limit (P)
ho Horizontal offset (P)
sc2 Start 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: *VnmrJ Liquids NMR*  
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: *VnmrJ Liquids NMR*  
Related:  
- **wc2**  
  Width of chart in second direction (P)  
- **wcmax**  
  Maximum width of chart (P)

**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: *VnmrJ Liquids NMR*  
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 (\texttt{werr=''}, where the value is given by two single quotes with no space between them).

See also: *VnmrJ Liquids NMR*

Related: \texttt{acqstatus} Acquisition status (P)  
\texttt{react} Recover from error conditions during \texttt{werr} processing (M)  
\texttt{werr} Specify action when error occurs (C)

\textbf{wet} \hspace{2cm} \textbf{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.

See also: *appetype, hetero2d, homo2d, std1d, wet1d*

\textbf{wet1d} \hspace{2cm} \textbf{Set up parameters for a WET1D pulse sequence (M)}

Applicability: Systems with LC-NMR accessory.

Description: Sets up for a WET1D LC-NMR experiment.

See also: *VnmrJ Liquids NMR*

\textbf{Wet1d} \hspace{2cm} \textbf{Set up parameters for wet $^1H$ experiment (M)}

Description: Set up parameters for wet $^1H$ experiment.

\textbf{wetdqcosy} \hspace{2cm} \textbf{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: *VnmrJ Liquids NMR*

\textbf{wetgcossy} \hspace{2cm} \textbf{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: *VnmrJ Liquids NMR*

\textbf{wetghmqcps} \hspace{2cm} \textbf{Set up parameters for a WETHMQCPS pulse sequence (M)}

Applicability: Systems with LC-NMR accessory.

Description: Sets up for a WETHMQCPS LC-NMR experiment.

See also: *VnmrJ Liquids NMR*

\textbf{wetghsqc} \hspace{2cm} \textbf{Set up parameters for a WETGHSQC pulse sequence (M)}

Applicability: Systems with LC-NMR accessory.

\texttt{wetghsqc('nucleus')}

Description: Sets up for a WETGHSQC LC-NMR experiment.

See also: *VnmrJ Liquids NMR*
**wetgmqcosy**  
Set up parameters for a WETGHSQC pulse sequence (M)

- **Applicability:** Systems with LC-NMR accessory.
- **Description:** Sets up for a WETGHSQC LC-NMR experiment.
- **See also:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

**Related:**

- **wbs**  
  Specify action when bss 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, the execute 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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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 $(2 \times ni)/sw1$, in seconds.

See also: VnmrJ Liquids NMR

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)
**W**

**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 \((2 \times \text{ni2})/\text{sw2}\), in seconds.  
**See also:** *VnmrJ Liquids NMR*  
**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)  
- w: 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. \(\text{wft}(\text{<options,}>'\text{nf}',\text{<start,}'>\text{finish},\text{<step>})\)  
2. \(\text{wft}('\text{inverse}',\text{exp_number,}exp\_\text{factor})\)  
**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 \(\text{lsfid, phfid,} and \text{lsfrq,}\) respectively, on the time-domain data prior to the weighting and Fourier transformation. The type of Fourier transformation to be performed is determined by \(\text{proc.}\) \(\text{wft}\) uses the same arguments as the command \(\text{ft,}\) and except for weighting, it functions the same as the \(\text{ft}\) command.  
**See also:** *VnmrJ Liquids NMR*  
**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 f2 for 2D data (C)*  
**Syntax:**  
1. \(\text{wft1d}(<\text{element_number}>\text{)}\)  
2. \(\text{wft1d}(\langle\text{options,}\text{<coefficients>}\rangle\text{)}\)  
**Description:** Performs the first Fourier transformation along the dimension defined by \(\text{sw,}\) with weighting and matrix transposition. This allows the display of \(t1\) interferograms with the \(\text{dcon}\) and \(\text{dconi}\) commands. Except for weighting, \(\text{wft1d}\) functions the same as the \(\text{ft1d}\) command. See the description of \(\text{ft1d}\) for further information.  
**Arguments:** Same as the arguments to \(\text{ft1d}\). See the \(\text{ft1d}\) command for details.  
**See also:** *VnmrJ Liquids NMR*  
**Related:**  
- dcon: Display noninteractive color intensity map (C)  
- dconi: Interactive 2D data display (C)  
- ft1d: Fourier transform along f2 dimension (C)  
- sw: Spectral width in directly detected dimension (P)
wft1da  **Weight and Fourier transform phase-sensitive data (M)**

Syntax: `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: $VnmrJ$ Liquids NMR

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: $VnmrJ$ Liquids NMR

Related: $ft1dac$  Combine arrayed 2D FID matrices (M)

$tocsys$  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:

$wft2d(rri,ri1,rr2,ir2,...ri1,ii1,ri2,ii2,...)$. $wft2d('ptype',...)$ 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: \texttt{wft2d(1,0,0,0)}  
\texttt{wft2d(2)}  
\texttt{wft2d(1,0,1,0,0,1,0,1)}  
\texttt{wft2d(.67,0,.33,0,0,.67,0,.33)}

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \begin{tabular}{ll}
\texttt{dconi} & Interactive 2D data display (C) \\
\texttt{ft1d} & Fourier transform along f_2 dimension (C) \\
\texttt{ft1da} & Fourier transform “halfway” for pure absorption 2D data (M) \\
\texttt{ft2d} & Fourier transform 2D data (C) \\
\texttt{wft1d} & Weight and Fourier transform f_2 for 2D data (C) \\
\texttt{wft1da} & Weight and FT “halfway” for pure absorption 2D data (M) \\
\texttt{wft2d} & Weight and transform for pure absorption 2D data (M) \\
\end{tabular}

\newpage

\textbf{wft2da \quad Weight and Fourier transform phase-sensitive data (M)}

\textbf{Syntax: \texttt{wft2da(<options>)}}

\textbf{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.

\texttt{wft2da} differs from \texttt{ft2da} only in that weighting of the time-domain data is performed prior to the Fourier transform. See the description of \texttt{ft2da} for further information.

\textbf{Arguments:} Same as used with \texttt{ft2da}. See the \texttt{ft2da} command for details.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \begin{tabular}{ll}
\texttt{ft1da} & Fourier transform phase-sensitive data (M) \\
\texttt{ft2da} & Fourier transform phase-sensitive data (M) \\
\texttt{wft1da} & Weight and Fourier transform phase-sensitive data (M) \\
\end{tabular}

\newpage

\textbf{wft2dac \quad Combine arrayed 2D FID matrices (M)}

\textbf{Syntax: \texttt{wft2dac(<mult1>,<mult2>,...<,multn>)}}

\textbf{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. \texttt{wft2dac} is used with TOCSY (with multiple mixing times).

\textbf{Arguments:} \texttt{mult1}, \texttt{mult2}, ..., \texttt{multn} are multiplicative coefficients. The \texttt{n}th argument is a real number and specifies the multiplicative coefficient for the \texttt{n}th 2D FID matrix.

See also: \textit{VnmrJ Liquids NMR}

\textbf{Related:} \begin{tabular}{ll}
\texttt{ft1dac} & Combine arrayed 2D FID matrices (M) \\
\texttt{ft2dac} & Combine arrayed 2D FID matrices (M) \\
\texttt{tocsy} & Set up parameters for TOCSY pulse sequence (M) \\
\texttt{wft1dac} & Combine arrayed 2D FID matrices (M) \\
\end{tabular}

\newpage

\textbf{wftt3 \quad Process f_3 dimension during 3D acquisition (M)}

\textbf{Description:} Allows f_3 processing of 3D data to be performed concurrently with data acquisition. To invoke this function, set \texttt{wnt='wftt3'} and use \texttt{au} to start the acquisition of the 3D data. When \texttt{wftt3} detects that all the FIDs comprising a (t_1, t_2) block have been acquired, it starts up the \texttt{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 f1 and f2 processing information but only valid f3 processing information. Once the f3 processing is complete, a new 3D information file can be created for the f1-f2 processing stages that contains valid f1 and f2 processing information.

The non-standard string parameter `path3d` can be used to specify the directory into which the f3 processed 3D data is to be stored. Normally, `path3d` is absent in the parameter set. If this is the case or if `path3d=' '`, the f3-processed 3D data is stored in the directory `curexp/datadir`. `path3d` can be created by entering `create('path3d','string')` `setgroup('path3d','display')`.

See also: *VnmrJ Liquids NMR*

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 Varian) 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` Determin 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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**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:**

```
write('keywords'><,color|pen>
<,'reverse'>,x,y<,template>) <:height>
```
(2) write('alpha'|'printer'|'line3'|'error',template)
(3) write('reset'|'file'|'fileline',file,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','%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 (linefeed) to the end of the string defined by the template argument. To append data without the automatic linefeed, 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 (linefeed) to the end of the data. Any linefeeds 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.

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')

See also: User Programming
Related: dtext Display a text file in the graphics window (M)

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: VnmrJ Liquids NMR, 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)

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.

wrtp
Command string executed after rt 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.
'a' 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 (this option not implemented on MERCURYplus/Vx systems).

'fn', where n is an integer, sets shimming is done prior to data collection of every nth FID (e.g., wshim='f16' shims prior to acquiring FIDs 1, 17, 33, etc.). This method is only relevant to arrayed or 2D experiments (this option not implemented on MERCURYplus/Vx systems).

See also: VnmrJ Liquids NMR

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 $f_1$ 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: VnmrJ Liquids NMR; 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: VnmrJ Liquids NMR; 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: *VnmrJ Liquids NMR: User Programming*

**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 in directly detected dimension (P)
- `wtfile1` User-defined weighting in 1st indirectly detected dimension (P)
- `wti` Interactive weighting (C)

**wti**

**Interactive weighting (C)**

**Syntax:** `wti<(element_number)>`

**Description:** Allows weighting parameters to be set interactively for both t₂ FIDs and t₁ interferograms. *wti* responds appropriately to `phfid` and `lsfid` for t₂ FIDs and to `phfid1` and `lsfid1` for t₁ 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 gives 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:** *VnmrJ Liquids NMR*

**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:** *VnmrJ Liquids NMR*

**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)
- `wti` Interactive weighting (C)

**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:** *VnmrJ Liquids NMR*
**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 `vnmr/sys/global` and hence common to all experiments.

**Values:**  
Number, in mm.

**See also:**  
*VnmrJ Liquids NMR*

**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:**  
*VnmrJ Liquids NMR*

**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:**  
*VnmrJ Liquids NMR*

**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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
Related: `shimset` Type of shim set (P)

### xdiag

**Threshold for excluding diagonal peaks when peak picking (P)**

Description: Used by the 112d program to exclude diagonal peaks when peak picking.

To create the 2D peak picking parameters xdiag and th2d in the current experiment, enter `addpar('112d')`.

Values: Peaks within xdiag Hz of the diagonal will not be picked by 112d. Setting xdiag to 0.0 will cause 112d to pick all peaks, including diagonal peaks.

See also: VnmrJ Liquids NMR
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)

### xpol

**Cross-polarization (P)**

Applicability: Systems with a solids module.
Description: Selects cross-polarization or direct polarization in solid-state NMR experiments such as XPOLAR1.

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: UNITY/INOVA systems with a solids module. MERCURY with CP/MAS module.

Description: Sets up the solid-state NMR cross-polarization experiment XPOLAR using the parameters preferred for the UNITY/INOVA. Otherwise, xpolar1 contains the same functionality as xpol.

See also: User Guide: Solid-State NMR

Related: harotor  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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

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: VnmrJ Liquids NMR

Related: shimset  Type of shim set (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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
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: VnmrJ Liquids NMR
Related: shimset  Type of shim set (P)

yz  YZ shim gradient (P)
Description: Holds current setting of the YZ radial shim gradient.
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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

Related: `shimset` Type of shim set (P)
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)
zap Set up for gradient refocused high-speed imaging sequences (M)
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: \texttt{z\{reset1,reset2,\ldots\}}
**Z**

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: *VnmrJ Liquids NMR*

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`. Only on *UNITY INOVA* systems, `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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

### 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:** VnmrJ Liquids NMR

### 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:** VnmrJ Liquids NMR

### 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:** VnmrJ Liquids NMR

### 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:** VnmrJ Liquids NMR

**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:** VnmrJ Liquids NMR

### 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: *VnmrJ Liquids NMR*

**z3x2y**

**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: VnmrJ Liquids NMR*

**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: VnmrJ Liquids NMR*

**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: VnmrJ Liquids NMR*

**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: VnmrJ Liquids NMR*

**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: VnmrJ Liquids NMR*

**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: VnmrJ Liquids NMR*

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: VnmrJ Liquids NMR*

**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.
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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

**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: *VnmrJ Liquids NMR*

**zap**

**Set up for gradient refocused high-speed imaging sequences (M)**

Applicability: Systems with imaging capabilities.

Description: Sets up a pulse sequence consisting of a slice-selective excitation pulse to generate transverse magnetization.

See also: *VnmrJ Imaging NMR*

Related: *gs* Slice selection gradient strength in DAC units (P)

**zeroneg**

**Set all negative intensities of 2D spectra to zero (C)**

Description: Sets to zero all negative intensities of 2D-J spectra.

See also: *VnmrJ Liquids NMR*

Related: *fold* 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 \(\pm \frac{width}{2}\).

See also: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

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: *VnmrJ Liquids NMR*

**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: `VnmrJ Liquids NMR`

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: `VnmrJ Liquids NMR`
Symbols

& (ampersand) character, 559
.talk file, 335
/ (slash) character, 334
? (question mark) character, 479
@ (at) character, 334

Numerics

13C gHSQC exp, setting up parameters for, 276
13C HMQC exp, changing parameters for, 301
13C HMQCTOXY exp, changing parameters for, 305
13C HSQC exp, changing parameters for, 305
13C HSQCTOXY exp, changing parameters for, 305
15N gHMQC exp, setting up parameters for, 275, 276
15N gHSQC exp, setting up parameters for, 276
15N HMQC exp, changing parameters for, 300, 301
15N HMQCTOXY exp, changing parameters for, 305
15N HSQC exp, changing parameters for, 305
15N HSQCTOXY exp, changing parameters for, 305
16-bit integer precision, 181
180-degree pulse power calibration, 617
180-degree refocus pulse length, 406
shape, 406
1st indirectly detected dimension absolute value display mode, 90
additive weighting constant, 92
clear reference line, 129
cursor difference, 151
cursor position, 109, 126
data display mode, 172
first-order phase, 346
Fourier number, 241
Gaussian function, 274
Gaussian shift constant, 274
incremented delay, 138
line broadening, 331
number of increments of evolution time, 384
 phased spectra display mode, 408, 431
power display mode, 474
reference line frequency, 503
reference line position, 502
scale spectral width, 527
set frequency referencing, 552
set reference line, 506
sinebell constant, 524
sinebell shift constant, 525
spectra width, 595
start of plot, 569
user-defined weighting, 662
width of plot, 658
zero-order phase, 509
200-kHz receiver option, 181
2D display, showing, 258
2D DOSY display building up, 242
2D experiments acquire and Fourier transform, 262
axis labels, 93
baseline correction, 97
color intensity map, 143
combine arrayed FID matrices, 250, 254
control dconi display, 145
copy peak picking file to another file, 338
cross-relaxation experiment, 615
data display, 144
display a spectrum, 193
display FIDs, 154
display resolution, 76
display spectra in whitewash mode, 193, 194
draw grid on 2D display, 286
exchange experiment, 388
f2 ridges, 244
find and integrate peaks, 336
first point multiplier, 110, 245
first-order phase set to zero, 106
fold J-resolved spectrum, 242
Fourier transform 2D data, 250
Fourier transform arrayed 2D FID data, 250
general setup, 533
gray scale image, 311
heteronuclear 2D-J, 299
heteronuclear chemical shift correlation, 299
homonuclear J-resolved 2D, 302
horizontal axis selection, 619
INADEQUATE pulse sequence, 315
incremented delay, 138
intensity of spectrum at a point, 366
interactive weighting, 663, 664
interleaving control, 310
J-correlation experiment, 615
LC-NMR acquisition parameters, 332
normalization, 387
number of increments of evolution time, 384
parameter creation, 50
peak integration threshold, 611
peak picking display control, 339
peak picking parameters, 50
phase selection, 432
plot 2D peak picking results, 447
plot grid over 2D plot, 443
plot heteronuclear J-resolved 2D spectra, 444
plot homonuclear J-resolved 2D spectra, 445
plot X,H-correlation 2D spectrum, 446
plotter units conversion, 306
processing mode for 2D data, 453
processing parameter group, 158
project 2D data onto axis, 465
pseudo-echo weighting parameters, 467
reverse detection heteronuclear multiple quantum, 300
rotate 2D data, 508
search data set for maximum intensity, 426
select for processing, 266
set scaling factor, 527
sinebell weighting, 564
spectra plotting, 437
spectra plotting in whitewash mode, 438
spectra processing, 463
spectral drift correction, 142
stacked spectra display, 202
start of chart in second direction, 526
submit to acquisition, 281
symmetrize INADEQUATE data, 242
t1 dimension, 354
type of data processing, 462
vertical offset of traces, 639
volume value, 316, 317
weight and Fourier transform, 654
weight and Fourier transform 2D data, 655
weight and Fourier transform phase-sensitive data, 655, 656
2D experiments setup, 533
2D phase encode image center position, 455
2D phasefiles
  calculate, 312
  format arguments, 313
2D spectra, plotting, 449
2D spectra, setting negative intensities of, 676
2H chemical shift, 567
2nd indirectly detected dimension
  absolute value display mode, 91
  additive weighting constant, 92
  clear reference line, 129
  cursor difference, 152
  cursor position, 109, 126
  data display mode, 173
  first-order phase, 346
  Fourier number, 242
  Gaussian function, 274
  Gaussian shift constant, 275
  incremented delay, 139
  line broadening, 332
  number of increments of evolution time, 384
  phased spectra display mode, 431
  power mode processing, 475
  reference line frequency, 503
  reference line position, 502
  right phase, 510
  scale spectral width, 527
  set frequency referencing, 553
  set reference line, 306
  sinebell constant, 524
  sinebell shift constant, 525
  spectral width, 595
  start of plot, 569
  user-defined weighting, 662
  width of plot, 658
32-bit integer precision, 181
3D experiments
  3D plane index selected, 315
  3D plane projection selected, 315
  3D plane type currently displayed, 440
  axis labels, 93
  create experiment, 109
  create parameters, 411
  display 2D color map of plane from 3D data, 183
  display 2D projection plane from 3D data, 185
  display 3D data file, 164
  display group of parameters, 159
  display next 3D plane, 384
display previous 3D plane, 458
display series of 3D planes, 201
extract planes from 3D spectral data, 270
f3 ridges, 244
find and integrate peaks on 2D plane, 336
first point multiplier, 245
Fourier transform 3D FID into 3D data, 254
Fourier transform arrayed 3D data sets, 250
horizontal axis selection, 619
incremented delay, 139
N-type display, 392
number of increments of evolution time, 384
parameter creation, 50
path to 2D planes, 417
phase cycling type, 433
plot peak picking on 2D plane, 447
plot series of 3D planes, 450
process f3 dimension, 656
processing coefficient file, 534
region selective 3D processing, 469
reset parameters after partial transform, 496
select 2D plane without displaying, 529
selective 2D processing, 253
selective transformation, 252
set 3D processing, 533
spectral dc correction, 570
t1 and t2 dimensions, 355
terminal 3D FT process, 325
time-domain dc correction, 231
transformed data file, 120
type of data processing, 462
weight and FT phase-sensitive data, 655, 656
3D, showing/hiding drawings in, 537
3rd indirectly detected dimension
  incremented delay, 139
  number of increments of evolution time, 385
  spectral width, 596
3rd rf channel
  create parameters, 238
  display group of parameters, 159
  display template for parameters, 412
  parameter retrieval, 50
4D experiments
  create acquisition parameters, 412
  incremented delay, 139
  number of increments of evolution time, 385
  parameter creation, 50
  phase cycling type, 433
4th rf channel
  create parameters, 238
  display group of parameters, 159
5th rf channel
  create parameters, 239
90-degree pulse, 405
90-degree pulse power calibration, 617
90-degree pulse width, 473

A
aborting acquisition
  with error, 38
  with no error, 296
absolute intensity display mode, 53
absolute intensity group, 53
absolute magnet frame data, generating, 511
absolute-value 2D experiment, 432
absolute-value COSY pulse sequence, 495
absolute-value data display mode, 172, 173
absolute-value data file, 120
absolute-value display mode, 90
absolute-value MQF COSY parameter set, 281
absolute-value ROESY parameter set, 290
accounting program, 637
acq_errors file, 45
acqaddr parameter, 246
acqbin directory, 532
acqfil directory, 223, 602
Acqmeter window, 42, 43
acquisition
abrupt with error, 38
abrupt with no error, 296
acquire FID with no processing, 281
acquisition parameter arrays, 139
action when bs transients accumulate, 648
action when error occurs, 650
action when specified transients accumulate, 657
array of acquisition parameter, 534
automated proton and carbon, 296
automated proton and COSY, 298
automated proton, carbon, DEPT, 297
automated proton, carbon, HETCOR, 297
calculate pixel size, 495
carbon, 104
carbon and APT automatically, 106
carbon and DEPT automatically, 108
create 2D parameters, 411
create 3D acquisition parameters, 411
create 4D acquisition parameters, 412
data points to acquire, 77
data points to be acquired, 390
dataq is acquired, 140
delay before acquisition, 66
determine if active for experiment, 219
display status information, 563
DSP type, 200
estimate acquisition time, 612
fluorine, 226
hardware values, 544
interactive display, 41
LC-NMR 2D parameters, 332
loop control, 531
make equal to time requested, 612
number of echoes, 382
number of scans, 391
number of slices, 391
number of transients, 391
oversampling factor, 399
perform Autoshim experiment, 560
perform experiment, 79
phosphorus, 406
read hardware values, 487
recover from error, 485
resume paused queue, 497
resume stopped acquisition, 483
stop acquisition, 522
stopped by temperature interlock, 613
submit Autolock experiment, 341
submit change sample, Autoshim experiment, 522
submit setup experiment, 587
submit spin setup experiment, 571
time to acquire FID, 77
trigger pulses before acquisition, 612
trigger signals to wait, 392
acquisition bus trap, 47
acquisition computer
block size, 101
resetting, 39
Acquisition Controller board, 509
acquisition parameters group, 158
acquisition queue, resume after pause, 497
acquisition status, 45
acquisition status line, 41
Acquisition Status window, 44
activating current window activity, 323
activating/removing a mark, 548
active parameter, 396
active pulse length parameter list, 447
active pulse power level parameter list, 475
activity in current window, 323
ADC overflow warning, 45
add series of FIDs together, 48
add/subtract experiment, 575
add current FID, 47
add current spectrum, 569
clear experiment, 113
delete experiment, 113
interactive mode, 48
subtract FID, 588
subtract spectrum, 576
addAstack command, 48
adding a stack, 48
additive weighting constant, 91, 663
allocateWithId procedure, 378
alternateSlices(intmode) command, 67
alternating slices, 67
AM data conversion, 121
ampersand (&) character, 43, 44
amplifier band in use, 498
amplifier mode control, 67
amplifier type, 68
AMX data conversion, 121
AnalogPlus digital filter, 149
analyze.inp file, 69, 220, 222
analyze.list file, 70, 119, 601
analyze.out file, 70, 119, 222, 427
AP Interface Type label, 116
application code dimension, 382
application mode, 72
APT acquisition, 296
APT experiment, changing parameters for, 73
APT pulse sequence, 73
APT spectra
plot automatically, 440
process automatically, 73
arc cosine calculation, 40, 74
arc sine calculation, 74
arc sine of number, 76
arc tangent calculation, 74, 77
arc tangent of two numbers (Y,X), 77
argument, type, return identifier for, 624
array index for transformed image, 213
array of an acquisition parameter, 534
arrayed 1D spectra, 583
array of an acquisition parameter, 534
arrayed 2D FID matrices, 655, 656
arrayed experiment
control interleaving, 310
arrayed imaging data
fit to T1 data, 601
fit to T2 data, 602
arrayed parameter, returning number of elements in an, 565
arrayed parameters
enter as linearly spaced, 75
order and precedence, 75
arraying LP parameters, 350
assign sysgcoil, 541
asynchronous decoupler mode, 168
attached proton test, 73
attenuator
course type, 107
control, 187, 188, 618, 619
fine, 228
upper safety limit, 186
attributes of parameters, 164
audio filter board, 82
Audio Filter Type label, 82, 116
audio filters bandwidth, 228
auto lk gradient map generation, 83
auto.conf file, 213
auto_dir macro, 567
autocalibration, 40
getting with CH3I sample, 80
routines, 81, 82
setting up with CH#I sample, 80
with autotest sample, getting, 81
with autotest sample, setting up, 80
Autogain, see automatic gain
Autolock, see automatic lock
Automake Shimmap button, 279
automated
analysis of DEPT data, 85
carbon acquisition, 104
carbon and APT acquisition, 106
carbon and DEPT acquisition, 108
fluorine acquisition, 226
phosphorus acquisition, 406
proton acquisition, 294
proton and carbon acquisition, 296
proton and COSY acquisition, 298
proton, carbon, APT acquisition, 296
proton, carbon, DEPT acquisition, 297
proton, carbon, HETCOR acquisition, 297
automated gradient map generation macros, 82
automatic
2D normalization, 387
2D peak picking, 336
2D processing, 389
analysis of COSY data, 40, 41
APT spectra processing, 73
calibration, 40
COSY- and NOESY-type spectra plot, 442
generic processing, 464
heteronuclear J-resolved 2D spectra plot, 444
homonuclear J-resolved 2D spectra plot, 445
integral scale adjustment, 319, 320
macro execution, 551
plot APT-type spectra, 440
process FIDs, 464
spectra plotting, 448
vertical scale adjustment, 642
X,H-correlation 2D spectrum plot, 446
automatic gain
enable Autogain, 262, 263
ers, 46
automatic lock
ers, 46
status, 66
submit Autolock experiment to acquisition, 341
automatic phasing, 71
optimized, 72
zero-order term, 72
automatic sequence setup for gradients, 436
automatic shimming, 661
create shim method string, 383
method selection, 370
submit Autoshim experiment to acquisition, 522, 560
automatic stacking for arrays, 89
automatic vertical scale adjustment, 642, 643
automatic data file prefix, 87
automation directory
absolute path, 85
check for enter queue, 85
preparation for run, 83
automation directory name, 277
automation mode, 85
check if active, 84, 219
automation parameter group, 161
automation run
controlling macro, 84
enter sample information, 213
prepare automation directory, 83
resume suspended run, 88
starting, 85
suspend current run, 89
autoscaling resumes, 89
Autoshim on Z button, 279
Autoshim, see automatic shimming
autoshimming, 584
gradient, 278
average value of input, 91
axes, showing/hiding, 538
axis gradients, 117
axis labels, 92
FID displays and plots, 73, 93
units, 92

B
background execution, 559
background VNMR processing, 633
backup current probe file, 40
balance gradients, 210
bandpass filter offset for downsampling, 198
bandwidth for shaped pulse, 471
bandwidth of audio filters, 228
bandwidth of digital filter, 196
baseline correction, 96, 613
linear, 142
sensitivity adjustment, 356
zero-order, 356
baseline flatness, 304
beeper sound, 97
beginning interactive image planning, 583
Bessel filters, 66
BINOM pulse sequence, 98
binomial water suppression, 98
blanked amplifiers, 508
block size action, 649
block size storage, 101
block size transients, 648
bore size of magnet, 98

boxes
  draw on plotter or display, 98
  selected by mark command, 99

BR24 pulse sequence, 100, 132
Brickwall digital filter, 149
broadband amplifier, 68
broadband channel tuning, 101
Bruker data files, 579
  convert to VNMR, 120
  read files from 9-track tape, 485
Bruker data, phasing, 72
bruker.par file, 579
Butterworth filter, 66, 82
button labels, 374
button values, reporting, 277

C
C13.par file, 556
Calculate Target button, 439, 565, 581, 640
calculated spectrum display, 199
calibration
  decoupler pulse, 476
  gradient strength, 465
  gradients, 291
  rf pulse identity, 501
  shaped pulses, 95
calibration file, printing, 85
carbon
  acquisition, 296, 297
  automated acquisition, 104
  plotting, 441, 451
  process 1D carbon spectra, 105
  vertical scale adjustment, 642
carbon decoupler calibration macros, 81
carbon gradient ratio calibration macros, 81
carbon observe calibration macros, 81
carbon-enriched molecules, 297
Carr-Purcell Meiboom-Gill T2, 125
cartridge tape, 604
center frequencies of nD experiments, 493, 494
center of screen display limits, 109
center sequence calibration, 535
chained acquisition, 86
chained experiments, 112
CHAN readout, 623
change sample experiment, 111
changing working directory, 107
channels
  assign frequencies for probe tuning, 623
  available for use, 393
  rf frequencies, 570
  rf generation on each channel, 504
  set frequency of rf channels, 539
  waveform generator on channel, 505
characters in a string, 334
chart
  maximum width, 650
  maximum width in second direction, 650
  starting position, 526
starting position in second direction, 526
width, 649
width in second direction, 649
chart paper
  preprinted paper for HP plotters, 304
chemical shift offset frequency, 549
chemical shifts list, storing, 178, 425
chemist-style parameters, 100
class C amplifiers, 68, 107, 117
decoupler high-power control, 161
decoupler low-power control, 167
Clear Marks button, 439
clearing
  experiment text, 130
  integral reset points, 132
clearing all stacks, 112
clearStacks command, 112
cmd parameter, 406
Coarse Attenuator label, 107, 117
course attenuator type, 107
coe file, 534, 570
coefficient to construct interferogram, 227, 228
coefficients for digital filtering, 195, 580
coil calibration data, 470
Cold Probes, 130
COLOC sequence, 446
color intensity map, 311
display, 143
  without screen erase, 146
color selection for drawing, 426
colors for plotting, 113, 549
combining arrayed 2D FID matrices, 250, 254
comm port for sample changer, 116
command execution, 217
commands
  addAstack, 48
  alternateSlices(intmode), 67
  clearStacks, 112
  dbsetup remove, 141
dbupdate, 141
display which command or macro is used, 657
edit online description, 365
loadPrescription(char* path), 340
online description, 365
removeAstack(int index), 495
rename, 299
comparing shim sets, 163
compiling
  user PSG object library, 467
  user pulse sequences, 532
  user-written weighting functions, 663
combining pulse sequence, 532
compiled FIDs in experiment, 108
completed transients, 130
complex 3D transformed data file, 120
complex Fourier transform, 461, 462
complex points to left-shift ni interferogram, 354
complex points to left-shift ni2 interferogram, 355
complex points to left-shift np FID, 354
complex time-domain data points, 349, 350
compressed 2D data conversion, 239
configuration information, 661
configuration parameters
  display and possibly change, 114
Configure label, 116
conjugate gradient list, 581
Index

conpar file, 115, 127
console hardware status, 310
Console label, 115
console parameter, 119
contact time, 119
continuous wave (CW) modulation, 174
contour display
display control, 145
contour plot, 143, 424
display, 181
width of plotting area, 649
without screen erase, 146
contour plot display, 182
conversion units for parameters, 625
converting
32-bit data files to VNMR, 579
Bruker data, 120
compressed 2D data to standard format, 239
data in table order to linear order, 602
Hz or ppm to plotter units, 306
VXR-style data to VNMR, 125
VXR-style text files to UNIX, 644
coordinate information from image display, 341
copying
experiment data to subfile, 125
files, 123, 124
local file to remote host, 216
one parameter tree to another, 288
peak file to another file, 338
remote file to local host, 215
stored phasefile to current experiment, 512
system macro to become user macro, 361
user macro files, 358
CORBA client, 130
CORBA server, 130
corrected difference between successive spectra, 233
correlated spectroscopy, 124
cosine value, 123
cosine-squared window function, 577
cost accounting, 637
COSY
acquisition, 298
automatic analysis and plot, 408
automatic analysis of data, 40, 41
phase-sensitive, 124
plotting, 442
pulse sequence, 124
COSY experiment, changing parameters for, 124
COSY-like correlation spectra, 242
cp command (UNIX), 123, 124
CP/MAS amplifier, 68
CPMG2 pulse sequence, 125
creating
FID display parameters, 234
LC-NMR parameters, 415
parameters in a parameter tree, 127
UNIX directory, 374
creating/deleting a mark, 548
cross-polarization, 666
cross-relaxation, 615
crusher gradient level, 265
CryoBay Monitor software, 130
cubic curve fitting, 70, 221
curecc file, 541
curpar file, 127
current experiment
correct parameter characteristics, 238
determine if acquisition active, 219
current FID data block, 110
current gradient coil, 264
current window, 131
current working directory, 473
current-type parameter tree, 127
cursor
adjust tau2 to start of acquisition, 622
difference of two frequency cursors, 151, 152
difference of two time-domain cursors, 152
frequency offset calculation, 395
mode, 366
move cursor to center spectrum, 109
move cursor to nearest line, 385
move spectral window according to cursors, 376
reset integral to zero at cursor, 671
set decoupler frequency to cursor position, 527, 528
split difference of two cursors, 575
state in df, ds, or dconi programs, 129
cursor position, 126
time domain, 128
curve fitting, 69, 220
cutoff point for VT regulation, 643
CW amplifier mode, 67
cycle phase, 124
cycled BR24 pulse sequence, 132
cycled MREV8 pulse sequence, 132
CYCLEN8E sequence, 132

D
D2PUL pulse sequence, 138
DAC, converting gauss/cm value to, 234
data acquisition mode, 547
data conversion to linear order, 602
data display mode, 172, 173
data entry, updating, 267
data file display in current experiment, 146
data point, determining size of a, 72
data points to be acquired, 390
data processing type on FID, 461, 462
data set conversion from VXR-style to VNMR, 120, 125
data station system configuration, 596
data truncation limit, 131
data.fdf file, 255, 257
database for VnmrJ, 141
datadir directory, 657
date of data acquisition, 140
dbsetup remove command, 141
dbupdate command, 141
dc correction, 231
de offsets removed from FIDs, 146
dconi.out file, 366
decay curves, 429
decoupler
adjust tip-angle resolution time, 170
decoupling sequence, 195, 196
field strength, 169
field strength calculation, 295
fine power attenuator, 187
frequency, 155
frequency offset array, 528
frequency offset control, 177
high-power control, 161
homodecoupling control, 302
linear modulator power, 188
low-power mode, 167
mode during status periods, 167, 168
modulation frequency, 116, 169
modulation mode, 173, 174, 175
nucleus lookup, 175
power level with linear amplifier, 186
power to switchable probe caution, 186, 187
proton decoupler pulse calibration, 455
pulse calibration, 476
pulse length, 454
pulse sequence diagram, 185
set frequency to cursor position, 527
tip-angle resolution, 191
used as transmitter, 138
used for pulsing, 174
WALTZ decoupling present, 648
decoupler 2 parameter values
set from probe file, 536
decoupler modulation frequency, 169, 170
decoupler parameter values
set from probe file, 536
def file, 177
Default button, 149
default directory for Files menu system, 149
defaultdomain file, 539, 548
defaultrouter file, 539, 548
Define excitation band, 420, 530
Define excitation band for solvent suppression, 420
delay
first, 138
incremented delay for pulse sequence, 138, 139
overhead delay between FIDs, 137
post-trigger, 302
preacquisition, 66, 409
pre-trigger, 484
wait for another trigger, 208
wait to acquire a spectrum, 208
delay-type parameter, 127
deleting
all stacks, 112
experiments, 151
file, parameter, or FID directory, 150
files, 507
selected slice, 151
selected stack/slice, 150
spectra from analysis, 151
user macro, 150
deleting/creating a mark, 548
DEPT
acquisition, 297
analysis and plot, 409
automated complete analysis, 85
automatic analysis and spectrum editing, 52
plotting data, 442
pulse sequence, 152
spectra array processing, 153
DEPT experiment, changing parameters for, 152
dep.t.out file, 52
DEPTGL pulse sequence, 152
destroying
parameters, 153
parameters of a group, 153
devicenames file, 550
deviceatable file, 550
dg window, 606
dgroup of a parameter, 538
diagonal parameter arrays, 75
diagonal peaks threshold during peak picking, 666
dialog box, 119
dialog box from a macro, 162
dialog, starting a, 177, 178, 214
dialoglib directory, 177
difference between cursors in Hz, 151
difference between cursors in seconds, 152
difference NOE experiment, 132
diffusion analysis, 221
add to current display, 223
add to current plot, 427
display, 222
diffusion constant, 429
calculation, 430
diffusion experiment analysis, 70
diffusion gradient level, 266
Diffusion Ordered Spectroscopy (DOSY), 179
digital filter type, 149
digital filtering
bandwidth, 196, 580
bandwidth for oversampling, 397
coefficients for oversampling, 397
create downsampling parameters, 413
create parameters for oversampling, 416
downsampling factor, 180
file of FIR digital filter coefficients, 236
inline type, 200
number of coefficients, 195, 580
parameter creation, 50
digital lock display, 487
digital resolution measurement, 191
digitally filtered FIDs, 163, 581
dimension of application code, 382
dimension of experiment, 76, 105
dimensionality of experiment, 268
dimensions of voxel, 639
direct polarization, 667
directly detected dimension
absolute value display mode, 90
additive weighting constant, 91
data display mode, 172
first-order phase, 345
Fourier number, 241
Gaussian function, 273
Gaussian shift constant, 274
line broadening, 331
phase angle display mode, 407
phased spectra display mode, 430
power display mode, 474
reference line frequency, 502
reference line position, 502
scale spectral width, 526
set reference line, 505
sinebell constant, 524
sinebell shift constant, 525
spectral width, 594
start of plot, 569
Index

user-defined weighting, 662
width of plot, 658
zero-order phase, 509
directories
  change working directory, 107
  create new UNIX directory, 374
  default for Files menu system, 149
delete, 150
display current working directory, 473
get information about files, 268
list files, 163
list files in directory, 334, 353
move directory, 378, 495
path to current experiment, 131
remote VXR-style system, 213
remove empty directories, 507
remove from experiment, 112
rename directory, 378, 495
stored queue experiments, 85
user’s macro directory path, 358
user’s menu directory, 369
user’s manual directory, 365
user’s private VNMR files, 629
VNMR system, 597

disabling image planning, 388
disk file errors, 47
display
  acquisition information, 44
  lock level, 43
  spinner speed, 43
  temperature, 43
display limits
  set for full screen, 258
  set for full screen with room for traces, 258
  set for left half of screen, 333
  set for right half of screen, 505
display mode for plotter, 451
display parameters
  create 3D display parameters, 411
  full recall of set, 245
  move between experiments, 368
  recall set, 483
  save as a set, 521
  set to full spectrum, 226
display parameters group, 159
display style, setting to stripes/lines, 537
display templates for 3rd rf channel parameters, 412
display templates for pulse sequence, 555
displaying
  2D color map of 3D plane, 183
  2D data interactively, 144
  2D spectra in whitewash mode, 193
  2D spectra in whitewash mode with no screen erase, 194
  3D data file, 164
  3D parameter group, 159
  3D plane projection, 185
  3D planes, 201
  3rd/4th rf channel parameter group, 159
  acquisition information, 44
  acquisition parameter group, 158
  acquisition status information, 563
  acquisition time, 223
  add another diffusion analysis, 223, 427
  adjust display parameters, 237
  arrayed acquisition parameters, 139
  automation parameters, 161
color intensity map, 143
contour plot, 143
contour plot with screen erase, 182
contour plots, 181
create 2D parameters, 411
current working directory, 473
data file in current experiment, 146
dialog box from a macro, 162
display parameter group, 159
error messages, 216
Ethernet address, 210
experiment library, 223
experiment time, 223
fid, 154, 155
fid as connected dots, 180
fid file in current experiment, 147
fids in whitewash mode, 158
fids of 2D experiment, 154
files in directory, 163
formatted text, 658
full window display, 664
grid on 2D display, 286
horizontal LC axis, 140
inset spectrum, 317
integral amplitudes, 182
integral with a spectrum, 192
integrals at reset points, 165
LC-NMR parameters, 161
limNET nodes, 176
line frequencies above threshold, 166
lock level, 43
log file for experiment, 223
menu for planning target scan, 439
message on acquisition status line, 41
multiple images, 173
next 3D plane, 384
noninteractive gray scale image, 311
normalized integral amplitudes, 183
normalized integrals, 167
overlay as center lines/stripes, 163
overlay as stripes, 165
parameter screen menu, 161
parameter value, 479
parameters and their attributes, 164
peak frequencies, 182
phase file in current experiment, 147
plot is same as display, 664
plotted contours, 181
polynomial curves, 222
previous 3D plane, 458
processing parameter group, 158
pulse calibration data file, 470
pulse sequence diagram, 185
pulse sequence generation errors, 467
recalculated simulated spectrum, 199
remote VXR-style directory, 213
scale under spectrum or fid, 194
set for center of screen, 109
set full screen with room for traces, 258
set limits for full screen, 258
shim method string, 197
shim parameter group, 161
spectra in whitewash mode, 207
spectrum, 192
Index

spin simulation parameter arrays, 165
spin simulation parameter group, 160
spinner speed, 43
stacked FIDs, 156, 157, 158
stacked spectra, 202
stacked spectra automatically, 203
stacked spectra automatically with no screen erase, 204
stacked spectra horizontally, 204
stacked spectra horizontally with no screen erase, 206
stacked spectra with no screen erase, 206
strings in text window, 211
stripes/lines, 537
system macro file, 360
target slices, 190
target voxels, 190
temperature, 43
text file for current experiment, 609
text file in graphics window, 208
text files, 107
time of acquisition, 223
time of drift correction, 142
timeout, 107
timeout flag, 107
timeout seconds, 107
timeout value, 107
timeouts, 154
which command or macro is used, 657
width adjustment, 676
Distortionless Enhancement by Polarization Transfer, 152
done codes, 45
DOSY (Diffusion Ordered Spectroscopy) experiment, 179
dosyfit program, 179
double-precision data acquisition, 181
double-precision VNMR FID data, 114
double-quantum filtered COSY pulse sequence, 189
downsampling
  bandpass filter offset, 198
  bandwidth of digital filtering, 196
  creating parameters, 413
digital filter coefficients, 195
  factor, 180
  inline type, 200
  parameter creation, 50
  setting parameters, 375
DPFGSE-noe experiment, changing parameters for, 389, 616
DQCOSY experiment, 189
DQCOSY experiment, changing parameters for, 189
drawing a line between points, 189
drawings
  showing/hiding in 3D, 537
  showing/hiding order of, 538
drift correction
  2D spectra traces, 142
  activity flag, 142
calculation, 142
cancel, 107
group, 142
ds, 145
d.out file, 366
DSP parameter creation, 50
DSP type (see digital filtering), 200
dummy scans, 579
Dynamic Angle Spinning (DAS), 140
dynamic binding, 533

E

ecc file, 210
ecctabl reference table, 210, 269
eccTool window display, 210
echo command (UNIX), 211
echo planar imaging. See EPI experiments
echo position, determine, 283
echo time, 606
echoes
  index for transformed image, 211
  number to be acquired, 382
eddy current
  compensation data, 210
  compensation data analysis, 211
  compensation file, 131
  settings, 211
testing, 286
eddylib directory, 210, 269
editing
  files, 212
  macros, 359
  menu file, 369
  parameter file with user-selected editor, 412
  parameter file with vi editor, 412
  UNIX text files, 634
  user macro, 361
ejection of sample, 210, 212
  elements, returning number of, 565
  elliptical filters, 66, 82
time of acquisition, 223
time of drift correction, 142
timeout, 107
timeout flag, 107
timeout seconds, 107
timeout value, 107
timeouts, 154
  which command or macro is used, 657
  width adjustment, 676
EPI experiments
  acquisition delay time, 608
  apply phase correction map, 423
calculate slice gradient, 582
  calculate slice selection parameters, 582
centering echoes, 608
close phase correction map, 423
collect EPI data, 215
collection phase encoding gradient, 311
  display EPI data, 215
display image, 214
effective echo position, 212
generate phase correction map, 423
generate phase file, 214
  number of EPI images to collect, 311
  open phase correction map, 424
  process EPI data, 215
  process image, 214
  readout dephasing gradient adjuster, 288
  readout gradient adjuster, 287
  readout gradient dephaser, 212
  reverse spectral data, 214
  save images in FDF for ImageBrowser, 215
  set up parameters, 215
Ernst angle pulse calculation, 216
errmsg file, 663
error codes, 45
error conditions recovery, 485
error during acquisition, 45, 650
error handling control, 314
error message display, 216
errors in pulse sequence generation, 467
Ethernet

01-999252-00 A0604
VnmrJ 1.1D Command and Parameter Reference
address display, 210
disconnect host computer, 548
host computer connection, 539
Euler angle from magnet frame, 435, 468, 611
evolution dimension
  set spectral width, 554
evolution time increments, 384, 385
excitation pulse, 406
excitation pulse power, 617
executing VNMR command, 217
exiting from VNMR, 219, 638
exp5 (add/subtract experiment), 49, 113
experiment data retrieval, 513
experiment directory path, 131
experiment numbers list, 365
experiment parameters, restoring, 322
experiment text file
  append string, 79
  clear text, 130
experiment time display, 223
experimental frequency of transition, 113
experimental lines, assigning transitions, 76
experiments
  abort acquisition with no error, 296
  acquire and Fourier transform, 261
  acquisition time estimate, 612
  action when bs transients accumulate, 648
  action when error occurs, 650
  action when experiment completes, 652, 653
  add parameters, 49
  add parameters for FID display, 234
  append string to text file, 79
  calculate dimension, 105
  clear text file, 130
  completed transients, 130
  correct parameter characteristics, 238
  correct parameter limits and step sizes, 414
  create workspace, 109
delete an experiment, 151
determine if acquisition active, 219
dimension, 76
dimensionality, 268
display acquisition time, 223
display data file, 146
display FID file, 147
display log file, 223
display phase file, 147
edit text file, 610
experiment library display, 223
  fit data to lineshapes, 237
  get text from data file, 272
  join existing experiment, 321, 322
  make FID element using numeric text input, 363
  move display parameters between experiments, 368
  move FIDs between experiments, 370
  move parameters between experiments, 376
  nucleus selection, 556
  number of completed FIDs, 108
  parameters for basic experiment, 555
  pulse sequence setup, 282
  recalculate number of transients, 612
  remove inactive lock and join experiment, 626
  remove old files and directories, 112
replace text file, 609
resume a stopped acquisition, 483
retrieve FIDs from a file, 511
retrieve parameters from file, 512
retrieve shim coil settings, 512
save FIDs, 590
save parameters, 591
save text to a data file, 472
select 1D experiment for processing, 266
select 2D experiment for processing, 266
set up $T_1$ experiment, 180
setup macro, 629
setup macros, 239
shim values to use, 340
shimming conditions, 661
solvent selection, 556
stop acquisition, 522
string parameters for storage, 382
submit Autolock experiment to acquisition, 341
submit Autoshim experiment to acquisition, 560
submit change sample, Autoshim to acquisition, 522
submit setup experiment to acquisition, 587
submit to acquisition, 79, 281
text file display, 609
expfit.out file, 427
exponential analysis, 600, 601, 602
exponential curve, 220
exponential curve fitting, 70, 119, 221
exponential curves display, 222
exponential curves plot, 427
exponential $T_1$ or $T_2$ data fitting, 313
exponential value of number, 219
exponential weighting, 331, 332
external time base, 666
extract entries in VXR-style directory, 149
extrapolated dispersion mode, 232

F
F1 linear prediction parameters, setting, 548
$f_1$ scaling factor for 2D multipulse sequences, 527
$f_1$, $f_2$ display, 392
$f_2$ ridges, 110
$f_2$ processing of 3D data, 656
FDF files, 164, 229, 230, 255, 257, 590, 591, 593
FDM program, running, 230
FID block, move, 370
FID block, reverse, 498
FID button, 281
FID data, move, 371
FID data, reverse, 501
fid file, 602
FID file, memory map open, 372
FID trace, move, 373
FID trace, reverse, 503
FID, memory map close, 371
fid.orig file, 602
FIDs
  absolute-value mode data display, 173
  acquisition time, 77
  action after FID finishes, 658
  action after last FID, 653
  add series of FIDs together, 48
  add to add/subtract experiment, 47
arrayed 2D FID matrices, 655, 656
automatic processing, 464
axis label units, 93
combine arrayed 2D FID matrices, 250, 254
complex points to left-shift np FID, 354
compress double-precision FID data, 114
copy FIDs into exp5 as array, 278
create display parameters, 50, 234
current data block, 110
cursor difference, 152
delete FID directory, 150
digitally filtered FID, 581
display as connected dots, 180
display FID files in current experiment, 147
display FID of 2D experiment, 154
display scale, 194
display single FID, 154, 155
display stacked FIDs, 156, 157, 158
display whitewashed FIDs, 158
file name prefix, 87
filtered, 580
first point multiplier, 244
Fourier transform 1D FIDs, 247
Fourier transform 2D data, 250
Fourier transform 3D FID into 3D data, 254
hypercomplex 2D Fourier transform, 253
imaginary part display, 155
interactive display, 41
interleave FIDs during processing, 310
left-shift FID to time-domain cursor, 614
make FID element using numeric text input, 363
move FIDs between experiments, 370
noise level measurement, 389
number acquired, 384
number of completed FIDs, 108
overhead delay between, 137
plot a scale under a FID, 466
plot in whitewash mode, 428
plot one or more FIDs, 443
prepare parameters for acqi display, 273
pulse breakthrough effects, 508
remove dc offsets, 146
retrieve from a file, 511
retrieve from experiment subfile, 513
save FID data in FDF format, 590
save in current experiment, 590
solvent subtraction, 414
start of FID display, 558
subtract FID from add/subtract experiment, 588
TPPI 2D Fourier transformation, 253
type of data processing, 461
vertical position, 640
vertical position of imaginary FID, 640
vertical scale, 633
weight and Fourier transform 1D FIDs, 654
weighting interactively, 663
width of FID display, 653
write numeric text file using a FID element, 660
zero-order phasing constant, 433
field of view for 2nd phase-encode axis, 348
field of view for phase-encode axis, 347
field of view for readout, 353
field position, 672
field-of-view
getting default, 267
setting to default size, 536
FIFO loop size, 234
Fifo Loop Size label, 116, 234
FIFO underflow error, 47
file, saving image planning to a, 524
files
append data to file, 660
automation data file name prefix, 87
Bruker data files for conversion, 579
clear a file, 659
delete, 150
delete one or more files, 507
display experiment library, 223
display in text window, 107
delete with user-selectable editor, 212
file name extension information, 268
find number of files in directory, 268
find words and lines in text file, 344
get text from data file, 272
handle interactively, 235
lines or records in file, 391
list files in directory, 163, 334, 353
load parameters from file into a tree, 246
make a copy, 123
make FID files using numeric text input, 363
making a copy, 124
move a file, 495
move file, 378
plot files, 638
plot text file, 451
print or plot to a file, 459
print text files, 469, 638
put text file into another file, 472
read 32-bit data files into VNMR, 579
read Bruker data files from tape, 485
remove old files from experiment, 112
rename a file, 495
rename file, 378
retrieve FIDs from file, 511
retrieve parameters from file, 512
retrieve shim coil settings from file, 512
return information from files display, 235
save FIDs in experiment, 590
save parameters from experiment, 591
save parameters from tree to a file, 246
save shim coil settings to a file, 592
text files display in graphics window, 208
transfer file from remote source, 215
transfer files to remote destination, 216
write formatted text to a file, 659
Files menu system
default directory, 149
filter bandwidth, 228
filter delays, 66
filter diagonalization method (FDM), 230
filtlib directory, 236
Find gzlvl1/gzwin button, 292
Find gzwin button, 292
fine attenuator, 618
fine attenuator configuration, 228
fine attenuator control, 187, 188, 618, 619
Fine Attenuator label, 117, 188, 228, 618
fine power attenuator, 187, 188
fine tuning readout gradient compensation, 288
finite impulse response (FIR) coefficients, 236
first delay in pulse sequence, 138
first point multiplier, 244, 245
first pulse width, 405
first-order baseline correction, 613
first-order phase, 345, 346
make zero, 105
first-order phase correction, 140
first-point multiplier, 110
fitspec.data file, 237
fitspec.inpar file, 237, 540
fitspec.outpar file, 237, 540, 562
fitting arrayed imaging data, 313, 601, 602
fixing/unfixing slice gap, 540
flag-type parameter, 127
flashc command, 239
Flexible Data Format (FDF), 229, 593
flip angle
list, 240
set rf power levels, 539
flip time, 405, 473
FLIPFLOP pulse sequence, 240
flow encoding gradient level, 274
fluorine
automated acquisition, 226
process 1D spectra, 227
fm-fm mode decoupling, 169
fm-fm modulation (swept-square wave), 174
folding-in problem, 228
foreground processing, 633
formatted text writing to a device, 658
formatting real number as a string, 243
four nucleus amplifier, 68
Fourier number, 241, 242
frequency limits of region, 271
frequency of a line, 269
gap between lines in spectrum, 263
gap mode, getting, 269
gap, fixing/unfixing slice, 540
GARP decoupling sequence, 169, 170, 174
gating time for receiver, 130, 508
gauss/cm, converting to DAC value, 234
Gaussian apodization constant, 663
Gaussian fraction, 199
golmms fracton for lineshape, 540
Gaussian function, 273, 274
Gaussian function shift, 663
Gaussian lineshape, 237
Gaussian low-pass filter, 235
Gaussian shift constant, 274, 275
gaussian time constant, see Gaussian function
Gaussian window function, 263
gc parameter, 264
gCOSY experiment, changing parameters for, 265
gCU (gradient compensation unit), 285
Gemini systems
convert data to VNMR, 120
deconvert files to UNIX format, 644
decompose files to UNIX files, 149
list contents of directory, 213
read tape, 603
general setup for 2D experiments, 533
generalized curve fitting to data, 220
generic automatic processing, 464
generating
a coronal overlay, 267
a sagittal overlay, 271
an active overlay, 267
an overlay based on scout image, 267
an overlay from saved parameters, 269
default field-of-view, 267
gap mode, 269
slice thickness, 267
slices, 267
transverse overlay, 272
gHMBC experiment, changing parameters for, 275
gHMQC experiment, setting up parameters for, 275
Index

gHMQCTOXY experiment, changing parameters for, 276
gHSQC experiment, changing parameters for, 276
gHSQCTOXY experiment, changing parameters for, 276
Gilson Liquid Handler window, 277
gilson.conf file, 213
global file, 127, 129
update after VNMR install, 628
global parameter tree
save parameters, 523
global-type parameter tree, 127
gmapz pulse sequence, 279, 280
gmapz.par file, 280
grad_cw_coef parameter, 429
grad_p_coef parameter, 429
grad_p1 array, 430
gradient
2nd phase encode increment, 283
3rd phase encode increment, 283
coil, 264
phase encode dephasing, 283
gradient amplifier installation tests, 606
gradient autoshimming, 278
gradient axis, 284
gradient calibration constant, 264, 291, 540
gradient calibration constant retrieval, 269
gradient calibration parameters
boresize, 98
gradient calibration pulse sequence, 465
gradient calibration value, 210
gradient coil configuration, 596
gradient coil configuration file, 117
gradient coil updating, 628
gradient COSY pulse sequence, 265
gradient evaluation pulse sequence, 261
gradient level trim, 290
gradient levels, 541
gradient list, 581
gradient map generation, 82, 83
gradient map generation, automatic, 82
gradient phase encoding increment, 282
gradient refocused high-speed imaging sequences, 676
gradient rise rate, 287, 621
gradient set
internal usable diameter, 98
gradient shape, 282
gradient shimming
display menu, 278
map shims, 278
pulsed field gradient strength, 292
set parameters, 278
spectral width percentage, 292
start acquisition, 278
start gradient autoshimming, 278
z-axis shims number, 292
Gradient Shimming System menu, 279
gradient spoiling time, 622
gradient step size, 285
gradient strength, 287
maximum value, 280
voxel selection, 291
X, Y, Z gradients, 291
gradient strength for each axis, 291
gradient strengths calibration for PGE, 429
gradient table generation, 128
gradient total limit, 290
gradients for X, Y, and Z axes, 285
Gradients label, 116
gradtables directory, 128, 541, 622
graphics window
display message with large characters, 96
display status, 285
display text file, 208
draw box, 98
write formatted text to screen, 659
Graphics Window colors, 535
graphics window, dividing into rows and columns, 542
GraphOn terminal window clearing, 112
gray scale contrast adjustment, 286
gray scale display adjustment, 286
gray scale image display, 311
gray scale image plot, 311
grid lines over 2D plot, 443
grid on a 2D display, 286
gripper abort, 46
Group A parameters, 160
group of parameters in tree, 543
H
H1.par file, 556
hardware Ethernet address display, 210
hardware shimming
list of shims, 298
hardware shims, 485, 534
hardware status of console, 310
hardware values in acquisition system, 544
hardware Z1 shimming, 298
HCCHTOCSY sequence, 297
height of peak, 244
HET2DJ pulse sequence, 299
HETCOR acquisition, 297
HETCOR experiment, changing parameters for, 299
HETCOR pulse sequence, 299, 446
HETCORCP1 pulse sequence, 299
HETCORPS pulse sequence, 299
heteronuclear 2D-J experiment, 299
heteronuclear chemical shift correlation, 299
heteronuclear J-resolved 2D spectra, 444
heteronuclear multiple-quantum coherence, 300, 301
heteronuclear Overbodenhausen experiment, 304
Hewlett-Packard plotter pens, 549
Hewlett-Packard plotters, 304, 665, 669
hiding a command, 300
hiding/showing
axes, 538
drawings in 3D, 537
filled polygon, 539
order of drawings, 538
high signal handling, 458
high-power pulse widths, calibrating, 81
Hilbert transform algorithm, 232
HMBC experiment, changing parameters for, 300
HMBC sequence, 446
HMQC experiment, changing parameters for, 300
HMQC phase-sensitive PFG pulse sequence, 276
HMQC pulse sequence, 275, 300, 446
HMQCR pulse sequence, 301
Index

HMQCTOCSY 3D pulse sequence, 301
HMQCTOCSY sequence, 301
HMQCTOXY experiment, changing parameters for, 301
HOHAHA experiment, 615
HOM2D pulse sequence, 302
HOMODEC experiment, changing parameters for, 302
Homodecoupler label, 118
homodecoupling control, 302, 303
homonuclear correlation, 124, 614
homonuclear decoupler present, 302
Homonuclear Hartmann-Hahn experiment, 615
homonuclear J-resolved 2D experiment, 302
homonuclear J-resolved 2D spectra, 445
homospoil, 304
pulse length, 306
pulses, 304
horizontal LC axis, 140, 418
horizontal offset, 302
horizontal projection of trace, 144
horizontally stacked spectra, 204
host computer
serial port connection to changer, 567
host computer connection to Ethernet, 539
host computer disconnect from Ethernet, 548
host disk errors, 47
hostname.le0 file, 539
hosts.3D file, 255
Hoult setting for final pulse times, 303
HSQC experiment, changing parameters for, 305
HSQC pulse sequence, 276, 304
HSQC-TOCSY 3D pulse sequence, 305
hypercomplex points to left-shift interferogram, 354

I

I1 and I2 values, 215
identifier, return for argument type, 624
idle mode for amplifiers, 67
IF Frequency label, 116
imag.c file, 291
image
annotate display, 311
center on the readout axis, 460
coordinate display information, 341
field of view size for readout, 353
position on 2D phase encode axis, 455
image planning
disabling, 388
saving as a milestone prescription, 523
starting/restarting, 583
ImageBrowser application, 100
ImageBrowser FDF files, 591
ImageBrowser program, 594
images
calculate 2D phasefiles, 312
display multiple images, 173
generate as ImageBrowser files, 591
save as ImageBrowser files, 591
imaginary part of FID, 155
imaging
application mode, 73
attenuator, 107
echo time, 606
intensity of excitation pulse, 617
macros and menus, 73
readout position, 375
imaging experiments
repetition time, 619
Imaging Gradient Coil label, 117
imaging gradients setup, 314
inactive parameter, 313
INADEQUATE data about 2-quantum axis, 242
INADEQUATE pulse sequence, 315
Incredible Natural Abundance Double-Quantum Transfer Experiment, 315
incremented delay for pulse sequence, 138, 139
index of experimental frequency of transition, 113
indirectly detected axis, 93
INEPT pulse sequence, 315
info directory, 534
info # file, 429
inline DSP, 200
Input board spectral width, 368
Insensitive Nuclei Enhanced by Polarization Transfer, 315
inserting a sample, 309, 317
inset spectrum, 317
integer-type parameter, 127
integral
display, 192
display mode, 318
integral value, 316
largest value in region, 317
normalization scale, 316
offset, 319
regions, 304
reset points, 671
scale, 319
set value, 545
integral amplitudes display, 182
integral amplitudes plot, 437
integral scale adjustment, 319, 320
integrals
clear reset points, 132
data truncation limit, 131
display in normalized format, 167
display list, 165
find integral values, 385
reset point amplitudes, 334
reset point frequencies, 335
integration, 1D spectrum, 318
intensity of spectrum at a point, 365
intensity threshold, 585
interactive acquisition display, 41
interactive phasing, 192
interactive probe tuning, 623
interactive UNIX shell, 560
Interactive View button, 439
interferogram coefficients, 227, 228
interferograms
first-point multiplier, 245
start of display, 558
type of data processing, 462
weighting interactively, 663
width of display, 653
zero-order phasing constant, 434
interlock to control lock level and spin speed, 314
Internet address, 43, 44
inverse cosine calculation, 40
inverse Fourier transform, 247, 248
inverse sine, 76
inverse tangent, 78
inversion prepulse recovery time, 612
inversion pulse intensity, 618
inversion pulse length, 435
inversion pulse shape, 436
inversion recovery experiments, 612
inversion recovery mode, 319
invert image, 312
ISIS, 639
iterated parameters list, 320
iterations in an iterative simulation, 385

J
J-correlation experiment, 615
joining
an existing experiment, 321, 322
joint arrays, 76
J-resolved 2D spectrum, 242
jump-and-return sequence, 323
JUMPRET sequence, 323

K
keyboard entries record, 491
keyboard focus to input window, 242
keyboard input into variables, 316
kinetics analysis, 70, 221, 326, 327, 427

L
label a stacked spectra display, 206
labeling an image display, 311
laboratory frame Overhauser experiment, 388
lastk file, 330
latching capabilities of frequency synthesizer, 331
Latching label, 117, 331
LC axis, 140, 418
LC-NMR
2D acquisition parameters, 332
add series of FIDs, 48
create parameters, 415
create pseudo-2D dataset, 278
delay for trigger, 208
display horizontal LC axis, 140
display LC-NMR parameters, 161
general 2D experiment setup, 333
set up parameters for LC-NMR sequences, 333
set up pulse sequence for LC-NMR run, 332
set up scout run, 553
TOCSY sequence, 333
least-squares curve fitting, 69, 220
left half of screen display limits, 333
left-shift FID to time-domain cursor, 614
left-shift ni interferogram, 354
left-shift ni2 interferogram, 355
left-shift np FID, 354
leg relay control, 334
lfs (low-frequency suppression) option, 413
limits for scales in regression, 526
limits of parameter in a tree, 546
limNET nodes database, 176
line amplitudes list, 339
line assignments for spin simulation, 111, 165
line broadening, 331, 332
line broadening factor, 664
line drawing between points, 189
line drawing capability, 374, 426
line frequencies, 339, 565
line frequencies and intensities
display list, 166
find values, 386
line in a text file, 344
line list plotting, 447
line listing intensity and frequency, 269
line narrowing sequence, 377
linear amplifiers, 68, 117
decoupler power level, 186, 187
power level, 617
linear curve fitting, 70
linear fitting to data, 221
linear modulator power, 188, 189, 619
linear monotonic order data, 602
linear prediction
algorithm, 346, 347
algorithm data extension, 350, 351
arraying parameters, 350
calculation start point, 586, 587
coefficients to calculate, 349
create parameters, 415
data extension, 348
data extension start point, 585, 586
multiple operations, 350
number of data points, 349, 350
output spectrum, 352, 353
parameter creation, 50
print output, 351, 352
printout, 350
type of data processing, 461, 462, 463
linear scaling of image intensity, 311
linearly spaced array values, 75
line-narrowing multiple-pulse, 100
lines of text, look up from a text file, 344
lines, showing, 537
lineshape modification, 540
linewidth for spin simulation, 566
linewidth measurement, 191
load time counter, 666
loadPrescription(char* path) command, 340
local file transfer to remote host, 216
local host name display, 210
local oscillator (L.O.), 344
localized spectroscopy, 639
localized spectroscopy experiments
repetition time, 619
location of sample in tray, 341
location to start a line, 374
lock
acquisition time constant, 342
automatic control, 66
automatic phase adjustment, 341
capture, 66
digital lock display, 487
frequency, 547
gain value, 343
interactive, 41
lock frequency adjustment, 342
lock parameters setup, 546
loop time constant, 343
Index

phase value, 343
power value, 343
read current lock level, 487
remove inactive lock, 626
solvent selection, 567
solvent used, 330
time constant, 343
lock file, 633
lock frequency
  track changes, 335
Lock Frequency label, 116, 118, 342
lock level display, 43
lock level interlock, 314
log file, 223
log file for experiment, 223
logarithm of a number, 339
login macro, 98
loop size of fifo, 234
looping control for real-time arrays, 547
looping processes control, 531
Lorentzian lineshape, 237, 540
low signal handling, 458
low-band amplifier, 68
lowercase format of string, 243
low-pass digital filter, 414
low-pass Gaussian filter, 235
lpanalyz.out.# file, 350, 351, 352

M
macelib directory, 98, 127, 150, 218, 360, 491
macros
  activated by VNMR bootup, 98
  automatic execution, 551
  before experiment starts, 262
  change action of abort command, 39
  check for existence, 218
  copy system macro to become user macro, 361
  copy user macro file, 358
  create without text editor, 126
  delete user macro, 150
  display dialog box, 162
  display system macro, 360
  display user macro in text window, 358
  display which macro is used, 657
  edit online description, 365
  edit user macro with vi editor, 361
  edit with macro editor, 359
  hide command with same name, 299
  keyboard entries, 491
  list system macros, 361
  list user macro file names, 359
  load macro into memory, 360
  name of invoking macro, 358
  name storage for macros, 382
  online description, 365
  real-value storage parameters, 483
  remove macro from memory, 472
  remove system macro, 361
  remove user macro from directory, 360
  restore normal abort function, 39
  return values to calling macro, 497
  terminate calling macro, 39
  terminate execution, 497
  user’s macro directory, 358
  magic angle spinning, see MAS
Magnet Leg Driver Board Configuration ID, 458
magnet leg relay control, 334
magnetization recovery, 138
main magnetic field strength, 95
Make Shimmap button, 279
makeuser command, 628
manual directory, 365
map shims, 278
MARK button, 366, 367
Mark button, 439
mark output, 628
mark, removing/activating, 548
mark1d.out file, 77, 366, 572, 628
mark2d.out file, 99, 366, 439
MAS cross-polarization spin-lock contact time, 119
MAS spinning speed, 578
Max. Decoupler label, 118
Max. Narrowband Width label, 116, 181, 368
Max. Spectral Width label, 116
Maximum DMF label, 116
maximum frequency of any transition, 566
maximum gradient DAC value, 285
maximum gradient strength for each axis, 291
maximum limits on a parameter, 346
maximum of two spectra, 575
maximum parameter value array, 416
maximum transients accumulated, 379
mean of the data in regression.inp file, 454
measured line frequencies, 565
measured line frequencies array, 572
memory buffers, write to disk, 241
memory increased by removing macros, 472
memory map FID file, close, 371
memory map open FID file, 372
memory usage statistics, 378
Menu On button, 369
menulib directory, 235, 369
menus
  button command string, 378
  change status of menu system, 369
  edit menu with vi editor, 369
  label for button, 374
  menu displayed by Return button, 331
  path to user’s menu directory, 369
  return currently active menu, 383
  select menu without activation, 382
MERCURY
  broadband channel tuning, 101
  console type, 119
  probe tuning mode, 624
MERCURY series
  broadband channel tuning, 101
MERCURY-VX
  probe tuning mode, 624
message
  confirm using mouse, 119
  display with large characters, 96
  messages from send2Vnmr, 335
method string, 584
microimaging
  center sequence calibration, 535
  ECC tool window, 210
  eddy current compensation analysis, 211
  eddy current compensation data, 210
  field of view for phase encode, 347
  generate transverse magnetization, 676
Index

gradient amplifier installation tests, 606
gradient calibration constant, 540
intensity of an inversion pulse, 618
inversion pulse shape, 436
move data into reference table, 210
orientation of slice plane, 397
phase encoding, 283
phase encoding gradient increment, 282
refocusing pulse shape, 474
shape of excitation pulse, 406
shaped gradient tests, 289
update eddy current settings, 211

Milestone
parameters, get overlay from, 269
prescription, saving current planning as a, 523
minimum frequency of any transition, 566
minimum intensity threshold, 585
minimum limits on a parameter, 546
minimum of two spectra, 575
minimum parameter value array, 416
MLEV-16 decoupling sequence, 169, 170, 174
mode for n-dimensional data display, 619
modulation frequency of decoupler, 169, 170
modulation mode for decoupler, 173, 174, 175
mopos parameter, 620
mouse
confirming a message, 119
mouse position, reporting, 277
moving
files, 378, 495
parameters between experiments, 376
spectral window according to cursors, 376
transmitter offset, 376
MQCOSY pulse sequence, 377
MREV8 pulse sequence, 132, 377
multidimensional data display mode, 619
multiecho sequences, 382
multihost processing, 256, 257
multiple image display, 173
multiple receivers
add transformed data files with weighting, 507
combine data, 52
number currently active, 382
number of receivers, 392
set filter bandwidth, 377
set gain, 377
weighting for different receivers, 484
which receivers to use, 484
multiple-pulse line narrowing, 100, 377
multiple-quantum filtered COSY, 377, 615
multipulse experiments
f1 scaling factor, 527
scaling factor, 526
multislice experiments, 591
spin-echo imaging sequence, 531

N
name of pulse sequence, 532
name storage for macros, 382
natural logarithm of number, 339
negative intensities, setting 2D, 676
ni interferogram
number of complex point to left shift, 354
type of data processing, 462
ni2 interferogram
type of data processing, 462
zero-order phasing constant, 434
NMR resonance offset frequency, 497
node files, 215, 216
nodes file, 215, 217
NOE difference experiment, 388
NOE experiment, 132
NOESY
parameter set, 281
plotting spectra, 442
pulse sequence, 388
NOESY experiment, changing parameters for, 388
NOESY1D experiment, changing parameters for, 389
noise level estimate, 271
noise level in spectrum, 390
noise level of FID, 389
noise modulation, 174
noise multiplier, 389
normalized integral amplitudes, 183
normalized integral amplitudes plot, 437
normalized integrals display list, 167
normalized intensity mode, 387
nt array, 430
N-type display, 392
nucleus for decoupler, 175, 176
nucleus for observe transmitter, 614
nucleus selection, 556
nucleus to add to probe file, 614
nuctables directory, 175
number of increments of evolution time, 384, 385
Number of RF Channels label, 116, 393
Nyquist frequency, 310

O
object library for PSG, 467
oblique imaging capability, 531
observe nucleus transmitter frequency, 558
offset
  horizontal, 302
  integral, 319
  vertical, 639
offset frequency
calculate for nucleus and ppm, 549
online description of command or macro, 365
edit description, 365
open reel tape, 604
oph real-time variable, 124
order of parameter array, 75
orientation of slice plane, 397
out.c file, 122
overhead delay between FIDs, 137
overlay
  getting a coronal, 267
  getting an active, 267
  getting based on scout image, 267
  getting from saved parameters, 269
  getting sagittal, 271
  getting transverse, 272
  redrawing, refreshing an, 492
  redrawing/refreshing an, 492
  showing as center lines/stripes, 163
  showing as stripes, 165
oversampling
overrange of frequency synthesizer, 398
oversampling
Index

bandwidth, 397
factor for acquisition, 399
filter type, 398
number of coefficients, 397
parameter creation, 50, 416
setting parameters, 375
Oxford shim supply, 560
Oxford VT controller, 367
Oxford-Sorenson VT controller, 367

P

page change on plotter, 410
par directory, 556
parameter array, 534
parameter directory
delete, 150
parameter list
parameter names and values, 410
plotting, 455
power level parameters, 475
pulse length parameters, 447
pulse template parameters, 418
parameter screens display menu, 161
parameter set, converting to APT experiment, 73
parameter sets
correct saved parameter sets, 627
file name of retrieved set, 234
update all sets in directory, 627
parameter tree
copy parameters of group, 288
create new parameter, 127
destroy parameters of a group, 153
display parameters with attributes, 164
limits of parameter, 546
load parameters from file into a tree, 246
make parameter active, 396
make parameter inactive, 395
prune extra parameters, 466
remove a parameter, 153
set Dgroup of a parameter, 538
set group of parameter, 543
set values of string parameter, 538
systemglobal-type tree, 115
types of trees, 127
value of parameter, 556
write parameters to file, 246
parameter values, setting, 557
parameters
3rd rf/3D parameter group, 159
4th rf channel parameter display group, 159
acquisition/processing group, 158
add for FID display, 234
add parameter to probe file, 51
add to current experiment, 49
adjust values from setup macros, 239
adjusting, 322
adjusting plot, 322
arraying order and precedence, 75
automation parameter group, 161
basic experiment setup, 555
boxed for plotting, 100
center sequence calibration, 535
change type, 555
check existence, 218
chemist-style, 100
convert to PGE, 428
copy between trees, 288
correct limits and step sizes, 414
correct parameter characteristics, 238
create 2D parameters, 411
create 3D parameters, 411
create 4D acquisition parameters, 412
create for fourth channel, 238, 239
create for linear prediction, 415
create for third rf channel, 238
create LC-NMR parameters, 415
create new parameter in tree, 127
create oversampling parameters, 416
create parameters for 2D peak picking, 415
create solvent subtractions parameters, 413
customize parameter sets, 629
destroy a parameter, 153
destroy parameters of a group, 153
display control, 71
display from tree with attributes, 164
display parameters group, 159
display templates for third rf channel, 412
display values in text window, 211
displaying value, 479
downsampling, 413
edit parameter and its attributes, 412
full recall of display parameters, 245
full spectrum display, 226
get value, 273
gradient shimming, 278
limits of parameter in tree, 546
linearly spaced steps, 75
list to be iterated, 320
lock parameters setup, 546
make parameter active, 396
maximum values, 416
minimum values, 416
move between experiments, 376
move display parameters between experiments, 368
move parameters to target experiment, 619
plot list automatically, 71
plot on special chart paper, 304
prepare for acqi, 273
print all, 70
protection mode, 550
prune parameters from tree, 466
pseudo-echo weighting, 467
pss0, 469
radialAngles, 484
read from file and load into tree, 246
recall display parameter set, 483
reset after partial 3D FT, 496
resolution enhancement, 496
restoring current experiment, 322
retrieve from experiment subfile, 513
retrieve from file, 512
retrieve individual parameters from file, 513
retrieve parameter from probe file, 269
save display parameters as set, 521
save from experiment, 591
save from tree to file, 246
save parameters from global tree, 523
set group of parameter in tree, 543
set up for pulse sequences, 467
set up standard two-pulse sequence, 522
set voxel parameters, 640
shaped gradients testing, 289
shims parameter group, 161
sine window function, 563
sinebell weighting, 564
sine-squared window function parameter values, 564
spin simulation parameter arrays, 165
spin simulation parameter group, 160
spin system parameters to iterate, 315
step size values, 417
system configuration, 114
test state of parameter, 396
turn off active parameter, 395
types of values, 127
unit conversion, 625
update after new VNMR install, 628
value of parameter in tree, 556
VAST experiment parameter setup, 631
version of parameter set, 417
parlib file, 467
paths
2D planes from a 3D data set, 417
current working directory, 473
user’s macro directory, 358
user’s menu directory, 369
user’s shim settings directory, 562
VNMR system directory, 597
VNMR user directory, 629
Pbox
add parameter definition to pbox.inp file, 422
assign Pbox calibration data, 476
convert to Pbox default units, 423
converts to default units, 423
create Pbox shape file, 125
create shape definition, 478
define excitation band, 420, 530
define excitation band for solvent suppression, 420
display interactive modulation pattern, 197
display interactive pulse shape, 197
display last generated pulse shape, 197
display modulation pattern, 196
display pulse shape, 196
extract dmf value, 420
extract dres value, 420
extract fine power level, 421, 422
extract name of last shape, 421
extract power level, 421
extract pulse length, 421
generate a single-band shapefile, 477
open shape definition file, 396
plot modulation pattern, 468
plot pulse excitation profile, 457
plot pulse shape, 468
plot the last created pulse shape, 468
print pulse header, 456
reset temporary pbox/Vnmr variables, 423
simulate Bloch profile for a shaped pulse, 477
write a wave into file, 472
write wave definition string, 557
pbox
write wave definition string, 557, 559
pbox shape file, 125
pess.outpar storage file, 178, 425
peak frequencies display, 182
peak frequencies plot, 455
peak frequencies threshold, 610
peak height or phase measurement, 244
peak heights comparison, 53
peak noise, 271
peak number, 333
peak picking, 336
diagonal peak threshold, 666
parameters creation, 415
plot results, 447
peak printout threshold, 611
peak search range of data points, 391
peak truncation in spectra plot, 437
peak width of solvent resonances, 567
peak, selecting, 589
peaks.bin file, 338
peak-to-peak noise, 390
peaks
maximum number to use, 368
on HP plotter, 549
selection for drawing, 426
Performa I, II, III, 117
Performa modules, 285
PFG
absolute-value MQF COSY pulse sequence, 281
absolute-value ROESY pulse sequence, 290
amplifiers on/off control, 428
eddy current testing, 286
gradient calibration constant, 540
HMQC phase-sensitive pulse sequence, 276
HMQC pulse sequence, 275
HSQC pulse sequence, 276
NOESY parameter set, 281
selective excitation pulse sequence, 531
sequence for PFG testing, 406
TNNOEESY pulse sequence, 290
pge file, 429
PGE pulse sequence
   calibrate gradient strengths, 429, 430
   extract data, 429
   parameter conversion, 428
   plot results, 429
   print results, 429
   processing of data, 429
phantom for gradient calibration, 540
phase angle display mode, 172, 407
phase correction applied to interferogram, 140
phase cycling type, 432, 433
phase encode
   gradient levels, 541
   image center position, 455
   pulse length, 616
phase encode dephasing gradient, EPI sequence, 283
phase encode gradient increment multiplier, 284
phase encoding, 283
phase encoding gradient increment, 282
phase encoding gradient pulse length, 616
phase file
   display in experiment, 147
   phase of first pulse, 433
   phase of peaks, 244, 432
   phase parameters
      automatic calculation of, 71
      phase-correction angles, 345, 509, 510
Index

phased data display mode, 172
phased spectra display mode, 408, 430, 431
phase-encode axis, 347, 348
phasefiles, 312, 313, 314, 592
calculate 2D phasefiles, 312
copy stored phasefile, 512
transform and save images, 364
phase-sensitive 2D transformation, 346
phase-sensitive COSY pulse sequence, 124
phase-sensitive data, 249, 253, 655, 656
phasing
automatic, 71
control update region, 433
phosphorus
acquisition, 406
processing, 407
spectrum plotting, 451
\pi/3 shifted sinebell squared window function, 435
\pi/4 shifted sinebell squared window function, 436
pixel size calculation, 495
pl2dj macro, 448
planes
extract from 3D spectral data, 270
planes directory, 313, 512, 592
planlock parameter, 439, 581
planner lock, 440
planning a target scan, 439
Plot Design, joining, 321, 322
plot parameters
adjusting, 322
plot queue
show jobs in queue, 562
stop jobs and remove from queue, 325
plots, 429
plotter
characteristics, 550
device setup, 449
display mode, 451
Hewlett-Packard, 665
maximum number of pens, 368, 549
maximum width of plotting area, 650
plot contours, 424
reinitializing, 325
resolution of points drawn, 457
show plot queue, 562
stopping plot jobs, 325
submit plot and change plotter page, 410
write formatted text to plotter, 659
plotter units
converted from Hz or ppm, 306
Plotters color, 535
plotting
2D contour plots for 3D planes, 450
2D displayed resolution, 76
2D peak picking results, 447
2D spectra in whitewash mode, 438
adjust plot parameters, 237
arrayed 1D spectra, 441
ATP-type spectra, 440
axis label units, 92
boxed parameters, 100
carbon spectrum, 441
color assignments, 113, 549
contour plot with colors, 424
contours display, 181, 182
COSY data set automatically, 408
COSY spectra, 442
deconvolution analysis, 443
DEPT analysis, 409
DEPT data, 442
display same as plot, 664
draw box, 98
exponential curves, 427
FIDs, 443
FIDs in whitewash mode, 428
files, 638
formatted text, 658
grid on 2D plot, 443
heteronuclear J-resolved 2D spectra, 443
homonuclear J-resolved 2D spectra, 445
horizontal LC axis, 418
limit to center of page, 109
time list, 447
NOESY spectra, 442
non-arrayed 1D spectra, 448
noninteractive gray scale image, 311
parameter list, 410, 455
parameter list on special paper, 304
parameters automatically, 71
peak frequencies over spectrum, 455
PGE calculated results, 429
phosphorus spectrum, 450
plot a title, 613
plotter characteristics, 550
polynomial curves, 427
proton spectrum, 444
pulse sequence, 457
scale below spectrum or FID, 466
set full page plot with room for traces, 258
set limits for full page plot, 258
spectra, 437
spectra automatically, 448
spectra in whitewash mode, 452
spectral expansion, 52
start of plotting position, 526
start of plotting position in second direction, 526
text file, 451
X,H-correlation 2D spectrum, 446
plotting area, see chart
plotting scaling factor, 306
point-by-point, 575
pointer position, locating, 277
polarization transfer experiments, 152
polygon, showing/hiding filled, 539
polynomial curve, 220
polynomial curves display, 222
polynomial curves plot, 427
polynomial fitting of baseline, 96
Postscript printer, 665
post-trigger delay, 302
powder pattern
finding the center, 575
power data display mode, 172
power level calibration, 617
power level for decoupler with deuterium decoupler, 187
power level for decoupler with linear amplifier, 186, 187
power level of transmitter, 617
power spectra display mode, 474, 475
power, setting, 550
powers of 2 vertical scale adjustment, 642
ppm calculations, 492
ppm of solvent resonances, 567
preacquisition delay, 409
preamplifier signal level selection, 458
precedence of parameter array, 75
PRESAT sequence, 458
pre-trigger delay, 484
print queue
  show jobs in queue, 563
  stop print jobs and remove from queue, 326
printcap entry, 638
printer
  device setup, 459
  linewidth resolution, 457
  maximum width of chart, 650
  resolution in dots/mm, 457
  send text to printer, 459
  start print operation, 459
  stopping print jobs, 326
type, 638
  write formatted text on printer, 660
printing
  color assignments, 113, 549
  parameters, 70
  PGE calculated results, 429
  probe file after autocalibration, 40
  starting, 459
text file, 469
text files, 638
printing area, see chart
probe
  phase glitch removal, 240
  tuning, 479, 623
  tuning frequencies, 623
  tuning mode on MERCURY, 624
type, 460
probe directory, create new, 51
probe file, 40, 49, 549
  add parameter, 51
  retrieve parameter, 269
  set decoupler parameter values, 536
  update, 627
probe file, copying, 85
probe file, create new, 51
probe file, make copy, 85
probe gcal calibration macros, 82
probe protection control, 461
probe, copying, 85
probe, editing, 460, 461
procdat file, 533, 534
processed-type parameter tree, 127
processing
  1D carbon spectra, 105
  2D spectra, 463
  3D data processing information, 533
  arrayed 1D spectra, 463, 583
  create 2D parameters, 411
  create 3D processing parameters, 411
  DEPT spectra array, 153
  FIDs automatically, 464
  fluorine 1D, 227
  generic automatic, 464
  interleave FIDs, 310
  phosphorus 1D spectra, 407
  proton 1D, 295
select 1D experiment for processing, 266
selected 2D experiment, 266
simple 1D spectra, 462
solvent subtraction events, 414
processing mode for 2D data, 453
processing on FID, 461
processing on the interferogram, 462
processing parameters group, 158
procpar file, 114, 127, 627
procpar3d file, 534
procpar3d parameter set, 384, 533
programmable pulse modulation, 174
project 2D data onto axis, 465
projection plane, 185
protection mode of parameter, 550
proton
  acquisition, 296, 297
  automatic acquisition, 294
  pulse power level, 456
  spectra processing, 295
  spectra vertical scale adjustment, 643
  spectrum plotting, 444, 451
proton acquisition, 296, 298
proton chemical shifts spectrum calculating, 178
  calculating and showing, 425
  reducing to a list, 178
proton decoupler
  pulse calibration, 455
  proton decoupler calibrations, 83
  proton frequency configuration, 295
  Proton Frequency label, 116, 117, 295
  proton gradient ratio calibration macros, 81
  proton observe calibration macros, 83
  proton parameter set, getting, 80, 81
  pseudo-2D, 632
  pseudo-2D dataset, 278
  pseudo-echo weightings, 467
  psg directory, 532
  PSG errors, 467
  PSG message, 38
  PSG object library compilation, 467
  psg.error file, 467
  psglib directory, 532
  pss0 parameter, 469
  PTS frequency synthesizer, 117, 470
  P-type diagonal, 242
  P-type double-quantum axis, 242
  pulse amplifier
    mode, 67
      phase glitch removal, 240
      pulse breakthrough effects, 508
      pulse calibration data file
        update and display, 470
      pulse interval time, 476
      pulse length of decoupler, 454
      pulse power for shaped pulse, 471
      pulse power level, 456
        parameter list, 475
  pulse sequence
    compiling, 532
    display diagram, 185
    initiate compilation, 532
    label for screen, 469
    name to be used, 532
    phase-sensitive COSY, 124
Index

plotting a picture of a sequence, 457
set up parameters, 467
setup macro, 282
Pulse Sequence Controller board, 509
pulse sequence generation, see PSG
pulse sequences
  display templates, 555
  pulse template parameter list, 418
  pulse width in degrees, 405, 473
  pulse width length, 473
  pulse width of first pulse, 405
  pulse width optimum value, 216
pulscal file, 470, 539
pulsed field gradient strength, 292
Pulsed Field Gradients label, 118
pulse-type parameter, 127
pulsewidth, setting, 550
pure absorbptive display, 253
pwet pulse width, 652
pwxl parameter, 476

Q
quadratic fitting to data, 70, 221
quadrature detection
  frequency shifted, 247
quadrupole echo pulse sequence, 579
question mark (?) notation, 479

R
radial slice fan angle parameter, 484
radial/Angles parameter, 484
ratio parameter, 508
readout compensation gradient, 288
readout field of view, 353
readout gradient compensation, fine tuning, 288
readout gradient setting, 542
readout gradient strength, 287
readout image center position, 460
readout position, 375
real Fourier transform, 461, 462
real number formatted into string, 243
real number, returning square root of a, 578
real numbers, truncating, 622
real scan repetition, 391
real variable
  create real variable without value, 489
format as string, 243
real-time digital filters, 398
real-time DSP (digital filtering), 200
real-type parameter, 127
real-value storage for macros, 483
receiver
  channel imbalance, 125
gain, 262, 623
gating time, 130, 508
overflow warning, 45
receiver option, 200-kHz, 181
recording current window activity, 323
recording keyboard entries, 491
REDOR pulse sequence, 491
redrawing an overlay, 492
redrawing overlays, 492
reference deconvolution, 231
reference frequency
  position, 493
reference line
  clear referencing, 129
  frequency, 502, 503
  position, 502
  reference frequency, 492
set line, 505, 506
reference peak, see reference line
reference spectrum to TMS, 614
refocus pulse width, 406
refocusing gradient for slice selection, 290
refocusing pulse shape, 474
refresh command, 492
refreshing an overlay, 492
refreshing overlays, 492
regions
  divide spectrum into regions, 494
  find tallest peak, 425
  frequency limits of specified region, 271
  in spectrum, 392
  plot expansions, 52
  selection, 304
region-selective 3D processing, 469
regression analysis data input, 505
regression mode, 69
regression mode curve fitting, 220
regression scale limits, 526
regression.inp file, 222, 454, 505
RELAY-COSY pulse sequence, 495
RELAYH pulse sequence, 495
release procedure, 378
remote file transfer to local host, 215
remote machine name, 43, 44
removeAstack(int index) command, 495
removing
  a stack, 495
  all stacks, 112
dc offsets, 146
directories, 507
files, 507
selected slice, 151
selected stack/slice, 150
user macro, 360
removing/activating a mark, 548
renaming a command, 300
renaming files, 378, 495
reset points for integrals, 132
resetting acquisition computer, 39
resolution enhancement function, 331, 332
resolution enhancement parameters, 496
resolution equalization, 76
resolution on printers and plotters, 457
resonance offset frequency, 497
restarting/starting image planning, 583
restoreStack() command, 497
restoring
  a stack, 497
restoring a stack, 497
retrieving
  active overlays, 267
default field-of-view, 267
FDIs, 511
gap mode, 269
parameters, 512
slice thickness, 267
Index

slices, 267
Return button selection of menu, 331
reverse INEPT, 304
rf band in use, 498
rf channel selection, 499
rf channel type, 500
rf channels available, 393
rf channels frequencies, 539, 570
rf generation type, 504
rf power for desired flip angle, 539
rf pulse calibration identity, 501
rf pulse shape analysis, 471
rf pulses setup, 314
rf waveform generator, 505
ridges in FID display, 244
right half of screen display limits, 505
right phase parameter, 509
right phase-correction angles, 509
ROESY experiment, changing parameters for, 508
ROESY parameter set, 290
ROESY pulse sequence, 508
root-mean-square noise, 271, 390
rotating frame NOE experiment, 615
rotating frame Overhauser experiment, 508
rotational echo double-resonance, 491
rotor speed display, 305
rotor synchronization, 305
configuration parameter, 509
spinning rate, 579
Rotor Synchronization label, 116, 306, 509
RS-232 cable, 45
running FDM program, 230

S
s2pul3rf parameter set, 412
sample
change for acquisition, 111
ejection from probe, 210, 212
insert in probe, 309, 317
location of samples in tray, 341
spin rate, 579
submit change sample experiment to acquisition, 522
temperature, 409, 607
sample changer
automation data file prefix, 87
automation mode active, 84, 85
automation run preparation, 83
change sample experiment, 111
comm port, 116
controlling, 84
controlling macro for automation, 84
errors, 46
last lock solvent used, 330
resume suspended automation run, 88
serial port, 118
serial port connection, 567
starting automation run, 85
status window, 583
suspend automation run, 89
tray size, 620
Sample Changer label, 116, 118, 620
Sample Changer Serial Port label, 567
sample information for automation run, 213
Sample Management System serial port connection, 567
sample tray size, 116, 118
sampleinfo file, 84, 87
saving
current planning as milestone, 523
current planning to a file, 524
data, 589
digitally filtered FIDs, 163
display parameters, 521
experiment data to subfile, 594
FID data in FDF format, 590
FIDs in current experiment, 590
files using a base name, 523
images as FDF files, 593
images as ImageBrowser files, 591
images as phasefiles, 364
parameters from current experiment, 591
parameters to file, 246
phasefile in current experiment, 592
shim coil settings, 592
text file into another file, 472
scale below spectrum or FID, 194, 466
scale limits in regression, 526
scale spectral width, 526, 527
scaling constant, 379
scaling factor for multipulse experiments, 526
scaling factor for plots, 306
scaling factors, 92
scan in progress, 131
scout experiment, 620
scout run, 553
scout scan repetitions, 391
scopy parameter, 620
screen coordinates, translating, 621
screen display set for center, 109
screen distance, translating, 621
SCSI errors, 47
second decoupler
acquisition parameters, 159
adjust tip-angle resolution time, 171
decoupler mode, 168
decoupling sequence, 195
fine power attenuator, 188
frequency, 155
frequency offset array, 528
frequency offset control, 177
homodecoupling control, 303
linear modulator power, 189
modulation frequency, 169
modulation mode, 174
nucleus lookup, 175
power level with linear amplifier, 186
pulse sequence diagram, 185
set frequency to cursor position, 528
tip-angle resolution, 191
second delay, 138
seeing
drawings in 3D, 537
overlay as center lines/stripes, 163
overlay as stripes, 165
stripes/lines, 537
selected widths, setting, 554
selective excitation experiment, continuing, 589
selective excitation pulse sequence, 531
selective frequencies, setting, 554
Index

selective inversion, setting up, 555
send command to VNMR, 531
Seq label on screen, 469, 532
seqfil file, 185
seggenmake file, 532
seqlib directory, 224, 532
serial port connection, 567
serial port for sample changer, 118
Set colors for Graphics Window, 535
set colors for Plotter, 535
Set Default button, 149
Set Params button, 566
setting
default number of slices, 536
default slice thickness, 537
default type, 537
field-of-view to default size, 536
parameter values, 557
setup experiment, 587
setup macros, 239
s/f w button, 558, 653
shape of an excitation pulse, 406
shape of refocusing pulse, 174
shaped gradients tests, 289
shaped observe excitation sequence, 559
shaped pulse analysis, 471
shaped pulse calibration, 95, 471
shapeinfo file, 95, 471
shapelib directory, 95, 195, 196, 471, 476
shared amplifier type, 68
shell on UNIX, 559, 560
shift of stack center, 469
Shifted Laminar Pulses (SLP), 332, 566
shifted sinebell squared window function, 435, 436
shim coil settings, 488
retrieval from file, 512
save to file, 592
shim gradient, 665, 666, 667, 669, 670, 672, 673, 674, 675, 676, 677
shim method string creation, 383
shim method string display, 197
shim parameters group, 161
shim set type, 560
shim settings directory, 562
shim supply, 560
shim values comparison, 163
shim values used, 340
shimmap calculations, 292
shimmethods directory, 197, 370
shimming
automatic shimming conditions, 661
AutoShim method, 370
errors, 46
interactive, 41
Z1 hardware, 298
shims
list of, for hardware shimming, 298
read all shims, 485
set all shims, 534
shims directory, 513, 562, 592
Shimsset label, 116, 560
Show Target button, 190, 439
showing
overlay as center lines/stripes, 163
overlay as stripes, 165
stripes/lines, 537
showing/hiding
axes, 538
filled polygon, 539
intersection, 537
order of drawings, 538
sidechain assignments, 297
signal-to-noise ratio, 567, 608
estimate, 271
improvement, 236
maximum, 199
measurement, 198
sine value of angle, 563
sine window function values, 563
sinebell constant, 524
sinebell shift, 664
sinebell shift constant, 525
sinebell squared window function, 435, 436
sinebell time period, 664
sinebell weighting parameters selection, 564
sinebell-squared window function, 578
sine-squared window function, 564
single-voxel spectroscopy experiments, 639
SIS (12 bit) gradients, 117
SISCO Imager console type, 119
skyline projection, 465
Slice button, 439
slice gap, fixing/unfixing, 540
slice gradient levels, 543
slice parameters, 581
slice parameters set for target slice, 565
slice plane orientation, 397
slice position, 469
slice selection fractional refocusing, 289
slice selection gradient level, 290
slice selection gradient strength, 289
slice selection refocusing gradient, 290
slice thickness, 612
getting, 267
setting default, 537
slices
alternating, 67
deleting selected, 150, 151
getting, 267
setting default number of, 536
slices to be acquired, 391
slice-selective excitation pulse, 676
software preparation date, 498
software revision level, 497
solids
adjust tau2 to current cursor position, 622
cross polarization spin-lock experiments, 70
cross-polarization spin-lock analysis, 221
echo pulse sequence, 579
f1 spectral width scaling factor, 527
first pulse phase, 433
MREV8 multiple-pulse experiment, 377
multiple-pulse line narrowing, 132
rotor speed display, 305
rotor synchronization module, 509
scaling factor for multipulse experiments, 526
solid-state echo pulse sequence, 579
solid-state HETCOR sequence, 299
spinning speed for MAS, 578
VT controller in use, 367
solvent resonances ppm and peak width, 567
solvent selection, 556
solvent subtraction
create parameters, 413
filter bandwidth for filtered FID, 580
order of polynomial to fit digital filtered FID, 581
parameter creation, 50
solvent suppression, 251
solvent table information, 568
solvents file, 556, 567, 568
solvent-suppressed region, 580
spatial resolution calculation, 495
spectra
3D de correction, 570
absolute value display mode, 90
add spectrum to add/subtract experiment, 569
APT plotting, 440
automatic 1D integrate, 318
automatic phase, 71
automatic phase adjustment, 71
center cursor, 109
data truncation limit, 131
deconvolution, 237
delete spectra from $T_1$ or $T_2$ analysis, 151
display calculated spectrum, 199
display scale, 194
display single spectrum, 192
divide spectrum into regions, 494
drift correction calculation, 142
drift correction parameters, 107
extract planes from 3D spectral data, 270
find gap in spectrum, 263
find peak heights or phases, 244
find tallest peak in region, 425
fold COSY-like correlation spectra, 242
fold J-resolved 2D spectrum, 242
frequency shift of spectrum, 355, 356
frequency-independent phase, 432
full display, 226
horizontal offset of each spectrum, 302
inset spectrum display, 317
integral amplitudes display, 182
integral amplitudes plot, 437
integral regions, 304
intensity of a spectrum at a point, 365
interactively display, 41
move cursor to center, 109
move spectral window according to cursors, 376
noise limit, 390
normalized integral amplitudes display, 183
normalized integral amplitudes plot, 437
normalized intensity mode, 387
number of regions, 392
offset of integral, 319
peak frequencies display, 182
peak height comparison, 53
peak search range of data points, 391
phase adjustment, 71
phase angle display mode, 408
phased display mode, 407, 430, 431
phosphorus processing, 407
phosphorus spectrum plot, 450
plot a scale under a spectrum, 466
plot COSY automatically, 442
plot NOESY automatically, 442
plot one or more spectra, 437
plot peak frequencies, 455
plot spectra in whitewash mode, 452
power spectra display mode, 474, 475
processing simple 1D, 462
proton spectrum plotting, 444
reduce spectral width to minimum, 374
reference line frequency, 502
reference line position, 502
reference to TMS, 614
rotate 2D spectrum, 508
select spectrum without displaying, 529
signal-to-noise estimate, 271
signal-to-noise test, 608
solvent-suppressed region, 580
spectral integral display and plot, 318
stacked spectra display, 202, 203, 204, 206
subtract spectrum from add/subtract experiment, 576
threshold for integrating 2D peaks, 611
total width to be acquired, 594
update region during phasing, 433
vertical offset in stacked display, 639
vertical position, 640
vertical scale, 641
whitewash mode display, 207
spectra in 2D data set, rearrange, 605
spectra in a 2D data set, rearrange, 605
spectra, taking maximum of two, 575
spectral expansion automatic plot, 52
spectral subtraction, 576
spectral width, 594, 595, 596
Input board, 368
percentage for gradient shimming, 292
reduce to minimum, 374
set for given spectral window, 553
set in 2nd evolution dimension, 554
set in evolution dimension, 554
spectrometer proton frequency, 295
spectrometer system configuration, 596
spectrum, plotting on side, 449
spectrum, plotting on top, 449
spectrum, plotting on top and side, 450
spin automation hardware, 573
spin rate of sample, 571
spin setup experiment, 571
spin simulation
clear line assignments, 111
deconvolution start point, 628
display group of parameters, 160
index of experimental frequency, 113
intensity threshold, 585
linewidth, 566
maximum frequency of any transition, 566
measured line frequencies, 565
measured line frequencies array, 572
minimum frequency of any transition, 566
number of iterations, 385
parameter arrays, 165
parameters to be iterated, 320
perform spin simulation, 574
set parameters to iterate, 315
spin system entry, 576
transition amplitude, 111
transition frequency, 113
Index

transition number, 111
  using deconvolution as input, 237
vertical scale, 593
spin speed interlock, 314
spin system
  enter values for spin simulation, 576
  restoring to before last run, 625
spin-echo diffusion imaging sequence, 529
spinto file, 625
spino file, 165
spin.data file, 625
spin-lock contact time, 119
Spinner Control window, 573
spinner errors, 45
spinner speed display, 43
spinning speed for MAS, 578
spin.list file, 574
spins.outdata file, 199
Spinsight data, 511
spinsys directory, 315
spline fitting of baseline, 96
split difference between two cursors, 575
spoiler gradient level, 289
spoiling time for gradient, 622
square root image, 312
square root, returning, 578
square wave mode decoupling, 169, 174
SSECHO pulse sequence, 579
SSECHO1 pulse sequence, 579
stack
  removing a, 495
  restoring a, 497
stack center shift, 469
  along z axis, 469
stacked display width, 649
stacked FIDs, 156, 157, 158
stacked plot of 2D spectra, 193, 438
stacked spectra display, 202, 203, 204, 206
stacked spectra horizontal offset, 302
stacking control, 583
stacking mode, 582
stacks
  adding a, 48
  deleting all, 112
  deleting selected, 150
  removing, 495
  restoring, 497
standard 2-pulse sequence, 405, 522
standard application mode, 73
standard deviation of input, 91
standard flip angle list, 240
start of FID display, 558
start of interferogram, 558
start of plot, 569
starting
image planning, 267
  image planning with previous stacks, 271
  interactive image planning, 267, 269, 271, 272
VNMR directly, 636
  VNMR from UNIX, 636
starting Plot Designer, 321
starting/restarting interactive image planning, 583
startup macro, 101
static binding, 533
static magnet field value, 95
stdpar directory, 556
steady state pulses, 579
STEAM, 639
Step Size label, 117, 616
step size parameter value array, 417
stimulated echo technique, 584
stopping acquisition, 522
stored data, style, 120
stored queue, 85
streaming tape, 604
strength of pulsed field gradient, 292
string
  format for output, 243
  length in characters, 334
  select substring, 588
  text window display, 211
string parameter values, 538
string variable creation, 585
string-type parameter, 127
stripes, showing, 537
subfile, 125
substring selected from a string, 588
subtracting zero-frequency components, 413
sum of input, 91
sum of squares of input, 91
sum/difference spectrum, 49
summing projection, 465
sum-to-memory error, 47
Sun display, clear window, 112
swept-tune graphical tool, 479
switchable probe caution, 161, 186, 187
synchronous decoupler mode, 168
Synthesizer label, 117, 470
synthesizer value, 470
sysel path global parameter, 72
sysel path global parameter, 72
sysepath global parameter, 218
sysmenul path global parameter, 72
system configuration parameters, 114
system console type, 119
system macros
  copy system macro to become user macro, 361
  display in text window, 360
  edit online description, 365
  list system macro names, 361
  online description, 365
  remove system macro, 361
system type configuration, 596
System Type label, 115, 117, 119, 504, 596
systemglobal directory, 127
systemglobal-type parameter tree, 119
systemglobal-type parameter tree, 127

T

T1 analysis, 221
  delete spectra, 151
  plot curves, 427
  set up parameters, 180
T2 analysis, 70
T1 analysis, 221
T2 analysis, 70
T2 analysis, 70
T2 analysis, 601
Index

t2 dimension in, 355
table conversion file, 605, 606
read, sort, store, 606
table conversion reformating, 605
table convert file
read, sort, and store, 606
tablib directory, 602
tangent value of angle, 603
tapes
device selection, 604
display contents, 603
rewind tape, 604
taps in digital filter, 580
target experiment
move parameters, 619
target scan planning, 439
target slice parameters, 581
target slices, 190
set slice parameters, 565
target voxels, 190
tau2 adjustment, 622
Tcl script, 606
tcrush parameter, 265
tdelta parameter, 266
tdiff parameter, 266
temperature calculation curve, 607
Temperature Control window, 607
temperature display, 43
temperature interlock, 613
temperature of sample, 607
temperature regulation, 644
terminating
abort function in macro, 39
calling macro, 39
testing signal-to-noise of spectrum, 608
text file
display for current experiment, 609
edit file, 610
edit with vi text editor, 634
editor, 212, 634
graphics window display, 208
plotting, 451
print text files, 469, 638
put into another file, 472
search for words and lines, 344
write file using a FID element, 660
text output sent to printer, 459
text window
display files, 107
display status, 610
display strings and parameter values, 211
display user macro, 358
list files, 163
tflow parameter, 274
third decoupler
adjust tip-angle resolution time, 171
decoupler mode, 168
decoupling sequence, 196
decoy power attenuator, 188
frequency, 156
frequency offset array, 529
frequency offset control, 177
homodecoupling control, 303
linear modulator power, 189
modulation frequency, 169
modulation mode, 174
nucleus lookup, 176
power level with linear amplifier, 187
set frequency to cursor position, 528
tip-angle resolution, 192
three-axis gradients, 128
threshold for integrating peaks in 2D spectra, 611
threshold for peak printout, 611
threshold for printout of peak frequencies, 610
tilted box, drawing a, 604
time constant for lock, 343
time constant for lock acquisition, 342
time counter, 666
time-domain cursor position, 128, 622
time-domain cursors, 152, 614
time-domain solvent subtraction, 413
time-shared decoupling, 302
tip-angle resolution for decoupler, 191, 192
tip-angle resolution time for decoupler, 170, 171
TMS reference, 614
TNCOSYPS sequence, 614
TNDOCOSY sequence, 614
TNMOCOSY sequence, 615
TNNOESY parameter set, 290
TNNOESY sequence, 615
TNROESY sequence, 615
TNTOSY sequence, 615
TOCSY experiment, changing parameters for, 615
TOCSY pulse sequence, 250, 254, 615, 656
TOCSY 1D experiment, changing parameters for,
616
tof parameter, 335
total correlation (TOCSY) experiment, 615
traces
find maximum intensity, 426
select trace without displaying, 529
TRANSFER.par file, 620
transform images into phasefiles, 364
transformed image array index, 213
transformed image echo index, 211
transients completed, 130
transients setpoint action, 657, 658
transients to be acquired, 391
transition amplitude, 111
transition calculation, 76
transition frequency, 113, 566
transition number calculation, 111
transitions frequency, 566
transmitter
fine power, 618
frequency of observe nucleus, 558
frequency offset for observe transmitter, 280, 616
linear modulator power, 619
local oscillator (L.O.) gate, 344
move transmitter offset, 376
nucleus of observe transmitter, 614
positioning, 616
power level with linear amps, 617
pulse sequence diagram, 185
transmitter frequency, 155, 156
transmitter frequency, 155
transverse magnetization generation, 676
tray size on sample changer, 620
trigger pulses, 612
trigger signals to wait before acquisition, 392
Index

trim gradient level, 290
triple-quantum filtered 2D MAS experiment, 140
TROESY pulse sequence, 622
truncating real numbers, 622
truncation limit, 131
TUNE INTERFACE unit, 623
tuning
  broadband channel on MERCURY, 101
  mode, 624
tuning the probe, 623
Type of Amplifier label, 68, 117, 118
type of parameter, 555
type, setting default, 537
U
U+ H1 Only decouplers, 499
Ultra•nmr shim system, 488, 561
unfixing/fixing slice gap, 540
unit conversion for parameters, 625
UNIT•INOVA console type, 119
UNIX shell startup, 559, 560
unload imaging data, 54
unlocked experiment, 627
unshifted cosine-squared window function, 577
unshifted Gaussian window function, 263
unshifted sinebell-squared window function, 578
update VnmrJ database, 141
updating
data entry, 267
gradient coil, 628
updating revision global file and parameters, 628
upper case format of string, 243
Upper Limit label, 117, 617
Use Console Data button, 115
user macros
copy file, 358
copy system macro to become user macro, 361
create without text editor, 126
delete, 150
display in text window, 358
display with vi text editor, 361
library, 127
list user macro file names, 359
path to user macro directory, 368
remove user macro from directory, 360
user-defined weighting, 662
users currently on the system, 648
user-selectable editor, 359
user-supplied modulation, 174
user-written weighting functions, 663
usrwt.o file, 663
V
value of parameter in a tree, 273
values, setting parameter, 557
variable temperature, see VT
Varian shim supply, 560
VAST accessory, 632
VAST data analysis, 113
VAST experiments, setting up initial parameters for, 631
VAST microtiter plate, 113
version of parameter set, 417
vertical offset, 639
vertical position
  FID, 640
    imaginary FID, 640
  spectrum, 640
vertical projection of trace, 144
vertical scale adjustment, 642, 643
vertical scale for 2D displays, 641
vertical scale for projections and traces, 643
vertical scale for simulated spectrum, 593
vertical scale for spectrum, 641
vertical scale of FID, 633
vi command (UNIX), 212, 634
vi text editor, 361, 369, 634, 636
viewing
drawings in 3D, 537
overlay as center lines/stripes, 163
overlay as stripes, 165
stripes/lines, 537
VNMFR
  background processing, 633
  exiting, 219
  exiting from system, 638
  start in windowing system, 636
  start VNMFR application directly, 636
style of stored data, 125
system directory, 597
updating parameters and global file after install, 628
user directory, 629
vnmr_textedit file, 412
vnmr_v1 file, 412
vnmraddr parameter, 246
vnmreditor variable, 212, 412
VnmrJ
  software preparation date, 498
  software revision level, 497
VnmrJ database, 141, 142
VnmrJ database, updating, 141
vnmrsystem variable, 597
vnmruser variable, 629
volume localized spectroscopy sequence, 584
voxel
  dimensions, 639
  parameters, 640
  planning menu, 439, 640
  selection, 291
Voxel button, 439
voxel orientation, 639
voxel selection gradient levels, 543
voxel selection gradients setup, 314
VT Controller label, 116, 117, 644
VT controller type, 644
VT cutoff point, 643
VT errors, 45
VT FAILURE message, 644
VT regulation light, 613
VT system in use, 367
VT wait time, 644
VXR-style directory
decompose to UNIX files, 149
VXR-style systems
  convert data to VNMFR, 120, 125
decompose files to UNIX files, 149
read tapes, 603
remote directory display, 213
VXR-style text files
conversion to UNIX format, 644

W
w command (UNIX), 648
walkup automation, 648
WALTZ decoupling present, 648
WALTZ decoupling sequence, 169
WALTZ-16 modulation, 174
warning error codes, 45
water suppression, 98, 323, 458, 614, 615
waveform generator, 505, 559
decoupling, 191
pulse interval time, 476
test, 654
waveform generator decoupling, 192
Waveform Generator label, 117, 505
weight and Fourier transform
1D data, 654
2D data, 655
along $f_2$ for 2D data, 654
phase-sensitive data, 655, 656
weight.h file, 663
weighting
constant, 91
interactive weighting, 663
interactive weighting for 2D absorptive data, 664
user defined, 662
weighting function compilation, 663
WET1D pulse sequence, 651
WETDQOSY pulse sequence, 651
WETGOSY pulse sequence, 651
WETGHMQPS pulse sequence, 651
WETGHSQC pulse sequence, 651
WETNOESY pulse sequence, 652
WETPWXCAL pulse sequence, 652
WETTOCSY pulse sequence, 652
WFG (waveform generator), 285
WFG + GCU gradients, 117
what you see is what you get, 664
whitespace in text file, 344
word lookup in text file, 344
workspace for VNMR experiment, 109
write out memory buffers, 241
wtlib directory, 663

X
X Axis, Y Axis, Z Axis label, 117, 285
X gradient strength, 291
X,H-correlation 2D spectrum, 446
X1 shim gradient, 665
X2Y2 shim gradient, 665
X3 shim gradient, 665
X4 shim gradient, 666
XL systems
convert data to VNMR, 120
deconvert files to UNIX format, 644
decompose files to UNIX files, 149
list contents of directory, 213
read tape, 603
XPOLAR1 pulse sequence, 667
XY shim gradient, 667
XY32 decoupling sequence, 169, 174
XZ shim gradient, 667
XZ2 shim gradient, 667
x-zero position of Hewlett-Packard plotter, 665

Y
Y gradient strength, 291
Y1 shim gradient, 669
Y3 shim gradient, 669
Y4 shim gradient, 669
YZ shim gradient, 669
YZ2 shim gradient, 667
y-zero position of Hewlett-Packard plotter, 669

Z
Z gradient strength, 291
z0 calibration, automatic, 81, 82
Z0 field position, 672
Z0, automatic adjustment, 82
Z1 shim gradient, 672
Z1C shim gradient, 672
Z2 shim gradient, 672
Z2C shim gradient, 673
Z2X2Y2 shim gradient, 673, 674
Z2X3 shim gradient, 673
Z2XY shim gradient, 673
Z3 shim gradient, 673
Z3C shim gradient, 673
Z3X shim gradient, 673
Z3X3 shim gradient, 674
Z3XY shim gradient, 674
Z3Y shim gradient, 674
Z3Y3 shim gradient, 674
Z4 shim gradient, 674
Z4C shim gradient, 674
Z4X shim gradient, 674
Z4X2Y2 shim gradient, 675
Z4XY shim gradient, 675
Z4Y shim gradient, 675
Z5 shim gradient, 675
Z5X shim gradient, 675
Z5Y shim gradient, 675
Z6 shim gradient, 675
Z7 shim gradient, 675
Index

Z8 shim gradient, 676
z-axis shims used for gradient shimming, 292
zero-filling, 241, 496
zeroing phase, 130
zero-order baseline correction, 356
zero-order phase, 509, 510
zero-order phasing constant, 433, 434, 510
zero-order term automatic phase, 72
zfs (zero-frequency suppression) option, 413
ZX2Y2 shim gradient, 676
ZX3 shim gradient, 676
ZXY shim gradient, 676
ZY3 shim gradient, 677