LIST OF CROWD COMMANDS

Version 1.0

LIST OF CROWD COMMANDS

This file is the alphabetic list of commands in CROWD.
It is a copy of the on-line help.

ADDCRD

addcrd atn no_res crdx crdy crdz
Add the atom whose the name is atn, the residu number is no_res, and the cartesian coordinates are crdx,crdy,crdz, to the atom chained list. As, for a CH2 or a CH3 group, the spin name is the name of the first atom in the group, where the first character is replaced by a 'Q', the atom names given in the command ADDCRD cannot begin by 'Q'. Also, in an atom group containing several equivalent atoms, the atoms should be add in the same order than in the command ADDTYP.
example: if a methyl group was defined in the following way:
ADDTYP 3 ala hb1 hb2 hb3
the methyl atom in a molecule should be defined in the same order:
ADDCRD HB1 2 -4.911 1.530 1.050
ADDCRD HB2 2 -5.800 0.001 0.967
ADDCRD HB3 2 -5.989 1.218 -0.325

see also : ADDTYP LSTCRD CRD2DST DELCRD

ADDINTE

addinte no_res1 atn1 no_res2 atn2 inte_value
Add the experimental intensity inte_value between the spin atn1 from the residu number no_res1, and the spin atn2 from the residu number no_res2, to the experimental intensity chained list.

see also : DELINTE LSTINTE MIX_INT RAT2INT

ADDLATT

addlatt latt_rate atn
Add to all the auto-relaxation rates (diagonal elements of the relaxation matrix), whose the spin name is atn, the spin-lattice relaxation rate latt_rate.

see also : CREL_NOESY DIAG RAT2INT

ADDSEQ

addseq no_res res_name
Define a residue by its residu number and its residu name. The residu number should be less than the number of residues defined by the command NB_RES

see also : INISEQ DELSEQ LSTSEQ ADDCRD

ADDTYP

addtyp id res_name atn1 (atn2 atn3)

Add the spin type whose the identificator is id (id = 1, for H, id = 2 for CH2 with equivalent atoms, id = 3 for CH3), the residu name is res_name, and the equivalent atom name(s) is(are) atn1 (atn2, atn3), to the spin type chained list. If the spin is a CH2 or a CH3 group, its name will be the name of the first atom in the group, where the first character will be replaced by a 'Q'. The atom names given in the command ADDCRD cannot begin by a 'Q'.

see also : DELTYP LSTTYP INITYP ADDCRD

ALERT

alert text
Creates a graphic alert box displaying the text. User must click in the alert box to continue. Works only if the graphic mode has been intialzed by isuing, at least once, a graphic command.

see also : PRINT ERROR

APPROX_INT

Calculates approximative theoretical intensities as proposed in the CROWD approximation
T.E. Malliavin, M.A. Delsuc, V.Y. Orekhov and A.S. Arseniev (1995) J.Bio.NMR 5, 193-201.
This command can be used only in the isotropic-rigid motion model.

see also : SUM_INT LSTSUM MIX_INT MOTION_MODEL

BUTTONBOX

BUTTONBOX menu_name [button_name crowd_command | separator] (n times) *
Permits to construct a new menu in the menu_bar (will create one if none is available). This menu will appear as menu_name in the menu_bar. When menu_name is clicked by the user, a button box will appear, with as many entry as item in the list of commands. Each entry in the list is defined as a couple of word, the first one being the name of the button in the box, the second one being the crowd command to be executed if the button is clicked. The command will be executed exactly as if the user had typed it as the prompt level, thus the command can be a simple command, several commands on a line, of a macro name. However the WHILE, FOR, IF .. THEN, GOTO commands are not available; the IF .. any_command syntaxe is available. If a parameter is missing for the command, the user will be prompted in the text window.
The special entry separator is not associated to a command, but permits to enter a separator in the menu box.
The list is terminated with a star.

see also : CLOSEBUTTON FORMBOX DIALOGBOX

CALC_R

CALC_R int_level switch
Calculate three different R-factors from the experimental and theoretical intensities.
the intensity R-factor: Ri = sum(i) |Itheo(i)-Iexp(i)| / sum(i) |Iexp(i)|
the distance R-factor: Rd = sum(i)| |Itheo(i)|**(1/6)-|Iexp(i)||**(1/6) / sum(i) |Iexp(i)|**(1/6)
the mean error: Dnoe = (1/n) sum(i) |Itheo(i)-Iexp(i) / Iexp(i)|
where: sum(i) means the sum over the index i, | a| means the absolute value of a.
int_level is the experimental intensity threshold.
switch = 1: experimental intensities from the autorelaxation peaks are taken into account in the calculation.
switch = 2: experimental intensities from the autorelaxation peaks are not taken into account in the calculation.

see also : RAT2INT MIX_INT INT2RAT CAL_INT

CAL_INT

Calibrate the experimental intensities to get the same values for the sum of experimental and theoretica intensities.

see also : MIX_INT CALC_R MIX_INT

CD

CD new_directory_path
Changes the current working directory.

see also : SH

CLOSE

CLOSE file_name
Closes the file that was previously OPENed.

see also : OPEN FPRINT CONNECT

CLOSEBUTTON

This command closes the current menu bar, and all the associated opened menu box. It is equivalent to closing it with the graphical close box.

see also : BUTTONBOX

CONFIG

Output to the terminal the actual configuration of the system dependent parameters :
-Licence being used
-Version number
-Sizes of the different static buffers
-Operating System (VMS or UNIX)
-Graphic manager (X_Windows or VWS)

-Plotter driver (HP-GL or Postscript)

see also: MEMORY

CONNECT

CONNECT file_name
The journaling of all the actions, performed by CROWD is directed to the file_name. Journaling is resumed to the standard crowd.log files with DISCONNECT.

see also : DISCONNECT LOG OPEN CLOSE REPORT

CRD2DST

Determine the spins of the molecule, and check that all the equivalent atoms of each spin were read. Then, calculate the distances between all the spins from the atom coordinates.

see also : ADDCRD CREL_NOESY DIAG RAT2INT

CREL_NOESY

Calculate the NOESY relaxation rates from the interatomic distances, the atom types and using the defined motion model. The HOHAHA and the off-resonance effects are not included in the calculation.

see also: CRD2DST MOTION_MODEL CREL_ROESY RAT2INT

CREL_ROESY

Calculate the ROESY relaxation rates from the interatomic distances, the atom types and using the defined motion model.

see also: CRD2DST MOTION_MODEL CREL_NOESY RAT2INT

CREL_ROEOFF

CREL_ROEOFF theta
Calculate the off-resonance ROESY relaxation rates from the interatomic distances, the atom types and using the defined motion model. theta is the spin-lock nutation angle.
H. Desvaux, P. Berthault, N. Birlirakis and M. Goldman (1994) J. Magn. Res. 108A, 219-229.

see also: CRD2DST MOTION_MODEL CREL_ROESY RAT2INT

DBCLOSE

DBCLOSE array_name
Closes the db file previously associated to the array array_name by a previous DBOPEN command.

see also : DBOPEN

DBOPEN

DBOPEN file_name array_name

associates the array array_name with the db file file_name. Each operation on the array will be propagated to the file : access, change, delete, exist(), nextlm() etc... Any db file can be accessed.

see also : DBCLOSE VARIABLES FUNCTIONS

DEBUG

When this context is set to 1 some command generate some debuging information. Used mainly by developers.

see also : VERBOSE REFMACRO TIMER

DELCRD

delcrd atn no_res
Remove the atom whose the name is atn and the residu number is no_res, from the atom chained list.

see also : ADDTYP LSTCRD CRD2DST ADDCRD

DELINTE

delinte no_res1 atn1 no_res2 atn2
Remove the experimental intensity value between the atom whose the name is atn1 and the residu number is no_res1, and the atom whose the name is atn2 and the residu number is no_res2.

see also : ADDINTE LSTINTE INI_INTE MIX_INT

DELSEQ

delseq no_res
Undefine the residu whose the number is no_res in the residu array.

see also : ADDSEQ LSTSEQ INISEQ ADDCRD

DELTYP

deltyp id res_name atn1 (atn2 atn3)
Remove the spin type whose the identificator is id (id = 1, for H, id = 2 for CH2, id = 3 for CH3), the residu name is res_name, and the atom name(s) is(are) atn1 (atn2, atn3).

see also : ADDTYP LSTTYP INITYP ADDCRD

DIAG

diag switch
Diagonalize the relaxation rate matrix (switch = 'exp') or the intensity matrix (switch = 'log').

see also : CRD2RAT RAT2INT INT2RAT

DIALOGBOX

DIALOGBOX dialog_name ...
... [field_name field_type [field_list [enum_list] field_var default_value] | separator] (n times) *

Permits to construct a dialog box with editable fields. The box will appear in a window called dialog_name . The Crowd program will be blocked as long as the user has not closed the window. It will have two buttons at the button :
Ok closes the form and resume the execution of Crowd
Cancel closes the form, abort the command and generate an error

The dialog is also composed en fields which appear one on each line in the window. A dialog can hold as many field as needed. Each field has :
* a name field_name that will show on the left side of the field
* a type field_type which is chosen in the following list : (message, string, int, real, file, enum)
message type of field have no other field, this kind of entries serves to put plain text in the dialog
* All other type have a field_var entry which determines the name of the Crowd vairable in which the user input will be stored, and the default_value entry which determines the value of that variable when creating the form.
string int real and **file** types of field have no other entry. They appear in the window as editable string. When the user hits return there, the content of the string is copied to the Crowd variable field_var.
file entries will have a small arrow which permits to open a standard file dialog window.
enum have an additional entry : enum_list.. This kind of field will appear as a pop up menu permitting to choose between the entries defined in the list : enum_list. This list is a single blank separated word, each item being separated with commas.

The special entry separator is not associated to a variable, but permits to enter a separator in the dialog.

The list is terminated with a star.

see also : FORMBOX BUTTONBOX CLOSEBUTTON

DISCONNECT

Journaling is resumed to the standard crowd.log file when previously redirected with the CONNECT command.

see also : CONNECT LOG OPEN CLOSE REPORT

DUMP

Lists the values of all the variables presently defined within CROWD.
Each value is shown with its associated context : context 20 is the global context, higher context are associated to higher call levels.

see also : SET

ELSE

part of the IF command

see also : IF

ELSIF

part of the IF command
see also : IF

ENDFOR

part of the FOR command
see also : FOR

ENDIF

part of the IF command
see also : IF

ENDWHILE

part of the while command
see also : WHILE

ERROR

ERROR string
Generates an error, and issue the string to the user.
Permits to handle errors in command files.

see also ALERT PRINT

EXIT

EXIT yes/no
Exit this program. You can choose to keep or to remove the log file
QUIT and BYE are also equivalent.

FOR

FOR var = n TO p { STEP k }
... CROWD commands
ENDFOR

Control structure permitting to loop over a set of commands. The variable var will be created if does not yet exist. By default of STEP is 1, but any value can be used (including negative values in which case n should be larger than p).

FOR and ENDFOR should appear alone on one line, eventually followed by a comment.

see also : IF WHILE GOTO

FORMBOX

FORMBOX form_name crowd_command ...
... [field_name field_type [field_list [enum_list] field_var default_value] | separator] (n times) *
Permits to construct a form (dialog box with editable fields). The box will appear in a window called form_name . The program continues after having created the form, whitch will stay on screen as long as the user

dos not close it exolicitely. It will have three buttons at the button :
Apply executes the command crowd_command and the form will remain on screen
Ok executes the command crowd_command and closes the form
Cancel does not execute any command and closes the form

The command will be executed exactly as if the user had typed it as the prompt level, thus the command can be a simple command, several commands on a line, of a macro name. However the WHILE, FOR, IF .. THEN, GOTO commands are not available; the IF .. any_command syntaxe is available. If a parameter is missing for the command, the user will be prompted in the text window.

The form is also composed en fields which appear one on each line in the window. A form can hold as many field as needed. Each field has :
* a name field_name that will show on the left side of the field
* a type field_type which is chosen in the following list : (message, string, int, real, file, enum)
message type of field have no other field, this kind of entries serves to put plain text in the form
* All other type have a field_var entry which determines the name of the Crowd variable in which the user input will be stored, and the default_value entry which determines the value of that variable when creating the form.
string int real and **file** types of field have no other entry. They appear in the window as editable string. When the user hits return there, the content of the string is copied to the Crowd variable field_var.
file entries will have a small arrow which permits to open a standard file dialog window.
enum have an additional entry : enum_list.. This kind of field will appear as a pop up menu permitting to choose between the entries defined in the list : enum_list. This list is a single blank separated word, each item being separated with commas.

The special entry separator is not associated to a variable, but permits to enter a separator in the form.

The variables used by the editable entries can be global variables, in which case they should have been created as global before calling the FORMBOX command. If this is not the case, they are created in a special context, available only to the command crowd_command when executed with the **Apply** or **Ok** buttons. This special context is completely local to the form. This permits to have several forms, using the same variables syntax, without interaction.

The list is terminated with a star.

see also : DIALOGBOX BUTTONBOX CLOSEBUTTON

FPRINT

FPRINT file_name string
Outputs the string to the file. file must have previously be OPENed (see OPEN).

see also : OPEN CLOSE PRINT

FUNCTIONS

the following functions and operators are available within evaluated expression :

Operators :
o the regular 4 operations, modulo and power
    + - / * % ^
o arithmetic comparison
    == != < > <= >=
o string comparison
    s= s!
o logical operators
    | & !
o string concatenation
    // ;

functions :
o the mathematical functions
    sqrt(x) cos(x) sin(x) atan(x) log(x)
    exp(x) abs(x) int(x) power2(n) max(x,y)
    min(x,y)
o the alphanumeric functions :
    toupper(st) tolower(st) sp(i)
    len(st) index(st1,st2)
    subst(st,i,j) head(st) tail(st)
    headx(st,char) tailx(st,char)
o file access
    eof()

see also : VARIABLES

GOTO

GOTO label
Will branch the flow a macro to the label. Labels are alphanumeric strings, and appear preceded by an = sign in macro text. Labels should be alone on the line. GOTO cannot be used in interactive mode, but only in macros.

see also : IF

This is the CROWD Version 1.0 program.

Type help Name_of_item to get info about a specific command. H can be used instead of HELP.
Use apropos topic, to find commands name related to a specific topic.
Use help primer for a first introduction.

IF

Used for tests in command.
2 syntaxes are available :

IF test THEN

```
..commands on several lines
{ ELSIF test2 THEN
..commands }   (eventually many exclusive tests )
{ ELSE      (default case)
..commands }
ENDIF
```
The different commands will executed conditionnally on the value of the tests. A non-zero value is considered as true.
Permits to construct complex tests in command files.

IF test remaining_of_the_line
execute the remaining_of_the_line conditionally on the value of test. It will be executed only if test is true (non zero). The remaining of the line can span several lines by using the line continuation sign : \
This form can be used at the prompt level, as well as in call-backs, for instance in graphic buttons. It is called the one-line IF.

Tests available are :
for numeral : == != < > <= >=
for strings : s= s!
for combinations: & | !
as well as several tests functions and variables :
exist() eof() $arg $c_joined, etc...

IF(in the first syntaxe) ELSIF ELSE and ENDIF should appear alone on one line, eventually followed by a comment. There is no limitation for the one-line IF.

see also FUNCTIONS GOTO FOR WHILE

INICRD

Initialize the atom chained list.

see also : LSTCRD CRD2DST ADDCRD DELCRD

INITINPROGRESS

INITINPROGRESS n
Presets for n iterations, the progress bar, of the form :
In Progress : 0%....25%....50%....75%....100%
The progress bar is then updated with the INPROGRESS command.

see also : INPROGRESS PRINT

INIT_INT

Initialize the intensity matrix by the experimental intensities. The intensity matrix elements which are not experimentally determined are set to zero.

see also : MIX_INT ADDINTE INT2RAT

INIT_INTE

Initialize the expertimental intensity chained list.

see also : MIX_INT ADDINTE DELINTE LSTINTE

INISEQ

Undefine all the residus whose the residu number is smaller than the number of residus defined by the NB_RES command. The residu name of undefined residus are set to 'UNK'.

see also : ADDSEQ DELSEQ LSTSEQ ADDCRD

INIS2

Initialize the generalized order parameter array.

see also : MOTION_MODEL S2 LSTS2 TAU DST2RAT

INITAU

Initialize the local correlation time array.

see also : MOTION_MODEL TAU LSTTAU S2 DST2RAT

INITYP

Initialize the atom type chained list.

see also : ADDTYP LSTTYP DELTYP ADDCRD

INPROGRESS

INPROGRESS i
Displays the progress of the operation in the progress bar, inited with the INITINPROGRESS command.

see also : INITINPROGRESS PRINT

INT2DST

INT2DST diam seed
Generate interatomic diatnces from the experimental intensity values. The distances are randomly chosen by using the random generator seed, into the following intervals:
if intensity >= 0.2, 1.8 <= distance < 3.5
if 0.05 <= intensity < 0.2, 1.8 <= distance < 4.5
if intensity <= 0.05, 1.8 <= distance < 5.0
if there is no intensity, 4.5 <= distance <= diam
M. Ikura, G.M. Clore, A.M. Gronenborn, G. Zhu, C.B. Klee and A. Bax (1992) Science 256, 632-638.

see also : INTE RAT2INT CRD2DST

INT2RAT

int2rat ratmax
Cakulate the relaxation rate matrix from the eigenvalues and eigenvectors of the intensity matrix, at a mixing time defiend by the command TMIX. All the non diagonal relaxation matrix elements lerger to ratmax are put

equal to ratmax, in order to correct the relaxation matrix distorsions.

see also : RAT2INT DIAG MIX_INT TMIX

LOG

The LOG file is a text file journaling all the actions performed during the session in the CROWD program. This file is called crowd.log. When exiting the program, you are asked whether you want to keep this file for latter use.

see also : CONNECT DISCONNECT REPORT

LOOK

LOOK atn1 no_res1 atn2 no_res2

Shows the distance, the relaxation rate and the intensity between the spins, whose names are atn1 and atn2, and whose residu number are no_res1 and no_res2.

see also : CREL_NOESY DIAG RAT2INT

LSTCRD

lstcrd no_res

List the atom names and coordinates of the residu number no_res.

see also : ADDTYP ADDCRD CRD2DST DELCRD

LSTINTE

List the experimental intensity values

see also : ADDINTE DELINTE INI_INTE MIX_INT

LSTMIX

List the mixing time values.

see also : NB_TMIX TMIX WRBLDP

LSTSEQ

LSTSEQ idx_beg idx_end

List the sequence residu names from the residue number idx_beg to the residu number idx_end.

see also : ADDTYP DELTYP INITYP ADDCRD

LSTSUM

List the sum of intensities (used in the CROWD approximation
T.E. Malliavin, M.A. Delsuc, V.Y. Orekhov and A.S. Arseniev (1995) J.Bio.NMR 5, 193-201.)

see also : SUM APPROX_INT MIX_INT

LSTS2

List the generalized order parameter values.

see also : S2 INIS2 MOTION_MODEL CREL_NOESY

LSTTAU

List the local correlation time values.

see also : TAU INITAU MOTION_MODEL CREL_NOESY

LSTTYP

LSTTYP res_name
List the atom types of the residue name res_name

see also : ADDTYP DELTYP INITYP ADDCRD

MACRO

Macros are text files holding regular CROWD commands. Loops, tests, controls, input/output and variables are available within macro.
Type more /usr/local/crowd/macro/list for a short description of all the available macros

see also : GOTO IF FOR WHILE SET VARIABLES FUNCTIONS

MEMORY

The Crowd program holds all the data in memory. It presents several buffers to store the differents objects it handles:
molecule residus (RES): res(szres)
atom coordinates (CRD) : crd(szat)
inter-atomic distances (DST): dst(szmat)
relaxation rates (RATE): rateh(szmat)
theoretical intensities (INT): inttheo(szmat)
mixing times (TMIX): tm(sztm)
experimental intensities (INTE): intexp(szmat)
internal correlation times (TAU): tau(szmat)
order parameters (S2): s2(szmat)
where szres is the maximum number of molecule residus, szat is the maximum number of atoms and szmat = szat*(szat+1)/2. The current szat, szres and sztm values are given by the command CONFIG.
The other arrays are used for handling chained list, and as working arrays for matrix diagonalisation.

see also : CONFIG

MESSAGE

message text

This command permits to build interactive macros. If no argument is present on the calling line when MESSAGE is executed, the text will be issued to the user when the next data entry will be needed; if an argument is present, nothing will happend.

If the macro is called from a menu button, then the text will appear in the data entry dialog box.

see also : PRINT VARIABLES ALERT

MIX_INT

Produces a "mixed" intensity matrix, from the theoretical intensity matrix and the experimental intensities. The intensity matrix element, for which an experimental intensity value was read, is replaced by this value.

see also : ADDINTE RAT2INT INT2RAT CALC_R

MOTION_MODEL

motion_model id
This command determine the kind of motion model used:
id = 1: rigid-isotropic model. The global correlation time is defined by the command TAUC.
id = 2: local correlation time model. A correlation time is defined for each atom pair by the command TAU.
id = 3: Lipari-Szabo "model-free" approach. The order parameter and internal motion correlation time are defined by the commands S2 and TAU.

see also : TAUC S2 TAU

MUNSET

munset list_of_variable_name *
Removes all the variables which names are found in list_of_variable_name from the variable table. The list is terminated by a * sign. Note that no $ sign are needed.

see also : SET UNSET VARIABLES

NB_RES

Define the number of residues in the sequence.

see also : ADDSEQ INISEQ LSTSEQ ADDCRD

NB_TMIX

Define the number of mixing times.

see also : TMIX LSTMIX RAT2INT INT2RAT

OMEGA

Define the hydrogen resonance frequency in MHz.

see also : TAUC TMIX

OPEN

OPEN file_name
Opens the file called file_name for reading or writing. Writing is performed with the FPRINT command. Reading is done through the <file_name syntax.
e.g. read <file_name
read the file which name is found as the next line of the file file_name
set line = <file_name

set the variable line to the contents of the next line of the file file_name
OPENing an already OPENed file will rewind the file to the beginning.

see also : CLOSE FPRINT

PENTE

PENTE calib_int calib_dst
Calculate the interatomic distances from the experimental intensities, by using the initial slope approximation: intensity is proportional to the inverse of the distance sixth power. calib_int and calib_dst are respectively calibration intensity and distance. Write the interatomic distances to the journal unit.

see also : INT2DST CREL_NOESY CONNECT

PRINT

PRINT string
This command permits to issue a string to the user. Permits interactivity in command file.

see also : FPRINT ERROR ALERT SHOWTEXT PLOTTEXT

QUIT

QUIT yes/no
Equivalent to QUIT

see also : LOG

RAT2INT

Calculate the intensity matrix from the eigenvalues and eigenvectors of the relaxation matrix, at the first mixing time array element, defined by the command TMIX.

see also : INT2RAT DIAG MIX_INT TMIX

REPORT

Write a report of the current iteration, giving a list of all atom pairs, for which an experimental intensity has been read. This output line format is: res_name1 res_no1,atn1, res_name2, res_no2,atn2,disti, expint, rate,theoint,dstc, where res_name are residu names, res_no are residu numbers, atn are atom names, disti is the initial inter-atomic distance, expint is the experimental intensity, rate, theoint and dstc are the current relaxation rate, theoretical intensity and interatomic distance.

see also : WR_INT WRIBLDP CONNECT DISCONNECT LOG

SET

SET variable_name = value
SET static_variable := value

Assigns the value of the variable variable_name to value. Creates the variable if it does not already exist. Note that you should not use a $ sign before variable_name. value can be any kind of value : number, string, read from a file, another variable, a calling parameter ($_) or an evaluated expression.
If value is missing, or if $_ is used and no parameter are available, the user is prompted for input.
When the variable is created during the execution of a macro, the first syntax creates a local/volatile variable which value cannot be accessed by subsequent called macros, and that will be removed at the end of the macro. The second syntax creates a global/static variable that can be accessed by any macros, and that will remain until explicitly removed with the UNSET command.

see also : UNSET MUNSET DUMP VARIABLES FUNCTIONS

SH

SH string
Will pass the string to the operating system. For instance, used by macros such as pwd, ls, more, etc...
SH alone will enter the Operating system

see also : CD

SMOOTH_INT

Modify the theoretical intensity to obtain consistent data: the sum of each row should be smaller than the number of spins, and each non-diagonal intensity should be smaller than the diagonal intensity located in the same row.
This command should be applied after each MIX_INT command

see also : MIX_INT CAL_INT CALC_R INT2RAT

SUM

sum atn no_res t1_value
Read the sum of intensitie of the spin, whose the name is atn and the residu number is no_res (used in the CROWD approximation
T.E. Malliavin, M.A. Delsuc, V.Y. Orekhov and A.S. Arseniev (1995) J.Bio.NMR 5, 193-201.)

see also : LSTSUM APPROX_INT MIX_INT

S2

S2 atn1 no_res1 atn2 no_res2 s2_val
Read the order parameter value s2_val between the spin atn1 from the residu number no_res1, and the spin atn2 from the residu number no_res2.

see also : TAU INIS2 LSTS2 MOTION_MODEL

TAU

```
TAU atn1 no_res1 atn2 no_res2 tau_val
Read the local correlation time tau_val (ns) between the
spin atn1 from the residu number no_res1, and the spin
atn2 from the residu number no_res2.

see also : S2 INITAU LSTTAU MOTION_MODEL
```

TAUC

```
Define the global correlation time, in nanosec.

see also : S2 TAU MOTION_MODEL CREL_NOESY
```

TIMER

```
When this context is set to 1, then the time needed to
execute each command is given. If the command TIMER 1 is
given while the value of timer is already 1, the timing
is issued, and the clock reseted to 0. Within macros,
only the overall execution time of the macro is given,
unless an explicit TIMER 1 or TIMER 0 is given.

see also : VERBOSE DEBUG REFMACRO SH
```

TMIX

```
TMIX tm_indx tm_val
Put the value tm_val at the tm_indx element number in
the mixing time array.

see also : NB_TMIX LSTMIX RAT2INT INT2RAT
```

UNSET

```
unset variable_name
Remove the variable variable_name  from the variable
table. Note that no $ sign are needed.

see also : MUNSET SET
```

VARIABLES

```
List of the internal CROWD variables :
$_   value of the next parameter present on the calling
line. CANNOT BE USED WITHIN EVALUATED EXPRESSIONS.
```
if the following command is used :
```
@test 3 test.001
```
within the file test, $_ will be 3 the first time and test.001 the second time.
If no value is present on the calling line, the user will be prompted for the value
```
$ARG True (1) if arguments are present on the calling
line (within macro only)
$CONFIG_OS     The system type, as returned by the
CONFIG command
$CONFIG_WIN    The window manager, as returned by the
CONFIG command
```
$DNOE the mean error between theoretical and experimental intensities.
$LEVEL the intensity threshold.
```
$LICENCE  The licence as returned by the CONFIG command
```
$MOTION_MODEL the value of the motion model identificator.
$NB_RES the number of residus
$NB_SPIN the number of spins
$NB_TMIX the number of mixing times

$NBTYP_RES the number of residu types.
$OMEGA the hydrogen resonance frequency.
$RANDOM a random variable in the range 0..1 with equiprobable distribution
$RANDOMG a random variable with normal law, unit variance and zero mean
$RANDOMZ same as $RANDOM but resets the random series
$RFACTD the distance R-factor value between theoretical and experimental intensities.
$RFACTI the intensity R-factor value between theoretical and experimental intensities.
$SEQ[$i] the ith residu name in the molecule sequence.
$SZAT maximum number of places in the atom chained list.
$SZRES maximum number of residus in molecular sequence.
$SZTM maximum number of mixing times.
$SZTYP maximum number of places in the atom type chained list.
$TAUC the global correlation time.
$THETA the spin-lock nutation angle.
$TMIX[$i] the ith mixing time value.
$VAR_MAX The total number of user variable available, as returned by the CONFIG command
$VERSION The current version, as returned by the CONFIG command

see also : FUNCTIONS SET UNSET MUNSET

VERBOSE

When this context is set to 1 the program switches to a more verbose mode. Specially, text output will be generated during macro execution. When verbose equals to 0, no text is generated.

see also : DEBUG

WHILE

```
WHILE test
..commands
ENDWHILE
```

Commands in the WHILE .. ENDWHILE structure will be executed as long as the test holds true. Can be used only in Macros.

WHILE and ENDWHILE should appear alone on one line, eventually followed by a comment.

see also : IF FOR GOTO FUNCTIONS

WRIBLDP

WRIBLDP filename
Calculate at the mixing times defined in the mixing time array, and write to filename the intensities corresponding to the atome pairs, for which an experimental intensity was read. The intensities are calculated from the relaxation matrix diagonalisation decomposition. The build-up file looks like a 2D NMR data matrix, where each column is a build-up curve

between a given atom pair. The file format is binary and can be read by the command READH in Crowd. The order of the atom pairs is given by the command LSTINTE.

see also : WR_INT TMIX RAT2INT LSTINTE

WR_INT

WR_INT filename int_thres header
Write to filename the intensities, whose absoluate values are larger than int_thres. header is the file header.

see also : WRIBLDP TMIX RAT2INT