TopSpin

● Acquisition

Commands and Parameters

Version 003

Innovation with Integrity

Document Number: Unknown

P/N: H9775SA3

Contents

Contents

Contents

1 Introduction

1.1 About This Manual

This manual is a reference to TopSpin acquisition or acquisition related commands and parameters. Every command is described on a separate page with its syntax and function as well and its main input/output files and parameters. Although file handling in TopSpin is completely transparent to the user, it is sometimes useful to know which files are involved and where they reside. For example, if you have permission problems or if you want to process or interpret your data with third party software.

Some of the commands referred to in this manual are processing commands. They are all described in the Processing reference manual.

1.2 Conventions

Type of Information	Font	Examples
Shell Command, Commands, "All what you can enter"	Arial bold	Type or enter **fromjdx zg**
Button, Tab, Pane and Menu Names "All what you can click"	Arial bold, initial letters capitalized	Use the **Export To File** button. Click **OK**. Click **Processing**…
Windows, Dialog Windows, Pop-up Windows Names	Arial, initial letters capitalized	The Stacked Plot Edit dialog will be displayed.
Path, File, Dataset and Experiment Names Data Path Variables Table Column Names Field Names (within Dialog Windows)	Arial Italics	*$tshome/exp/stan/nmr/lists* *expno, procno,*
Parameters	Arial in Capital Letters	VCLIST
Program Code Pulse and AU Program Names Macros Functions Arguments Variables	Courier	`go=2` `au_zgte` `edmac` `CalcExpTime()` `XAU(prog, arg)` `disk2, user2`
AU Macro	Courier in Capital Letters	`REX` `PNO`

Table 1.1: Font and Format Conventions

File/Directory Conventions

<tshome> - the TopSpin home directory (default C:\Bruker\Topspin under Windows or /opt/Topspin under Linux)

Header Conventions

>SYNTAX - only included if the command described requires arguments

>USED IN AU PROGRAMS - only included if an AU macro exists for command described

1.3 About Dimensions

TopSpin can acquire up to 8 D dimensional data. The directions of a dataset are indicated with the terms F1, F2, F3 etc. which are used as follows:

1D Data:

>F1 - acquisition direction

2D Data:

>F2 - acquisition or direct direction

>F1 - indirect direction

3D Data:

>F3 - acquisition or direct direction

>F2 - indirect direction

>F1 - indirect direction

In 3D processed data, F2 is always the second and F1 the third direction. In 3D raw data, this order can be the same or reversed, depending on the value of AQSEQ (see the description of this acquisition parameter).

1.4 About Digitally Filtered Avance Data

The first points of the raw data measured on an Avance spectrometer are called group delay. These points represent the delay caused by the digital filter and do not contain spectral information. However, they contain information about the digital filtering and are required for processing. The first couple of points of the group delay are always zero. The group delay only exists if digital filtering is actually used, i.e. if the acquisition parameter DIGMOD is set to digital.

1.5 Usage of Acquisition Commands in AU Programs

Many acquisition commands described in this manual can also be used in AU programs. The description of these commands contains an entry USAGE IN AU PROGRAMS. This means an AU macro is available which is usually the name of the command in capitalized letters.

> **i** Note that IconNMR automation automatically calls acquisition AU programs. If, in this manual, the entry USAGE IN AU PROGRAMS is missing, no AU macro is available.

Usually, such a command requires user interaction and it would not make sense to put it in an AU program. However, if you still want to use such a command in AU, you can do that with the XCMD macro which takes a TopSpin command as argument. Examples are:

>XCMD("eda")

>XCMD("setdef ackn no")

AU programs can be set up with the command **edau**.

Acquisition commands can also be used in an TopSpin macro. These are scripts created with **edmac** containing a sequence of TopSpin commands or Python commands.

1.6   Starting Commands from the TopSpin Menu

This manual describes all acquisition commands as they can be entered on the command line. However, they can also be clicked from the upper toolbar or from the TopSpin popup menus. Most acquisition commands can be found under the **Spectrometer** menu.

> **i** Note that this menu is only available after TopSpin has been configured as a spectrometer (command **cf**). There, the command line commands which correspond to the menu entries are specified in square brackets.

1.7   Command Queuing

In TopSpin 2.1 and newer, command spooling has been implemented. Acquisition commands like **zg**, **rga**, **atma** and **go** are automatically queued, if this feature is **on** (default **off**, can be set with the command **set**). All other commands can be queued with the command **qu**, e.g. **qu xfb**.

Queued commands can be viewed in the Spooling field of the acquisition status bar.

> **i** Note that the spooling field must be activated in the User Preferences window (command **set**).

2   TopSpin Parameters

2.1   About TopSpin Parameters

TopSpin parameters are divided in acquisition and processing parameters. In this manual, we will mainly concern ourselves with acquisition parameters. Furthermore, we will discuss the acquisition related lock and prosol parameters.

The following terms will be used:

Acquisition Parameters

Parameters that must be set by the user, for example with **eda**, and that are interpreted by acquisition commands, for example **zg**.

Acquisition Status Parameters

Parameters that are set by acquisition commands like **zg**. They represent the status of the raw data and can be viewed, for example with **dpa**. Some acquisition status parameters are used as input by processing commands.

Lock Parameters

Parameters that are used for locking the magnetic field. They can be set up with the **edlock** command and are interpreted when you lock in, either with the **lock** command or from the BSMS keyboard/BSMS display.

Prosol Parameters

Probe and solvent dependent parameters, mainly pulse lengths and power levels. They can be set up with the **edprosol** command. The **getprosol** command reads the prosol parameters and copies them to the corresponding acquisition parameters.

> **i** Note that entering **getprosol** is equivalent to clicking the **AcquPars** tab and then clicking the ⬛ button.

Input Parameters

Parameters that are interpreted by the commands described in this manual. They can be:
- Acquisition parameters (input of, for example, **zg**)
- Lock parameters (input of **edlock**, **lock** and **lopo**)
- Prosol parameters (input of **edprosol** and **getprosol**)

Output Parameters

Parameters that are set or modified by commands described in this manual. They can be:
- Acquisition status parameters (output of, for example, **zg**)
- Lock parameters (output of **edlock**)
- Prosol parameters (output of **edprosol**)

Temporary Parameters

Parameters that are not stored in parameters files and not interpreted directly by acquisition commands. They are related to other parameter that are directly interpreted by acquisition commands. If you change a temporary parameter, for example in **eda**, the related parameters will be automatically adjusted. An example of a temporary parameter is AQ that is determined by the equation:

$$AQ = TD/(2*SW*SFO1)$$

Acquisition parameters can be set with the parameter editor **eda** and acquisition status parameters can be viewed with **dpa**. Alternatively, each parameter can be set or viewed by entering its name in lowercase letters on the command line. For example, for the parameter TD:

- **td** - set the parameter TD
- **s td** - view the status parameter TD

The dimensionality of the dataset is automatically recognized. For example, for a 2D dataset the following dialog box is offered:
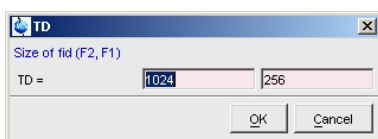
Although status parameters are normally not changed by the user, a command like **std** allows you to do that. This, however, would make the dataset inconsistent.

Before an acquisition has been performed, the acquisition status parameters of a dataset do not contain significant values. After the acquisition, they represent the status of the raw data.

Most acquisition status parameters are set to the same values as the corresponding acquisition parameters. In other words, the acquisition command has done what you told it to do. There are, however, some exceptions:

- When an acquisition was interrupted, the acquisition status parameters might not have been updated yet.
- Some acquisition parameters are automatically adjusted by the acquisition command, e.g. RG and FW.
- The values of some parameters are a result of the acquisition. They cannot be set by the user (they do not appear as acquisition parameters) but they are stored as acquisition status parameters. Examples are AQSEQ, YMAX_a and NC.

2.2　Parameter Value Types

With respect to the type of values they take, acquisition parameters can be divided into three groups:

- Parameters taking integer values, e.g. NS, TD, DR
- Parameters taking real (float or double) values, e.g. SW, O1, DE
- Parameters using a predefined list of values, e.g. AQ_mod, DIGTYP

You can easily see to which group a parameter belongs from the parameter editor opened with the command **eda**.

> Note that the values of parameters which use a predefined list are stored in the parameter file as integers.

The first value of the list is always stored as 0, the second value as 1 etc. The next table shows the values of the parameter AQ_mod as an example:

Parameter value	Integer stored in the proc(s) file
qf	0
qsim	1
qseq	2
DQD	3

Table 2.1: Parameter value types

2.3   Parameter Files

TopSpin parameters are stored in various files in the dataset directory tree.

In a 1D Dataset:

- \<dir>/data/\<user>/nmr/\<name>/\<expno>/
    - *acqu* - acquisition parameters
    - *acqus* - acquisition status parameters

In a 2D Dataset:

- \<dir>/data/\<user>/nmr/\<name>/\<expno>/
    - *acqu* - F2 acquisition parameters
    - *acqu2* - F1 acquisition parameters
    - *acqus* - F2 acquisition status parameters
    - *acqu2s* - F1 acquisition status parameters

In a 3D Dataset:

- \<dir>/data/\<user>/nmr/\<name>/\<expno>/
    - *acqu* - F3 acquisition parameters
    - *acqu2* - F2 acquisition parameters
    - *acqu3* - F1 acquisition parameters
    - *acqus* - F3 acquisition status parameters
    - *acqu2s* - F2 acquisition status parameters
    - *acqu3s* - F1 acquisition status parameters

2.4   Acquisition (eda) Parameters

This paragraph contains a list of all acquisition parameters with a description of their function. Most of them are interpreted by various acquisition commands like **zg**, **go**, **ii**, **resume**, **gs** and **rga**. Some, however, are only interpreted by specific commands which are then specified in the list below. Acquisition parameters can be set by entering **eda** on the command line, clicking the AcquPars tab of a data window or by typing the parameter names in lowercase letters on the command line.

AQ - Acquisition Time In Seconds

- Takes a float value
- Temporary parameter calculated from the equation:

    $AQ = TD/(2*SW*SFO1)$

- AQ represents the time to acquire one scan. If you change AQ, TD is changed accordingly. The above equation holds for DIGTYP = SADC. Other digitizers require a 1-4 extra dwell times. This number is automatically detected from your digitizer by the acquisition software.

AMP[0-31] - Amplitude Of Pulses

- Takes float values in percent
- Can be set from **eda** by clicking **AMP ** Array ****
- Can also be set by entering **amp0**, **amp1** etc. on the command line
- Can also be set from the **gs** window
- Interpreted by the pulse program statements `amp0, amp1, amp2` etc.
- The parameter AMP defines the percentage of the maximum pulse power. It can be used instead of or in addition to the parameter PL. The advantage of AMP is that the relation between AMP values and pulse power is more linear than in the case of PL. Furthermore the pulse phases is more stable.

AQ_mod - Acquisition Mode

- Takes one of the values **qf**, **qsim**, **qseq**, **DQD**
- Can be set from **eda** or by entering **aq_mod** on the command line
- The values of AQ_mod have the following meaning:

  qf = single channel detection.

  qseq = quadrature detection in sequential mode. Two channels are used, whose reference phase differs by 90°. In the resulting fid, two successive data points have been acquired by different detectors with a time difference of DW.

  qsim = quadrature detection in simultaneous mode. Two channels are used, whose reference phase differs by 90°. In the resulting fid, two successive data points have been acquired simultaneously by the two detectors. The time difference between these points is 2* DW.

  DQD = digital quadrature detection. Simultaneous mode that eliminates quad images and O1 spikes. AQ_mod can only be set to DQD when the parameter DIGMOD is set to *digital* or *homodecoupling digital*. When you set DIGMOD to analog, AQ_mod automatically changes to **qsim**. Furthermore, DQD can only be used up to a certain spectral width as is shown in table *Table 2.6 [▶ 34]*. Above this value, acquisition commands automatically switch the acquisition mode to qsim. In that case, the acquisition parameter AQ_mod = DQD but the acquisition status parameter AQ_mod = qsim.

AUNM - Name Of An Acquisition Au Program

- Takes a character array value
- Can be set from **eda** or by entering **aunm** on the command line
- Interpreted by **xaua**
- The command **xaua** executes the AU program specified by AUNM. Although this can be any AU program, AUNM is normally used to specify an AU program that performs an acquisition. For example, in several standard parameter sets, AUNM is set to **au_zg**. The command **xaua** can be entered on the command line or called from AU program with its macro XAUA.

BF1 - BF8 - Basic Frequency For Frequency Channel f1 to f8

- Take a double value (MHz).

- Are automatically set when NUC1, NUC2 etc. are selected from **edasp**.

- When you set up an experiment and define NUC1 in the routing table, BF1 is automatically read from the nucleus table. In the same way, BF2 is automatically read when NUC2 is defined etc. The routing table can be opened with **edasp** or by clicking NUC1 in **eda**. The nucleus table is created with the command **cf** that can be executed by the NMR Superuser. This command prompts you for the 1H basic frequency and then automatically calculates the basic frequencies for all other nuclei. For each nucleus, **cf** sets the basic frequency such that the most common reference substance for that nucleus would resonate at about 0 ppm. If you want to change the nuclei table, you can do that with the command **ednuc**. This is, for example, necessary if you are using a different reference substance for a certain nucleus.

> **i** Note, that if you execute cf and change the 1H basic frequency, you must click RESTORE in the nuclei table and execute **cfbsms**, after **cf** has finished.

CNST[0-31] - Array Of Constants Used In Pulse Programs

- Takes float values
- Can be set from **eda** by clicking **CNST ** Array **.**
- Can also be set by entering **cnst0**, **cnst1** etc. on the command line.
- Interpreted by the pulse program statements `cnst0`, `cnst1`, `cnst2` etc.
- The values of the parameter array CNST can be used as constants in a pulse program. For example, the pulse program line:

      "d2 = 1s/cnst2*2"

  Uses the value of CNST2 as a coupling constant. Note the difference between the pulse program statement `cnst2` and the TopSpin command **cnst2**. The latter is actually not a command but a way of setting the value of CNST[2].

CPDPRG1 - CPDPRG8 - Names of CPD Programs

- Take a character string value
- Can be set from **eda** by entering a name or by clicking the down arrow and then selecting a CPD program from the appearing list
- Can also be set by entering **cpdprg1**, **cpdprg2** etc. on the command line
- Interpreted by the pulse program statements `cpd1 - cpd8`, `cpds1- cpds8` and `cpdngs1 - cpdngs8`
- The values of CPDPRG1 - CPDPRG8 are the names of composite pulse decoupling (CPD) programs. The pulse program statements `cpd1` executes the CPD program defined by CPDPRG1, `cpd2` executes the CPD program defined by CPDPRG2 etc. In several Bruker CPD type parameter sets, the CPD program is specified by CPDPRG2 and executed on frequency channel f2. For example, the parameter set C13CPD contains the following settings:

      CPDPRG2 = waltz16

      PULPROG = zgpg30

  And the pulse program *zgpg30* contains the following line:

      d1 cpd2:f2

  The statements `cpd3` and `cpd4` which execute the CPD programs specified by CPDPRG3 and CPDPRG4, respectively, are often used in 3D experiments. Note, however, that the `cpd1 - cpd8` commands are equivalent and can be used to run any CPD program on any frequency channel.

cpds1 works like cpd1, except that it will execute the CPD program synchronously with the pulse program. This means cpds1 always starts the CPD program at the beginning, whereas cpd1 continues the CPD program where it was last stopped by the do statement. The difference between cpds2 and cpd2, cpds3 and cpd3 etc. is equivalent.

cpdng1 works like cpd1, except that the transmitter gate for channel f1 will not be opened. cpdngs1 works like cpds1, except that the transmitter gate for channel f1 will not be opened. For the channels f1, f2 ect. the statements cpdng2, cpdng3 etc are available.

The list of CPD programs which appears when you click the down arrow in **eda**, contains both Bruker and user defined CPD programs. Bruker CPD programs must be installed, once, with **expinstall**. You can set up your own CPD programs with the command **edcpd**.

D[0-31] - Array Of Delays

- Takes float values (seconds)
- Can be set from **eda** by clicking **D ** array ****
- Can also be set by entering **d0**, **d1**, **d2** etc. on the command line
- Interpreted by the pulse program statements d0 - d31, id0 - id31, dd0 - dd31, rd0 - rd31
- The pulse program statement d0 causes a delay of D0 seconds, d1 causes a delay of D1 seconds etc. In principle, all delays can be used for any purpose. In Bruker pulse programs, however, some conventions are followed. These are listed in the file Param.info that can be viewed with **edpul**. For example, D1 is used as a relaxation delay, D0 is used in combination with IN0 and ND0 as incrementable delay in 2D experiments. D0 and D10 are used as incrementable delays in 3D experiments. Note however, that all delays D1 - D31 are incrementable, not only D0 and D10. For more information click:

  Help => Manuals => [Programming Manuals] Pulse Programming Manual

DDR - Digital Digitizer Resolution

- Takes an integer value
- Temporary parameter calculated according to the equation:

$$DDR = {}^{2}\log(DECIM) + 1$$

- Cannot be set by the user
- DDR expresses the enhancement of the digitizer resolution by digital filtering. The total digitizer resolution, as defined by DR, is the sum of the hardware resolution (see table *Table 2.3 [▷ 22]*) and DDR.

DE - Pre-Scan Delay

- Takes a float value (microseconds)
- Can be set from **eda** or by entering **de** on the command line
- DE is executed as a part of the go statement. DE consists of 4 pre-scan subdelays DEPA, DERX, DE1 and DEADC. These subdelays start simultaneously at the beginning of DE and after each subdelay a certain action is performed:
  - DEPA: the preamplifier is switched from transmit to observe mode (default 2 μsec)
  - DERX: the receiver gate is opened (default 3 μsec)

– DE1: the intermediate frequency (if required) is added to the frequency of the observe channel. This corresponds to the execution of the `syrec` statement (default 2 μsec). The intermediate frequency is only used for AQ_mod = DQD or, if your spectrometers has an RX22 receiver, for any value of AQ_mod.

– DEADC: the digitizer is enabled (default 4 μsec)

DE can be set from **eda** or from the command line. The subdelays can be set with the command **edscon**. Their maximum value is DE - 1 μsec.

After DE, the digitizer starts to sample the data points.

For DIGMOD = analog, the parameter DE has a different purpose. It is used to achieve a near zero first order phase correction of the spectrum. In this case, DE does not consist of the above subdelays and is automatically adjusted when SW or DW are changed.

DECIM - Decimation Factor Of The Digital Filter

- Takes an integer value
- Cannot be set by the user
- Avance spectrometers use the concept oversampling which means that the data points are sampled much faster than specified by DW. This results in a larger number of points than specified by the user; a multiple of TD. Before the data are written to disk, they are digitally filtered which reduces (decimates) their number to TD. The decimation factor is defined by the following equation:

   DECIM = DW/DWOV

   where DWOV is the oversampling dwell time.

> **i** Note that DECIM can only take an integer value and DWOV must be greater than the minimum value for the current digitizer (see table *Table 2.3 [▶ 22]*). See also the acquisition status parameter DECIM.

DIGMOD - Digitizer Mode

- Takes one of the values analog, digital, homodecoupling-digital, **baseopt**
- Can be set from **eda** or by entering **digmod** on the command line
- In most standard parameter sets, DIGMOD is set to **digital** which means that oversampling and digital filtering is used. Oversampling means that the data points are sampled much faster that specified by DW. This results in a larger number of points than specified by the user; a multiple of TD. Before the data are written to disk, they are digitally filtered during which their number is reduced (decimated) to TD. For homodecoupling experiments on a Avance-AQX spectrometers, DIGMOD must be set to **homodecoupling-digital**. Digital filtering is then switched on but the amount of oversampling is smaller (a larger DWOV is used). For homodecoupling experiments on Avance-AQS spectrometers, DIGMOD must be set to **digital** because the reduction of the oversampling rate is not necessary. For DIGMOD = **analog**, digital filtering is switched off and analog filters are used. In that case, your Avance spectrometer works like a AMX/ARX spectrometer. However, since only a limited number of analog filter values is available for Avance, setting DIGMOD to analog is not recommended.

In TopSpin 2.0 and newer, DIGMOD can also be set to **baseopt** (for a description of this value, see parameter DSPFIRM).

DIGTYP - Digitizer Type

- Takes one of the values listed in the next table.
- Can be set from **eda** or by entering **digtyp** on the command line.

- DIGTYP must be set to the value which corresponds to the digitizer in your spectrometer. If you enter **digtyp** on the command line, you can choose from all digitizers which are available for Bruker spectrometers. However, if you click on DIGTYP in **eda**, only the digitizer(s) which exist in your spectrometer will appear. If you start your experiment with a Bruker standard parameter set (read with **rpar**), DIGTYP is usually set to the correct value. This is the value that was entered during the installation of the parameter sets with **expinstall**. If your spectrometer contains more than one digitizer, you might want to change the default value of DIGTYP.

> **i**   Note that the SADC digitizer cannot be used for sequential acquisition (AQ_mod = qseq).

digtyp	digitizer resolution (bit)	DWOV range (microseconds)
FADC (BC133)	12	0.05
HADC (HRD16)	16	2.5 - 5.0
SADC	16	3.325 - 6.65
HADC+	16	2.5 - 5.0
SADC+	16	3.325 - 6.65
IADC	16	0.1/0.05

Table 2.2: DIGTYP - Digitizer Type

DQDMODE - Sign Of The Frequency Shift During Digital Quadrature Detection

- Takes one of the values **add** or **subtract.**
- Can be set from **eda** or by entering **dqdmode** on the command line.
- DQDMODE defines the frequency shift applied in Digital Quadrature Detection mode as positive (add) or negative (subtract).

DR - Digitizer Resolution

- Takes an integer value.
- DR is the sum of the hardware resolution (see table *Table 2.3 [▶ 22]*) and the digital digitizer resolution DDR. It is automatically set to the maximum resolution of the current digitizer (DIGTYP). Because Avance spectrometers use the principle of oversampling, this value can be higher than the resolution of the digitizer.

  Usually, you want to use maximum resolution and keep this value of DR. In some cases, however, it is useful to set DR to a lower value. For example, if you want to acquire a large number of scans which might cause overflow for the maximum value of DR. However, to solve this problem, you can also set the parameter OVERFLW to **check** to halt the acquisition as soon as data overflow would occur.

DS - Number Of Dummy Scans

- Takes an integer value.
- Interpreted by the pulse program statement `go=n`, `gonp=n` and `rcyc=n`.
- Can be set from **eda** or by entering **ds** on the command line.

- Dummy scans are scans during which no fid is accumulated. Other than that, they are identical to normal scans, which means they take the same time (AQ) and perform phase cycling. Dummy scans are used to reach steady state conditions concerning T1 relaxation. This is necessary whenever the recycle delay of the experiment is shorter then 4 times the T1 value of the measured nucleus. Furthermore, they are used to establish a stable temperature. This is especially important in decoupling and TOCSY experiments where the irradiation high power increases the sample temperature. Dummy scans are performed if DS > 0 and the pulse program contains a `ze` statement before the `go=n` or `rcyc=n` loop. If a `zd` is used instead of `ze`, dummy scans are omitted.

DSLIST - Dataset List

- Takes a character array value.
- Can be set from **eda** by entering a name in the **DSLIST** field or by clicking the down arrow and selecting a name from the appearing list.
- Can also be set by entering **dslist** on the command line.
- Interpreted by the pulse program statements `wr #n, wr ##, ifp, dfp, rfp.`

DSLIST defines the name of a variable dataset list. Such a list can be created with **edlist => List type** : **ds** and has the following format:

```
sucrose 1 1 C:\ guest new
sucrose 2 1 C:\ guest new
fructose 1 1 D:\ guest old
```

Where the option **new/old** is used to delete/keep a possibly existing dataset.

TopSpin 2.1 and newer support the usage of blanks in the top level directory, data name and user name, by specifying them in double quotes, e.g.:

```
"sucrose low" 1 1 "C:/my documents" "John Smith" new
```

The list defined by DSLIST is interpreted by pulse program statements:

`wr ##` - stores the data in the dataset defined at the current list position

`wr #n` - stores the data in the dataset defined at list position `n`

`ifp` - increments the dataset list position

`dfp` -decrements the dataset list position

`rfp` - resets the dataset list position

where `n` = 1,2,3 etc.

DSPFIRM - Firmware Used For Digital Filtering

- Takes one of the values **sharp, user_defined, smooth, medium, rectangle**.
- Can be set from **eda**.
- DSPFIRM defines the filter function used for digital filtering. This determines the maximum spectral width that can be used. For high resolution experiments, DSPFIRM is usually set to **sharp**. The values **medium** and **smooth** are used for other applications. Note that **smooth** cannot be used for AQ_MOD = DQD. For DSPFIRM = **user-defined**, an external file is read from   <tshome>/exp/stan/nmr/lists/DSPFIRM.

In TopSpin 2.0 and newer, DSPFIRM can also be set to **rectangle**. The oversampled data are then filtered in such a way that the initial points of the FID are corrected. Setting DSPFIRM to **rectangle** will automatically set the parameter DIGMOD to **baseopt** and vice versa. In order to be able to correct the first points at the start of the FID, the exact position of the time 0 point must be known. This point is normally somewhere within the excitation pulse; for instance, for a normal 1-pulse-experiment with a 90° excitation pulse p0, it is at p0 * (1 - 2 / pi). However, this depends on the experiment. Therefore, a new

parameter ACQT0 was introduced in TopSpin 2.0, which can only be set within the pulse program. It measures the zero time from the beginning of the delay DE which is also the start of the `go` macro or the end of the excitation pulse. For the example given above, the following line has to be added at the beginning of the pulse program zg:

```
acqt0=-p0*2/3.14159;
```

This information enables the alignment of the data acquisition relative to the time 0 point of the FID. As always, the time is measured from left to right, and since this time is before the start of the FID, a negative time results. If this statement is missing, it is not possible to use the rectangle filter. As a result, the value of DE chosen by the operator may be temporarily prolonged by the program. The effects of this filter are:

– No first order phase distortion, so first order phase correction is not necessary.

– No so-called smilies (distortions of the spectrum at the left and right edges of the spectrum).

– The baseline of the resulting spectrum will be exactly 0 provided that no other effects distort the FID and that a correct zero order phase correction has been done.

– Signals at the very edge of the spectrum are not attenuated or distorted nor are they folded in.

The method needs some more internal memory, therefore, for larger TD, the memory on the DRU (RCU) may be a limiting factor.

DW - Dwell Time

• Takes a float value.

• Can be set from **eda** or by entering **dw** on the command line.

• Temporary parameter, calculated from the equation:

$$DW=10e6/(2*SW*SFO1)$$

• The dwell time is the time between the acquisition of two successive data points. Although it is normally calculated from SW, you can also set **dw**. In that case, the spectral width is adjusted according to the equation:

$$SW=10e6/(2*(0.05+DW)*SFO1)$$

When you set DW, you will often notice that the value you enter is slightly adjusted. The reason is that, when oversampling is used, the relation.

$$DW=DWOV*DECIM$$

Must be fulfilled and DECIM can only take integer values.

DWOV - Oversampling Dwell Time

• Cannot be set by the user.

• Automatically set according to DIGTYP and DW.

• Has a minimum value which depends on the digitizer (see table *Table 2.3 [▷ 22]*).

• The parameter DWOV reflects the principle of oversampling. This is used when DIGMOD = digital and means that the data are sampled much faster than specified by the user. In other words, a data point is sampled every DWOV µs rather than every DW µs where DWOV is only a fraction of DW. DWOV is set to the minimum value that can be handled by the digitizer or DSP-firmware (see table *Table 2.3 [▷ 22]*). Actually the value of DWOV is often a little above the minimum because the following relation must be fulfilled:

$$DWOV = DW/DECIM$$

Here, DECIM is the decimation factor that can only take integer values. For DIGMOD = analog (oversampling/digital filtering is switched off), DECIM is automatically set to 1 and DWOV is set to the value of DW.

EXP - Experiment Performed

- Takes a character array value.
- Is set by IconNMR.
- IconNMR sets EXP to the value of the parameter set that was used for the experiment.

FCUCHAN[0-8] - Routing Between Logical Frequency Channels and FCU's

- Array of integer values.
- Can be set from **eda** but is normally set graphically from the **edasp** window (see this command for more information).
- The values of FCUCHAN define the relation of the FCU's to the logical frequency channels. For example, FCUCHAN[1] = 2 means that FCU 2 is used for logical channel f1.

FIDRES - FID Resolution

- Takes a float value (Hz).
- A temporary parameter calculated from the equation:

    FIDRES=SW*SFO1/TD.

- Although FIDRES is normally calculated from SW, you can also set FIDRES. In that case, TD is adjusted while SW remains the same.

> **i** Note that the value that you enter for FIDRES if often adjusted a little. The reason is that TD is recalculated according to TD = SW*SFO1/FIDRES and rounded to the nearest power of two. FIDRES is then adjusted to fulfill the same equation.

FnMODE - Acquisition Mode Of The Indirect Directions (data ≥ 2D)

- Takes one of the values described below.
- Can be set with **eda** or by entering **fnmode**.
- Interpreted by the pulse program statement `mc`.

    The parameter FnMODE defines the acquisition mode of the indirect directions in a multi-dimensional experiment. Compared to `wr`, the `mc` statement simplifies the switching of the acquisition mode and allows you to use the same pulse program for various experiments. FnMODE can take the following values:

    - **undefined**

        This value must be used if the pulse program contains no `mc` statement.

    - **QF**

        Successive fids are acquired with incrementing time interval without changing any phase program. This corresponds to the `mc` clause F1QF or F2QF.

    - **QSEQ**

        Successive fids will be acquired with incrementing time interval and phases 0 and 90°. This corresponds to the `mc` clause F1PH or F2PH.

    - **TPPI**

        Successive fids will be acquired with incrementing time interval and phases 0°, 90°, 180° and 270°. This corresponds to the `mc` clause F1PH or F2PH.

    - **States**

        Successive fids will be acquired incrementing the time interval after every second fid and phases 0° and 90°. This corresponds to the `mc` clause F1PH or F2PH.

    - **States-TPPI**

Successive fids will be acquired incrementing the time interval after every second fid and phases 0°, 90°, 180° and 270°. This corresponds to the `mc` clause F1PH or F2PH.

– **Echo-Antiecho**

Special phase handling for gradient controlled experiments. This corresponds to the `mc` clause F1EA or F2EA.

For more information on the `mc` statement and the use of FnMODE click:

Help => Manuals => [Programming Manuals] Pulse Programming Manual

FQ1LIST - FQ8LIST - Irradiation Frequency Lists

- Take a character array value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a name from the appearing list.
- Can also be set by entering **fq1list**, **fq2list** etc. on the command line.
- Interpreted by the pulse program statements `fq1` to `fq8`.
- The parameters FQ1LIST to FQ8LIST define the names of frequency lists and are interpreted by the pulse program statement `fq1` to `fq8`. For example, the first time `fq1` is executed, the first value in the frequency list defined by FQ1LIST is read. The second time `fq1` is executed, the second value in this list is read etc. At the end of the frequency list, the list position is set back to the first value. Note that `fq1` can be executed multiple times because it occurs on several lines of the pulse program or because it occurs on a line that is part of a loop. In the same way `fq2` reads the list defined by FQ2LIST etc. The `fq1` to `fq8` statements must be preceded with a delay and followed by the frequency channel on which the pulse will be executed, for example:

```
d1 fq2:f2
```

In Bruker pulse programs, `fq1` is normally used on channel f1, and `fq2` on channel f2 but any combination is allowed. The delay must be greater than 2 µs. The frequency lists can be created or modified with the command **edlist**.

FW - Analog Filter Width

- Takes a float value.
- Can be set from **eda** or by entering **fw** on the command line.
- FW defines the width of the analog filter. For DIGMOD = digital/homodecoupling-digital, FW is automatically set for maximum oversampling. The value depends on the digitizer type, typically 125000 for HADC/2.

GP031 - Gradient Parameter Table

- Takes a list of real values (gradients) and character strings (filenames).

Index	GPX	GPY	GPZ	Filename
0	0.0	0.0	50.0	SINE.100
1	-50.0	-50.0	-50.0	RECT.1
2	75.0	75.0	75.0	GRADREC5m

Table 2.3: Gradient Parameters

- Can be set from **eda** by clicking **GP031 edit**.
- The gradients are interpreted by pulse program statements `gron0 - gron31` and `:gp0 - :gp31`.

- The filenames are interpreted by pulse program statements `:gp0` - `:gp31`.
- TopSpin allows you to use static gradients and shaped gradients. Static gradients have a constant strength during the time they are on. They are switched on by the pulse program statements `gron0` - `gron31`. These read the gradient strength for each direction from the GP031 table. The `groff` command switches the static gradients off. According to table *Table 2.3 [▶ 22]*, the pulse program section

  ```
  d21 gron2

  d22

  d23 groff
  ```

  would switch the X, Y and Z gradient on during the time D1+D2 with gradient strengths 75.0, 75.0 and 75.0, respectively. The gradient strength is expressed as a percentage of the maximum strength and runs from -100.0 to 100.0%. Static gradients do not use the **Filename** entry of the GP031 table.

  Shaped gradients have a strength that varies in time. They are switched on by the pulse program statements `gp0` - `gp31`. These interpret the **Filename** field of the gradient table. A file which is defined here contains a list of values between -1 and 1. Each value represents the relative gradient strength for a given time interval. They are multiplied with the values of GPX, GPY and GPZ to give the percentage of the maximum gradient strength for the respective direction. According to the table *Table 2.3 [▶ 22]*, the statement `p16:gp2` would switch on the X, Y and Z gradient on during the time P16 with gradient strengths 75.0, 75.0 and 75.0, respectively. The strength of each gradient would then vary in time according to the list of values in the file *gradrec5m*. When you click the down arrow to the right of each **Filename** field, a list of available files will appear. Such a list contains both Bruker and user defined gradient files. The former must be installed once with **expinstall**, with the option **Install Library Gradient Files** selected. The latter can be created with the *ShapeTool* (command **stdisp**). The gradient files reside in the directory:

  <tshome>/exp/stan/nmr/lists/gp

  The gradient parameters can also be set from the keyboard. For example, entering **gpx2**, **gpy2**, **gpz2** allows you to set the gradient strength for the three respective directions. With **gpnam2** you can set the shaped gradient file name.

GRDPROG - Gradient Program Name

- Takes an ascii string value.
- Interpreted by the pulse program statement `ngrad`.
- Standard gradient programs are delivered with TopSpin. They must be installed once, with the command **expinstall** with the option **Install Library Gradient Files** selected. The `ngrad` pulse program statement is mainly used on AMX/ARX spectrometers. On Avance systems, the `gron`/`groff` are normally used for gradient control. An exception is gradient shimming, where the `ngrad` statement is used.

HDDUTY - Homodecoupling Duty Cycle (in percent)

- Takes a float value.
- Can be set from **eda** or by entering **hdduty** on the command line.
- HDDUTY describes the ratio between the time used for homodecoupling and the time used for actual signal detection.

HPMOD - Routing Between High Power Amplifiers And Preamplifier Modules

- Array of integer values.
- Can be set from **eda** but is normally set graphically from the **edasp** window (see this command for more information).

TopSpin Parameters

HPPRGN - High Power Preamplifier Gain

- Takes one of the values normal or plus.
- Gain selection for spectrometers equipped with HPPR preamplifiers. By default, HPPRGN is set to **normal**. The value **plus** is only used for test purposes and should not be used for experiments.

INP[0-31] - Array Of Increments For Pulses P[0-31]

- Takes double values (µsec).
- Can be set from **eda** by clicking **INP ** array ****.
- Can also be set by entering **inp0**, **inp1**, **inp2** etc. on the command line.
- Interpreted by the pulse program statements `ipu0 - ipu31` and `dpu0- du31`.
- The pulse program statement `p0` executes a pulse with a length specified by P0. This length can be modified by the following pulse program statements:

    `ipu0` increments the pulse length by INP[0]

    `dpu0` decrements the pulse length by INP[0]

    `rpu0` resets the pulse length to the value of P[0]

In the same way, you can modify the length of the pulses executed by `p1` to `p31` with INP[1] to INP[31], respectively.

IN[0-31]- Array Of Increments For Delays D[0-31]

- Takes double values (sec).
- Can be set from **eda** by clicking **IN ** array ****.
- Can also be set by entering **in0**, **in1**, **in2** etc. on the command line.
- Interpreted by the pulse program statements `id0 - id31` and `dd0- dd31`.
- The pulse program statement `d0` causes a delay with a length specified by D[0]. This length can be modified by the following pulse program statements:

    `id0` - increments the delay by IN[0]

    `dd0` - decrements the delay by IN[0]

    `rd0` - resets the delay to the value of D[0]

In the same way, you can modify the length of the delays caused by `d1` to `d31` with IN[1] to IN[31], respectively.

In 2D dataset, IN[0] and ND[0] play a special role. In **eda**, they appear as the single parameters IN0 and ND0 in the F1 direction. They are determined by the following equation:

$$SW(F1) = 1/(SFO1 * IN0 * ND0)$$

Where IN0 is the spectral width and ND0 the number of occurrences of `d0` in the pulse program. If you change IN0 or ND0, SW is automatically recalculated. If you change SW, IN0 is recalculated and ND0 remains the same. You can set these parameters in **eda** or, from the command line, with:

in0

nd0

sw

Note that IN0 and ND0 only exist in the F1 direction but SW exists in F2 and F1.

In a 3D dataset, IN0 and ND0 play the same role in F1 indirect direction as they do in the 2D dataset. F1, however, is the third direction now. For the F2 indirect direction F2, the parameters IN10 and ND10 are used. In 3D, the following equations hold:

SW(F1) = 1/(SFO1 * IN0 * ND0)

SW(F2) = 1/(SFO1 * IN10 * ND10)

You can set these parameters in **eda** or, from the command line, with:

in0

nd0

in10

nd10

sw

Note that IN0 and ND0 only exist in F1, IN10 and ND10 only in F2 but SW exists in F3, F2 and F1.

L[0-31] - Array Of Loop Counters

- Takes integer values.
- Can be set from **eda** by clicking **L ** array **.
- Can also be set by entering **l0**, **l1**, **l2** etc. on the command line.
- Interpreted by pulse program statements `l0-l31`, `iu0-iu31`, `du0-du31` and `ru0-ru31`.
- The parameters L[0] - L[31] are read by the pulse program statements `l0 - l31`. These are mainly used in loop structures like `lo to x times l0` where `x` is a pulse program label; a number or a string which labels a previous line in the pulse program. An example of such a structure is:

```
4 (p1 ph1)
d2
..
lo to 4 times l3
```

The loop counter values can be varied as follows:

`iu0 - iu31` increment the loop counter used `l0 - l31` by 1.

`du0 - du31` decrement the loop counter used `l0 - l31` by 1.

Note that these increments and decrements only count during the execution of the current pulse program. They are not stored in the parameters L[0] - L[31]. Furthermore,

`ru0 - ru31` reset the loop counter used `l0 - l31` to L[0] - L[31]

The statements `l0 - l31` are also used in `if` structures (conditions). Two simple conditions are:

`if "(l3 != 0)"` : true if `l3` is unequal zero

`if "(l3 == 0)"` : true if `l3` equals zero

Further conditions are:

`if "(l3 operand expression)"`

where operand can be: ==, != , > , < , >= or <= and expression can be a number or an arithmetic expression built from pulses, delays and/or loop counters. The statements effected by a certain condition must be put between curly brackets. Furthermore, you can use the `else` structure for statements which must be executed if the condition is not true. An example is:

```
if "(l5 > 2)"
{
  p1 ph1
```

```
      }
      else
      {
      }
```

> ℹ️ Note that the syntax of the conditional statements is similar to C language syntax. However, you cannot use the C `"else if"` statement.

LOCNUC - Lock Nucleus

- Takes a character string value.
- Can be set from **eda** by entering a name in the **LOCNUC** field or by clicking the down arrow and selecting a nucleus from the appearing list.
- Can also be set by entering **locnuc** on the command line.
- Is interpreted by **edlock**, **lock**, **lopo**, **sref**.
- High resolution samples are usually locked on 2H or 19F. For these two nuclei, standard lock parameter files are delivered with TopSpin. These can be edited with the command **edlock**.

NBL - Number Of Blocks (Of Acquisition Memory)

- Takes an integer value.
- Interpreted by the pulse program statements `st`, `st0`, `ze`, `zd`, `wr`, `if`.
- The parameter NBL is used to acquire FID's in multiple memory blocks, for example in NOE difference experiments. For NBL = 1 (the default value), one FID (NS averages) is written to disk at the end of the acquisition. For NBL > 1, multiple FIDs are acquired in the acquisition memory before these are written to disk. The `st` statement increments the memory pointer by TD in order to use the next block. The statement wr #0 will write NBL FIDs to disk. The following pulse program statements interpret NBL:

  `st` - increment the memory pointer by TD

  `wr` - write NBL blocks to disk

  `st0` - set the memory pointer to the position of the first FID

  `ze, zd` - clear the acquisition memory of all NBL blocks

  `if` - increment the file pointer in the raw data file by NBL*TD

  `df` - decrement the file pointer in the raw data file by NBL*TD

  `nbl` - loop counter specifying the number of blocks

  If TD is not a multiple of 256 (1024 bytes), successive FIDs will still begin at 1024 byte memory boundaries. This is so for the FIDs in the acquisition memory as well as on disk. The size of the raw data file (*ser*) is therefore always a multiple of 1024 times NBL.

ND0 - Number Of Delays D0

- Takes an integer value.
- Temporary parameter.
- Only used in 2D and 3D datasets in the F1 direction.
- Number of **d0** statements in the increment loops of a pulse program for 2D or 3D experiments. Used to calculate the spectral width in the F1 direction according to:

  SW = 1 / (SFO1 * ND0 * IN0)

N0 is set to 2, if the evolution time contains a 180° pulse (e.g. HMQC, HMBC, HSQC). Otherwise, it is set to 1 (e.g. NOESY, COSY,ROESY,TOCSY).

ND10 - Number Of Delays D10

- Takes an integer value.
- Temporary parameter.
- Only used in 3D datasets in the F2 direction.
- Number of **d10** commands in the increment loops of a pulse program for 3D experiments. Used to calculate the spectral width in the F2 direction according to

    SW = 1 / (SFO1 * ND10 * IN10)

ND10 is typically set to 2, if the evolution time contains a 180° pulse. Otherwise, it is set to 1. Usually this is described in the pulse program comment section.

NS - Number Of Scans

- Takes an integer value.
- Interpreted by the pulse program statement `go=n`, `gonp=n` and `rcyc=n`.
- Can be set from **eda** or by entering **ns** on the command line.
- The pulse program statements `go=n` and `rcyc=n` loop NS times to the line which the label `n`. In Bruker pulse programs, the label `n` is usually 2. The acquired data are accumulated in memory. After NS scans, the pulse program continues with the next statement which is often `wr #0`. This statement writes the accumulated data to disk.

    After setting NS, you can calculate the resulting experiment time with the command **expt**. Then you can adjust NS such that the available time is properly used. Alternatively, you can set NS to a high value and halt the experiment (with the command **halt**) when time is up (see also DS and OVERFLW).

NUC1 - NUC8 - Nucleus For Frequency Channel f1 - f8

- Take a value from a predefined list of nuclei.
- Can be set from **eda** by clicking **Edit**.
- Can also be set from **edasp**.
- The parameter NUC1 assigns a nucleus to the frequency channel f1, NUC2 assigns a nucleus to the frequency channel f2 etc. In most routine experiments, only NUC1, NUC2 and NUC3 are used. For example:
    - 1D PROTON experiment without decoupling:
    NUC1 = 1H
    NUC2 = off
    NUC3 = off

    - 1D C13 experiment with 1H decoupling:
    NUC1 = 13C
    NUC2 = 1H
    NUC3 = off

    - 2D 1H experiment with 13C and 15N coupling:
    NUC1 = 1H
    NUC2 = 13C

NUC3 = 15N

In 2D datasets, NUC1 in the indirect direction (F1) must be set by selecting a nucleus from the listbox. In 3D datasets, this principle holds for both indirect directions, F2 and F1.

O1 - O8 -Irradiation Frequency Offset For Frequency Channel f1 - f8 in Hz

- Take a double value (Hz).
- Can be set from **eda** or by entering **o1**, **o2** etc. on the command line.
- Can also be set from **edasp** by adjusting OFSH1, OFSX1, OFSH2 etc.
- Can also be set from **gs** by adjusting **Offset** (FID display).
- O1 - O3 can be set by clicking the button ⚡ in the upper toolbar.
- O1 can be set by clicking the button ⚡ in the upper toolbar.
- The parameter O1 represents the irradiation (carrier) frequency offset. It is the center of the spectral region to be acquired. O1 is related to the basic frequency and the carrier frequency according to:

      O1 = SFO1 - BF1

The button ⚡ can be used if you are setting up your experiment from an existing spectrum. It sets the It sets O1 to the center of the currently displayed spectral region. The button ⚡ puts the cursor on the spectrum. You can then set the corresponding frequency offset by moving the mouse and clicking the left mouse button on a certain position. Because most Avance spectrometers are equipped with a maximum of four channels, O5 to O8 are normally unused. Therefore, they appear at the end of the **eda** table. See also O1P - O8P.

O1P - O8P - Irradiation Frequency Offset For Frequency Channel f1 - f8 in ppm

- Take a double value (ppm).
- Temporary parameters calculated from the equations:

      O1P = O1/BF1

      O2P = O2/BF2

      etc.

- Can be set from **eda** or by entering **o1p**, **o2p** etc. from the command line.
- Can be set by setting O1 - O8 (see these parameters).
- The parameter O1P represents the irradiation (carrier) frequency offset in ppm. Because chemical shift values are expressed in ppm, it is usually more convenient to set O1P rather than O1.

OVERFLW - Data Overflow Check

- Takes one of the values **check** or **ignore** (default is **ignore**).
- Can be set from **eda** or by entering **overflw** on the command line.
- For OVERFLW = **check**, acquisition commands check for possible data overflow.

> **i** Note that for OVERFLW = check, the performance of the spectrometer RCU is decreased and the smallest possible dwell times cannot be used any more. Thus, if no overflow is to be expected, you should set OVERFLW to ignore.

P[0-31] - Array Of Pulse Lengths

- Takes float values (μseconds).
- Can be set from **eda** by clicking **P ** array **.
- Can also be set by entering **p0**, **p1**, **p2** etc. on the command line.
- Interpreted by the pulse program statements `p0 - p31, ipu0 - ipu31, dpu0 - dpu31, rpu0 - rpu31`.
- The pulse program statement `p0` executes a pulse of P[0] μseconds, `p1` executes a pulse of P[1] μseconds etc. In principle, all pulses can be used for any purpose. In Bruker pulse programs, however, certain conventions are used. You can view the file that contains these conventions by entering the command **edpul param.info**.

PARMODE - Dimensionality Of The Raw Data

- Takes one of the values 1D, 2D,..., 8D.
- Can be set by changing the dimension from the parameter editor (**eda**) toolbar.
- Interpreted by **zg**, **rpar** and by all processing commands which access raw data (see Processing Reference Manual).
- The parameter PARMODE defines the dimensionality of the raw data. 1D-8D. It is interpreted by acquisition commands like **zg** and cross checked with the current pulse program. If the dimensionality of PARMODE and the pulse program are different, a warning will appear. If you want, you can still continue the acquisition.

  PARMODE is also interpreted by processing commands which access the raw data. If, for example, you enter **ft** on a 1D dataset, it is simply Fourier transformed. If however, you enter **ft** on a 2D dataset, you are first prompted to enter the FID number you want to Fourier transform. Processing commands which access processed data, like **abs**, interpret the processing parameter PPARMOD rather than the acquisition parameter PARMODE.

  If you change PARMODE and set it to a lower dimension, the unnecessary files are deleted. For example, if you change it from 2D to 1D the files *acqu2* and *proc2* are deleted. Furthermore, the processing status parameter PPARMOD is automatically set the chosen lower dimension. However, you are warned before this actually happens and you have the possibility of keeping all files.

  If you enter **rpar** to read a parameter set with a different dimensionality then the current dataset, a warning about this will appear. If you click OK, the data files and superfluous parameters files will be deleted. However, if you enter **rpar** with two arguments on the command line, i.e. **rpar <name> <type>**, this will happen without a warning. The reason is that **rpar** is often used in automation.

PHCOR[0-31] - Array Of Correction Angles For Phase Programs

- Takes float values (degrees).
- Can be set from **eda** by clicking **PHCOR ** array **.
- Can also be set by entering **phcor0**, **phcor1** etc. on the command line.
- Interpreted by the pulse program statements `ph0:r - ph31:r`.
- The option `:r` after a phase program statement in a pulse program adds phase correction factor to the phase. For example, the statement:

  ```
  p1 ph8:r
  ```

  executes a pulse with the current phase from phase program `ph8` plus the value of PHCOR[8.] For ph8 = 0 1 2 3 and PHCOR[8] = 2, the phase cycle would be 2° 92° 182° 272°.

TopSpin Parameters

PCPD[1-8] - Array of CPD Pulse Lengths

- Takes float values (μsec).
- Can be set from **eda** by clicking **PCPD ** array ****.
- Can also be set by entering **pcpd1**, **pcpd2** etc. on the command line.
- Interpreted by the CPD program command `pcpd`.
- The PCPD parameters represent pulse lengths for CPD decoupling. They are interpreted by the CPD command `pcpd`. Which PCPD pulse is used depends on the frequency channel on which the CPD program runs. For example, the pulse program statement:
  ```
  d1 cpd2:f2.
  ```
  Runs the CPD program defined by CPDPRG2 on channel f2. Therefore, a `pcpd` command in this CPD program will execute a pulse of length PCPD[2].

> **i** Note that the element PCPD[0] exists but cannot be used because there is no channel that corresponds to it.

PH_ref - Receiver Phase Correction

- Takes a float value (degrees).
- Interpreted by the pulse program statement `go=n phxx:r`.
- PH_ref adds a value to the receiver phase. For example, the pulse program statement:
  ```
  go=2 ph30:r
  ```
  starts the acquisition with receiver phase:

  ph30 + PH_ref

  The AU program **phtran** calculates the value of PH_ref for a 2D dataset from the spectrum phase correction values of a 1D row (for more information, type **edau phtran** and view the header of the AU program).

PL[0-31] - Array Of Power Levels

- Takes float values (dB).
- Can be set from **eda** by clicking **PL ** Array ****.
- Can also be set by entering **pl0**, **pl1** etc. on the command line.
- Can also be set from the **gs** dialog window.
- Interpreted by the pulse program statements `pl0`, `pl1`, `pl2` etc.
- The power levels PL[0] to PL[31] can be used to set the power for the frequency channels. The default power for channel fn is PL[n] (PL[1] for f1, PL[2] for f2 etc.) You can, however, explicitly assign a certain power level to a certain channel in the pulse program. For example, the pulse program statements:
  ```
  pl1:f2
  pl3:f4
  ```
  Set the power of channel f2 to PL[1] and the power of channel f4 to PL[3] Note the difference between the pulse program statement `pl1` and the command **pl1** entered on the TopSpin command line. The latter is not really a command but simply a way to set the parameter PL[1]. The parameters PL[0-31] can also be used to set the power of hard or shaped pulses in CPD programs. For example, the CPD program statement:
  ```
  p31:sp1:180 pl=pl1
  ```
  Sets the power of the shaped pulse `sp1` to PL[1].

POWMOD - Power Mode

- Takes one of the values **low**, **high** or **linear**.
- POWMOD defines the power mode for spectrometers equipped with a high power accessory. The value **linear** is unused.

PRECHAN - Routing Between Switchbox Outputs And Preamplifier Modules

- Array of integer values.
- Can be set from **eda** but is normally set graphically from the **edasp** window (see this command for more information).
- The values of PRECHAN define the connection between the switchbox outputs and the HPPR preamplifier modules. For example:

    PRECHAN[1] = 0 : Output 1 is connected to HPPR module 0

    PRECHAN[3] = 1 : Output 3 is connected to HPPR module 1

    PRECHAN[1] = 2 : Output 1 is connected to HPPR module 2

    PRECHAN[2] = 3 : Output 2 is connected to HPPR module 3

    For a standard HPPR configuration the module number correspond to the following units: 0 = 2H, 1 = X-BB, 2=1H, 3=User-Box and 4 = 19F. For more information on the HPPR preamplifier see the BASH spectrometer documentation.

PRGAIN - High Power Preamplifier Gain

- Takes one of the values **low** or **high**.
- Gain selection for spectrometers equipped with MSL preamplifiers. Normally, PRGAIN is set to **high**. The value **low** is only used for very strong NMR signals.

PULPROG - Pulse Program Used For The Acquisition

- Takes a character string value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a pulse program from the appearing list.
- Can also be set by entering **pulprog** on the command line.
- Also set by the command **edcpul <name>**
- Acquisition commands run the pulse program defined by PULPROG. If you set the pulse program in **eda**, you can click the down arrow to the right of the PULPROG field. A list of Bruker and user defined pulse programs will appear and you can click the one you need.

> **i** Note that Bruker pulse programs must be installed once with **expinstall**. If you start your experiment by reading a standard parameter set (with **rpar**), PULPROG is usually set to the appropriate pulse program.

QNP Nucleus Selection

- Takes one of the values 1, 2 or 3.
- If the current probe is set to a QNP probe (see **edhead**), acquisition commands will interpret the parameter QNP to switch the probe to the correct nucleus. QNP is a normally set with the AU program **qnpset**. Type **edau qnpset** to view this AU program.

RECCHAN[0-15] - Array Of Receiver Channels

- Takes integer values.

- Can be set from **eda** by clicking **RECCHAN ** Array **.

- RECCHAN enables the use of a different FCU than routed with **edasp** to generate the observe reference frequency. As such, the delay for the 22 MHz switching can be skipped.

RG - Receiver Gain

- Takes an integer value.

- RG controls the amplitude of the FID signal before it enters the digitizer. It is usually determined automatically with the command **rga**. This command performs an automatic determination of the optimum receiver gain. It runs several acquisitions with varying receiver gain until the maximum value is found that does not cause overflow. The parameter RG is then set to this value. If the RG value is already known from previous experiments, it can be set from **eda** or by entering **rg** on the command line. RG can also be set from the **gs** dialog.

  On AV-II spectrometers, the maximum RG value is 200 or 2000, depending on the spectrometer receiver. If you enter a larger value is entered it is automatically reduced to the maximum allowed value.

RO - Sample Rotation Frequency in Hz

- Takes an integer value.

- Can be set from **eda** or by entering **ro** or **ro <value>**.

- Interpreted by **ro acqu**.

- The command **ro acqu** will set sample rotation to the value of RO. It will wait for 60 seconds and then check if the specified rate has been reached. If this is not the case, an error message is displayed.
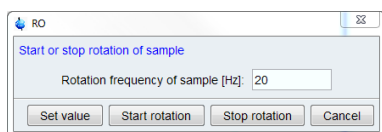
Figure 2.1: RO

RSEL - Routing between FCU's and Amplifiers

- Array of integer values.

- Can be set from **eda** but is usually set graphically from the **edasp** window (see this command for more information).

- The values of RSEL define the connections between the FCU's and the amplifiers. For example:

  RSEL[1] = 2 : FCU 1 is connected to amplifier 2

  RSEL[2] = 0 : FCU1 is not connected to any amplifier

  RSEL[0] is unused

SFO1 - SFO8 - Irradiation (carrier) Frequencies For Channels f1 to f8

- Take a double value (MHz).

- Are automatically calculated from the equation:

  SFO1 = BF1 + O1

  SFO2 = BF2 + O2

  etc.

  where O1, O2 etc. are set from **edasp** or **eda**

- can also be set with **gs** by adjusting **Frequency** (FID display)
- SFO1 can be set by clicking.
- The parameter SFO1 represents the iradiation (carrier) frequency for channel f1. It is usually set from the by defining the nucleus and frequency offset for channel f1 in the routing table (opened with **edasp**).

SP07 - Shaped Pulse Parameter Table

- Can be set from **eda** by clicking **SP07 edit**.
- Allows you to set power level, frequency offset, phase alignment and filename for 32 shaped pulses.
- Interpreted by pulse program statements like `sp0 - sp31`

Index	Power[dB]	Offset-Freq	Phase-Align	Filename
0	1.0	0.0	0.5	Gauss
1	20.0	0.0	0.5	Sinc1.1000
2	120.0	0.0	0.55	Q3.1000
3	..	..	..	..

Table 2.4: Parameters for Shaped Pulses

- When you open **eda** and click on **SP07 edit**, a list of shaped pulse parameters will appear as displayed in the table above.

  The table has 32 entries (index 0-31) which are interpreted by the pulse program statements `sp0 - sp31`. These occur on pulse program lines like:

      p1:sp2:f1

  This line interprets entry 2 of the table and execute a Q3.1000 shaped pulse on channel f1 with length P1, Power 120.0, Offset 0.0 and Phase 0.55. When you click the down arrow to the right of a **Filename** entry, a list of available shape files will appear. This lists contains both Bruker and user defined shape files. The former must be installed once with **expinstall**. The latter can be created with the *ShapeTool* (command **stdisp**).

  The SP07 entries are also available as acquisition parameters. They do not appear individually in **eda** but they can be set from the command line (see the next table).

  They also appear as acquisition status parameters when you enter **dpa**.

Description	Acquisition parameters	Commands
power level	SP[0-31]	**sp0** - **sp31**
frequency offset	SPOFFS[0-31]	**spoffs0** - **spoffs31**
phase alignment	SPOAL[0-31]	**spoal0** - **spoal31**
file name	SPNAM[0-31]	**spnam0** - **spnam31**

Table 2.5: Acquisition Status Parameters

SOLVENT - the Sample Solvent

- Takes a character string value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a solvent from the appearing list.
- Can also be set by entering **solvent** on the command line.

TopSpin Parameters

- Interpreted by **getprosol** and IconNMR automation.

- Also interpreted by **lock -acqu**, **lopo** and **sref**.

- The parameter SOLVENT must be set to the name of the solvent used in the current sample. Some acquisition parameters like pulse length and power level are dependent on the probe and the solvent. The command **getprosol** interprets SOLVENT and PROBHD and sets all dependent parameters accordingly. These parameters must be defined once, with **edprosol**, for all probes and solvents. In IconNMR automation, **getprosol** is automatically performed after a standard experiment has been read.

SW - Spectral Width in ppm

- Takes a double value (ppm).

- Can be set from **eda** or by entering **sw** on the command line.

- SW can be set by clicking the button in the upper toolbar.

- The spectral width should be set to such a value that all relevant peaks are expected to fall within that range. This means, for an unknown sample, SW should be set to a large value, like 50 ppm for a proton spectrum. The spectral width determines the dwell time according to the following equation:

$$DW=1/(2*SW*SFO1)$$

Where DW is expressed in µsec.

When you enter a certain value for SW, you may notice that it is slightly adjusted by TopSpin. The reason is that the digitizer hardware can only handle discrete values of DW. For DIGMOD = digital/homodecoupling-digital, the maximum allowed spectral width depends on the digitizer, the acquisition mode and the DSP firmware (see the next two tables).

Clicking the button in the upper toolbar will set SW to the region currently displayed on the screen. It will also set SFO1 to the frequency of the center of that region. For 2D and 3D experiments, SW as it is described above corresponds to the width in the acquisition direction. In the indirect directions, the spectral width are calculated from the parameters IN0, IN10, ND0, and ND10. In 2D, the following relations count:

$$SW(F1)=1/(SFO1*ND0*IN0)$$

$$SWH(F1)= 1/(ND0*IN0)$$

In 3D, the following relations count:

$$SW(F2)=1/(SFO1*ND10*IN10)$$

$$SWH(F2)= 1/(ND10*IN10)$$

$$SW(F1)=1/(SFO1*ND0*IN0)$$

$$SWH(F1)= 1/(ND0*IN0)$$

DSPFIRM DIGTYP	Sharp	Medium
FADC	25	100
HADC	25	25
HADC+	25	25
HADC2	25	100
SADC	18.75	18.75
SADC+	18.75	18.75
HRD16	25	25

DSPFIRM DIGTYP	Sharp	Medium
IADC	25	100

Table 2.6: Maximum SW (kHz) for AQ_mod = DQD

DSPFIRM DIFTYP	Sharp	Medium	Smooth
FADC	100	150	200
HADC	100	100	100
HADC+	100	100	100
HADC2	100	150	200
SADC	75	75	75
SADC+	75	75	75
HRD16	100	100	100
IADC	100	150	200

Table 2.7: Maximum SWH (kHz) for AQ_mod = qsim

SWH - Spectral Width in Hz

- Takes a double value.
- Can be set from **eda** or by entering **swh** on the command line.
- Is related to SW according to the following equation:

    SWH = SW * SFO1

    During experiment setup, SW is usually defined and SWH is automatically calculated from it. The maximum values depend on the digitizer, the acquisition mode and the DSP firmware as it is shown in table *Table 2.6 [▷ 34]* and in table *Table 2.7 [▷ 35]*.

> **i** Note that SWH is stored in the parameter file *acqu* as SW_h.

SWIBOX - Routing between Switchbox Inputs and Switchbox Outputs

- Array of integer values.
- Can be set from **eda** but is normally set graphically from the **edasp** window (see this command for more information).
- The values of SWIBOX define the connection between the swichbox inputs and switchbox outputs. For example, SWIBOX[1] = 3 means that Input 1 is connected Output 3.

TD - Time Domain; Number Of Raw Data Points

- Takes an integer value.
- The parameter TD determines the number of raw data points to be acquired. A large value of TD enhances the spectrum resolution, but also increases the acquisition time AQ. TD is usually set to a power of 2, for example 64k for a 1D spectrum. The FID resolution is related to the number of data points according to:

    FIDRES=SW*SFO1/TD

In a 2D experiment, TD in the acquisition direction (F2) has the same meaning as in 1D. In the indirect direction (F1), it represents the number of increments. As such, it is interpreted by pulse program statements like:

```
lo to n times td1
```

In a 3D experiment, TD in the acquisition direction (F3) has the same meaning as in 1D. In the indirect directions (F2 and F1), it represents the number of increments. As such, they are interpreted by statements like:

```
lo to n times td1 ; F1 loop in 2D or 3D experiments
```

```
lo to n times td2 ; F2 loop in 3D experiments
```

As an alternative to `lo to n times td1`, you can also use the `mc` statement. Do not confuse this with the TopSpin processing command **mc** that performs magnitude calculation.

TD0 - Loop Counter For One-Dimensional Experiments

- Takes an integer value.
- Interpreted by the pulse program statement `mc`.
- TD0 is normally used as a loop counter for multiple 1D experiments that are measured under varying conditions (for example varying temperature or pressure) and that are stored as 2D data. Similarly, it can be used for multiple 2D experiments that are stored as 3D data.

TE - Demand Temperature On The Temperature Unit

- Takes a float value.
- Can be set from **eda** or by entering **te** on the command line.
- Interpreted by **teset**.
- The command **teset** sets the temperature on the temperature unit to the value of TE. It be entered on the keyboard, or called from AU program with its macro TESET.

V9 - Maximum Variation Of A Delay

- Takes a float value (between 0.0 and 100.0 percent).
- Can be set from **eda** or by entering **v9** on the command line.
- Interpreted by pulse program statements like `d1:r, p1:r`.
- The pulse program statement `d1` causes a delay D1. The statement `d1:r`, however, causes a delay D1 plus a random value. As such, the delay is a different every time the statement `d1:r` is executed. The parameter V9 specifies, in percent, the maximum amount which is added to or subtracted from D1. As such, the effective delay varies between 0 and 2*D1. The `:r` option can be used for any of the statements `d0 - d31` and p0 - p31 to vary D[0-31] and p[0-31], respectively.

> **i** Note that the command **gs** ignores the `:r` option.

VALIST - Variable Amplitude (Power) List

- Takes a character array value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a name from the appearing list.
- Can also be set by entering **valist** on the command line.

- Interpreted by user defined pulse program statements.
- The parameter VALIST defines de name of variable amplitude (power) list. Such a list can be created with **edlist va** and has entries like:

    -6.0

    0.0

    3.0

    6.0

that represent attenuation values in dB.

The usage of a VA list is different from pulse and delays lists. You must define the statement by which a VA list is accessed in the pulse program. Such a statement can have any name, for example the name `vanam` is used in the examples below. The suffixes `.inc`, `.dex` and `.res` can be used to increment, decrement and reset the lists position, respectively. Furthermore, the caret operator (^) allows you to read a list value and increment the list position with one statement. The following pulse program entries illustrate the use of a variable amplitude list:

```
define list<power> vanam = <$VALIST>
```

definition of the power list

```
d1 vanam:f2 vanam.dec
```

set the power to the current value of the list and decrement the index

```
d1 vanam[2]:f3
```

set the power to the second value of the list

```
"vanam.idx = vanam.idx + 3"
```

increment the list index by 3

```
d1 vanam^:f4
```

set power to the current value of the list increment the index

As an alternative to using a list defined by the parameter VALIST, you can explicitly define a variable amplitude (power) list filename or even the list values in the pulse program. The following examples illustrates such definitions:

```
define list<power> vanam=<my_filename>
define list<power> vanam={10 30 50 70}
```

i	Note that the second definition does not require a list file. For more information on using variable amplitude lists click: **Help** => **Manuals** => [**Programming Manuals**] Pulse Programming Manual

VCLIST - Variable Counter List

- Takes a character array value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a name from the appearing list.
- Can also be set by entering **vclist** on the command line.
- Interpreted by the pulse program statements `lo to x times c`, `ivc`, `vcidx`.

- The parameter VCLIST defines de name of variable counter list. Such a list can be created with **edlist** => **List type** : **vc** and has entries like:

  4

  7

  20

  The currently defined list is interpreted by the pulse program statement: `lo to x times c`

  Where `x` is a pulse program label and `c` is the value at the current position of the counter list. When this statement is executed for the first time, the current position is the first entry in the list. The position is incremented by the statement `ivc`.

VDLIST - Variable Delay List

- Takes a character array value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a name from the appearing list.
- Can also be set by entering **vdlist** on the command line.
- Interpreted by the pulse program statements `vd`, `ivd` and `vdidx`.
- The parameter VDLIST defines de name of variable delay list. Such a list can be created with **edlist** => **List type** : **vd** and has entries like:

  10m

  50m

  2s

  Where m = milliseconds and s = seconds. The currently defined list is interpreted by the pulse program statement `vd` that reads the delay value at the current position. When `vd` is executed for the first time, the current position is the first entry in the list. The position is not incremented by `vd`; this is done by the statement `ivd`. As such, `vd` is normally used in combination with `ivd`. The statement `"vdidx=n"` sets the index to position `n` in the list.

VPLIST - Variable Pulse List

- Takes a character array value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a name from the appearing list.
- Can also be set by entering **vplist** on the command line.
- Interpreted by the pulse program statements `vp`, `ivp` and `vpidx`.
- The parameter VDLIST defines de name of variable pulse list. Such a list can be created with **edlist** => **List type** : **vp** and has entries like:

  10u

  50m

  2s

  Where u= microseconds, m = milliseconds and s = seconds. The currently defined list is interpreted by the pulse program statement `vp` that reads the pulse length value at the current position. When `vp` is executed for the first time, the current position is the first entry in the list. The position is not incremented by `vp`; this is done by the statement `ivp`. As such, `vp` is normally used in combination with `ivp`. The statement `"vpidx=n"` sets the index to position `n` in the list. A variable pulse list can only be used for hard pulses, not for shaped pulses or shaped gradients. As an alternative to a VP list, you can also specify a list of pulse values within the pulse program using a `define` statement. For more information on this topic click:

Help => **Manuals** => [**Programming Manuals**] **Pulse Programming Manual**

VTLIST - Variable Temperature List

- Takes a character array value.
- Can be set from **eda** by entering a name or by clicking the down arrow and selecting a name from the appearing list.
- Can also be set by entering **vtlist** on the command line.
- Interpreted by the AU program macros RVTLIST, VT, IVTLIST, DVTLIST.
- The parameter VTLIST defines de name of variable temperature list. Such a list can be created with **edlist => List type** : **vt** and has the following format:

  300

  320

  340

  Where each entry is a temperature value in Kelvin.

  Temperature lists are interpreted by the AU program macros:

  RVTLIST - open the temperature list defined by VTLIST

  VT - read the current value from the list and set it on the temperature unit

  IVTLIST - increment the current position in the list to the next value

  DVTLIST- decrement the current position in the list to the previous value

> ℹ️ Note that temperature lists are only interpreted by AU program macros, not by pulse program statements.

WBST - Number Of Wobble Steps

- Takes an integer value between 256 and 4096 (default 256).
- Can be set from **eda** or by entering **wbst** from on the command line.
- Can also be set by clicking ⟋ in the wobb window.
- Interpreted by **wobb**.
- The parameter WBST determines the number of steps (frequencies) used for tuning and matching a probe (wobbling). A probe is correctly tuned when the dip of the wobble curve is exactly at the center of the display. Normally, the default value of WBST (256) is high enough for exact tuning. If necessary, you can set WBST to a higher value for a better resolution.

> ℹ️ Note, however, that the maximum useful value is the screen resolution. Setting WBST to a higher value would not give you any advantage; it would only reduce the refresh rate. The command **atma** automatically calculates the optimum number of steps and does not interpret WBST. During tuning/matching **atma** temporarily sets the parameters WBST and WBSW to the calculated values and then resets them to their original values.

WBSW - Wobble Sweep Width

- Takes a double value between 1 KHz and 4 MHz.
- Can be set from **eda** or by entering **wbsw** from on the command line.
- Can also be set by clicking ▽ in the wobb window.

- Interpreted by **wobb**.
- The parameter WBSW sets the frequency range for tuning and matching a probe (wobbling). The center of the wobble region is determined by SFO1. When you change WBSW from the command line, **wobb** is automatically restarted.The command **atma** automatically calculates the optimum sweep width and does not interpret WBSW. During tuning/matching **atma** temporarily sets the parameters WBST and WBSW to the calculated values and then resets them to their original values.

ZGOPTNS - Acquisition (zg) Options

- Takes a character array value.
- Can be set from **eda** by entering **zgoptns** on the command line.
- The parameter allows you to set an option to acquisition commands like **zg** and **go**. As an alternative, acquisition options can also be specified on the command line or in the pulse program. For example, the option DQF can be set in the following three ways:
  - By setting the parameter ZGOPTNS to DQF
  - By specifying the option as an argument, e.g.: **zg –DDQF**
  - By defining the option in the pulse program, e.g.: `#define DQF`

2.5   Acquisition Status (dpa) Parameters

This paragraph contains a list of all acquisition status parameters with a description of their function. They are stored for each dataset in the file:

- <dir>/data/<user>/nmr/<name>/<expno>/

  acqus - acquisition status parameters

These can be viewed by entering **dpa** on the command line or clicking AcquPars tab in the data window and then clicking the **S** button. Some acquisition status parameters are interpreted by processing commands that work on raw data. Others are only stored as information for the user.

After an acquisition has finished, most acquisition status parameters have been set to the same value as the corresponding acquisition parameter. Sometimes, however, this is different. For example:

- Some parameters are continuously updated during the acquisition, e.g. NS, F1-TD (in 2D). When the acquisition is halted with **halt**, the current values are stored as acquisition status parameters.
- Some acquisition parameters are adjusted at the beginning of the acquisition, e.g. RG, FW, DR, SW. The modified values are stored as acquisition status parameters.
- The values of some parameters are a result of the acquisition. They cannot be set by the user (they do not appear as acquisition parameters) but they are stored as acquisition status parameters. Examples are AQSEQ, YMAX_a, LOCSHFT, NC.

The acquisition status parameters which are a result of or adjusted by the acquisition are listed below.

AQ_mod - Acquisition Mode

- Takes one of the values **qf**, **qsim**, **qseq** or **DQD**.
- Can be viewed from **dpa** or by entering **s aq_mod** on the command line.

- Normally, the acquisition status parameter AQ_mod is set to the same value as the acquisition parameter AQ_mod. If, however, AQ_mod = DQD but DIGMOD = analog, **zg** performs an acquisition in simultaneous mode and sets the status parameter AQ_mod to qsim. The same thing happens if your spectrometer is not equipped for DQD (see also the description of AQ_mod in chapter *Acquisition (eda) Parameters [▷ 13]*).

AQSEQ - Acquisition Order

- Used for datasets with dimension ≥ 3D.
- Takes one of the values **321**, **312** for 3D data.
- Takes one of the values **4321**, **4312, 4231**, etc. for 4D data.
- Can be viewed with **dpa** or by entering **s aqseq** on the command line.
- Interpreted by the processing command **tf3** and **ftnd**.
- AQSEQ describes the order in which the directions have been acquired. It is automatically set according to the loop structure in the pulse program. For example, a 3D pulse program usually contains a double nested loop with loop counters td1 and td2. If td1 is used in the inner loop and td2 in the outer loop, AQSEQ is set to 312. Otherwise it is set to 321.

  AQSEQ is evaluated by commands which process raw nD data, like **ftnd** or **tf3**. If the acquisition status parameter AQSEQ is not set, the processing parameter AQORDER is evaluated to determine the acquisition order.

BYTORDA - Byte Order of the Raw Data

- Takes one of the values **big** or **little**.
- Can be viewed with **dpa** or by entering **s bytorda** on the command line.
- interpreted by all processing commands which work on raw data.
- Big endian and little endian are terms that describe the order in which a sequence of bytes are stored in a 4-byte integer. Big endian means the most significant byte is stored first, i.e. at the lowest storage address. Little-endian means the least significant byte is stored first. TopSpin only runs on computers with byte order little endian. However, TopSpin's predecessor XWIN-NMR also runs on SGI workstations which are big endian. The byte order of the raw data is determined by the computer which controls the spectrometer and is stored in the acquisition status parameter BYTORDA (type **s bytorda**). This allows raw data to be processed on computers of the same or different storage types. The first processing command interprets BYTORDA, stores the processed data in the byte order of the computer on which it runs and sets the processing status parameter BYTORDP accordingly (type **s bytordp**). All further processing commands interpret this status parameter and store the data accordingly. As such, the byte order of the computer is handled automatically and is user transparent. 2D and 3D processing commands, however, allow you to store the processed data with a byte order different from the computer on which they run. For example, the commands **xfb big** and **tf3 big** on a Windows or Linux PC store the data in big endian although the computer is little endian. The processing status parameter BYTORDP is set accordingly.

DATE - Date Of Acquisition

- Takes an integer value (# seconds since 1970).
- Shows the date and time of the end of the acquisition specifying:

  `month day, year hour:minute:seconds time-zone`

  e.g. `November 10, 2004 6:06:19 PM GMT`

- The date is stored as an integer number, which expresses the number of seconds since January 1st 1970.

TopSpin Parameters

> Note that the start of the acquisition is not stored as a parameter but is available in the acquisition audit trail (command **audit acqu**).

DECIM - Decimation Factor Of The Digital Filter

- Takes an integer value.
- Can be viewed with **dpa** or by entering **s decim** on the command line.
- Interpreted by all processing commands which work on raw data.
- Also interpreted by **abs** and **apk**.
- Processing commands account for the group delay of digitally filtered data, by interpreting the following parameters:

    DSPFVS and DECIM (for DSPFVS < 20)

    or

    GRPDLY (for DSPFVS ≥ 20)

- DECIM must be interpreted by third party software which processes digitally filtered Avance data.
- For DSPFVS < 20, the commands **abs** and **apk** check whether DECIM is 1 (no oversampling) or greater than 1 (oversampling) to handle the so called smilies at the spectrum edges (see also the description of DECIM in chapter *Acquisition (eda) Parameters [▷ 13]*).

DTYPA - Data Type of The Raw Data

- Takes the value int or double.
- Can be viewed with **dpa** or by entering **s dtypa** on the command line.
- Interpreted by all processing commands which work on raw data.
- In TopSpin 2.0 and newer, 1D raw data are, if necessary, stored as double precision (64-bit) data. The spectrometer internal processor stores raw data in 32-bit integer format. If, during acquisition, data overflow would occur, data are transferred to the computer that controls the spectrometer where they are stored in 64-bit double precision format. The acquisition status parameter DTYPA shows whether data are stored as integers (DTYPA = 0) or doubles (DTYPA = 2).

> Note that processed data are always stored as integer data.

EXP - Experiment Performed

- Takes a character array value.
- IconNMR sets EXP to the value of the parameter set that was used for the experiment.

FnMODE - Acquisition Mode In The Indirect Directions (2D and 3D)

- Takes one of the values **undefined**, **QF**, **QSEQ**, **TPPI**, **States**, **States-TPPI** or **Echo-Antiecho**.
- Can be viewed with **dpa** or by entering **s fnmode** (2D).
- Interpreted by 2D and 3D processing commands that access raw data. These are usually **xfb** or **xf2** for 2D data and **tf3** for 3D data.

It is interpreted by processing commands to determine the Fourier transform mode in the indirect direction(s). For historical reasons, MC2 is interpreted when the acquisition status parameter FnMODE has the value **undefined**.

DSPFVS - DSP Firmware Version

- Takes an integer value between 10 and 23.
- Interpreted by processing commands that access raw data.
- Different DSP firmware versions filter the raw data in a different way. It is set according to the acquisition parameter DSPFIRM. Processing commands account for the group delay of digitally filtered data, by interpreting the following parameters:
- DSPFVS and DECIM (for DSPFVS < 20)

  or

  GRPDLY (for DSPFVS ≥ 20)
- DSPFVS must be interpreted by third party software which processes digitally filtered Avance data.

GRDLY - Group Delay

- Takes a double value ≥ 0.
- Interpreted by processing commands that access raw data
- Processing commands account for the group delay of digitally filtered data, by interpreting the following parameters:

  SPFVS and DECIM (for DSPFVS < 20)

  or

  GRPDLY (for 20 ≤ DSPFVS ≤ 23)
- GRPDLY must be interpreted by third party software which processes digitally filtered Avance data.
- 20 <= GRPDLY <= 23

LGAIN - Loop Gain; Lock Regulator Gain

- Is set to a value between -80 and 0 dB.
- Can be viewed with **dpa** or by entering **s lgain**.
- LGAIN is set at the end of the acquisition to the loop gain value used at that moment, i.e. the value currently set on the BSMS unit. This usually, but not necessarily corresponds to the value of LGain in the **edlock** table. For example, if lock-in was performed with the command **lock**, the loop gain is first read from the **edlock** table and set on the BSMS unit. However, pressing the **Autolock** or **Lock On/Off** key on the BSMS keyboard performs lock-in without first reading the **edlock** table.

> **i** Note in this respect that the current value of loop gain can also be changed from the BSMS keyboard (by pressing the two keys indicated with **MENU**) or by the command **lgain**.

LOCSHFT - Lock Shift

- Takes one of the values **true** or f**alse**.
- Can be viewed with **dpa** or by entering **s locshft**.
- The value of LOCSHFT indicates whether or not the sample was locked at the time the acquisition of the first scan has been finished.

LTIME - Loop Time; Lock Regulator Time

- Is set to a value between 0.001 and 1.0 seconds.
- Can be viewed with **dpa** or by entering **s ltime**.
- LTIME is a lock parameter (**edlock**) rather than an acquisition parameter (see the description of LGAIN above).

LFILTER - Loop Filter; Lock Regulator Cut-Off Frequency Of The Lowpass Filter

- Is set to a value between 1 - 200 Hz.
- Can be viewed with **dpa** or by entering **s lfilter**.
- LFILTER is a lock parameter (**edlock**) rather than an acquisition parameter (see the description of LGAIN above).

LOCKPOW - Lock Power

- Is set to a value between -60 and 0 dB.
- Can be viewed with **dpa** or by entering **s lockpow**.
- LOCKPOW is a lock parameter (**edlock**) rather than an acquisition parameter (see the description of LGAIN above).

MASR - MAS Spin Rate

- Takes a float value.
- Can be viewed with **dpa** or by entering **s masr**.
- Set by **masr get**.
- The acquisition status parameters MASR is continuously updateed when the MAS monitor is active (can be enabled with the command **set**).

NC - Normalization Constant

- Takes an integer value.
- Can be viewed with **dpa** or by entering **s nc**.
- Set by acquisition commands and by the processing commands **genfid**, **genser** and **addfid**.
- Interpreted by all processing commands that access raw data.
- Acquisition commands set NC to minus the value of DDR.

> **i** Note that for DIGTYP = analog, DDR is 0 and, as such, NC is also 0. The processing commands mentioned above create pseudo raw and NC is set according to the input processed data.

PROBHD - Probe

- Takes a character string value.
- Can be viewed with **dpa** or by entering **s probhd** on the command line.
- PROBHD is set at the end of an acquisition to the current probe as it was defined with **edhead** before the acquisition.
- LOCNUC

HOLDER - Sample Changer Holder Position

- Takes an integer value.
- Can be viewed by entering **s holder**.
- Set by IconNMR

AUTOPOS - Identification Information from BEST-NMR Rack Or Well-Plate

- Takes a character string value (A1-A12, B1-B12, ... , H1-H12).
- Can be viewed by entering **s autopos**.
- Set by IconNMR.

LOCKED - Lock Status During Acquisition

- Takes one of the values TRUE or FALSE.
- Can be viewed by entering **s locked** on the command line.
- The status parameter LOCKED indicates whether or not the magnetic field was permanently locked since the last successful **lock** command. For an nD experiment the parameter LOCKED is updated after the last FID has been acquired and stored to disk.

2.6　　Routing (edasp) parameters

The command **edasp** opens the routing table where you can select the nuclei and the spectrometer routing. When you select a nucleus for a certain frequency channel, the basic frequency and the default routing for that channel are automatically set.

Parameters Displayed In The Routing Table:

- BF1 - basic frequencies for channel f1.

  Same as the **eda** parameter BF1.
- NUC1 - nucleus for channel f1.

  Same as the **eda** parameter NUC1.
- SFO1 - irradiation frequency for channel f1.

  Same as the **eda** parameter SFO1.
- OFSX1 - irradiation frequency offset for the first X nucleus.
- OFSH1 - irradiation frequency offset for the first 1H.
- OFSF1 - irradiation frequency offset for the first F, 3H or Tl.
- OFSX2 - irradiation frequency offset for the second X nucleus.
- OFSH2 - irradiation frequency offset for the second 1H.
- OFSF2 - irradiation frequency offset for the second F, 3H or Tl.

  etc.

  When defined for channel f1, these parameters correspond to O1 in **eda**.

  When defined for channel f2, these parameters correspond to O2 in **eda**.

  etc.

> **i** Note that the command **edsp** reads the values for OF* that were stored by the previous **edsp** or **edasp** command. It the latter was performed on different dataset, the OF* values might differ from the corresponding O1, O2 etc. (see the description of **edsp**).

Parameters Which Can Be Viewed By Clicking PARAM

- FCUCHAN - connections between logical frequency channels and FCU's.
- RSEL - connections between FCU's and amplifiers.
- SWIBOX - connections between Switchbox inputs and Switchbox outputs.
- PRECHAN - connections between Switchbox outputs and Preamplifier modules.
- HPMOD - connections between high power amplifiers and Preamplifier modules.

edasp stores its parameters under the current dataset in the file:

<dir>/data/<user>/nmr/<name>/<expno>/.

acqus - acquisition parameters.

and the dataset independent file:

<tshome>/conf/instr/<instrum>

specpar - **edasp** parameters

> **i** Note that the routing table can also be opened from the eda dialog box, by clicking the NUC1 button.

2.7 Lock (edlock) Parameters

Lock parameters are used for locking the magnetic field and for referencing the spectrum. The lock-in procedure can be performed from the BSMS keyboard or with the command **lock** or **lopo**. Referencing the spectrum can be done with the command **sref**.

Lock Parameters Used For Locking The Magnetic Field:

- Lockfreq - lock irradiation frequency.
- Field - Field value (H0).
- BFREQ - Basic frequency.
- Solvent - Sample solvent.
- LPower - Lock power; the power used to irradiate the lock nucleus (-60 to 0 dB).
- LGain - Loop gain; lock regulator gain (-80 to 0 dB).
- LTime - Loop time; lock regulator time constant (0.001 to 1.0 seconds).
- LFilt - Loop filter, lock regulator cut-off frequency of the lowpass filter (1 to 200 Hz).
- LPhase - Lock phase; the phase of the lock signal.
- Nucleus - Observe nucleus.
- Distance - chemical shift of the lock nucleus (irradiation frequency offset).

Lock Parameters Used For Referencing:

- Ref. - chemical shift of the reference signal (default 0).
- Width - width of the region where the reference signal is searched.
- RShift - reference shift for default calibration.

The parameters LPower, LPhase, LGain, LTime and LFilt are probe and solvent dependent. They are stored for each probe and for each solvent separately in:

<tshome>/conf/instr/<instrum>/prosol/<probeID>/<solvent>/

bsmspar - probe dependent lock parameters

The other **edlock** parameters are only solvent dependent and are stored in the file:

<tshome>/conf/instr/

2Hlock - probe independent lock parameters

TopSpin Parameters

A

3 Spectrometer Configuration Commands

This chapter describes all TopSpin spectrometer configuration commands. These are commands which must be executed once, after installing a new version of the NMR Suite.

3.1 ampup

NAME

ampup - resets the controller board that controls the external amplifiers

DESCRIPTION

The command **ampup** resets the controller board that controls the external amplifiers. It can also be started from the menu as follows:

Click **Spectrometer => Accessories => Amplifiers => Transmitter power up**

ampup is available in TopSpin 2.0 and newer and partly replaces the command **acbdisp**, which no longer exists.

SEE ALSO

3.2 cf

cf - Configure the spectrometer

The command **cf** allows you to configure the spectrometer. It should be executed after installing a new version of TopSpin or if your spectrometer hardware has changed.

cf can be started from the command line or clicked from the menus:

- Spectrometer => Setup or
- Options => Spectrometer Tools

It starts with the Password request dialog window:
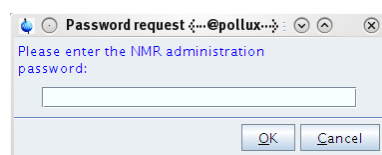
Figure 3.1: Password request

Spectrometer Configuration Commands

Enter the NMR Administrator password and click **OK**.

The cf window appears and displays the Select Spectrometer dialog box which shows all existing configurations. If it is the first TopSpin installation or TopSpin is installed in a different directory than the previous installation, only the standard configurations Bruker_default_avIII600, FOURIER300 and FOURIER60 (delivered with TopSpin) are shown. If TopSpin is installed in the same directory as a previous TopSpin installation, the existing configuration is shown and marked as selected (in our example spect).

To create a new configuration
- Click **New Spectrometer**, follow the dialog windows and finally click **Next**.

To delete an existing configuration
- Click **Delete Spectrometer**.

To use an existing configuration
- Select its entry and click **Next**.
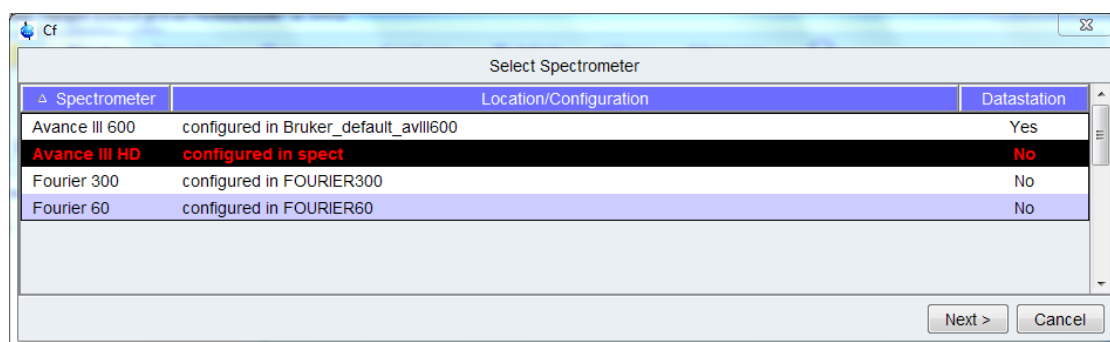
Figure 3.2: cf – Select Spectrometer

After clicking **Next** the dialog box **Edit Configuration Parameters** appears.
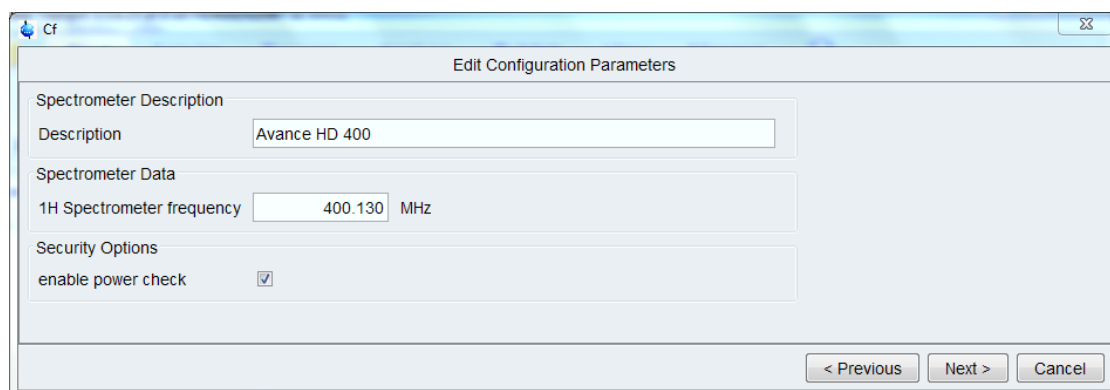
Figure 3.3: Edit configuration parameters

The *description* initially shows the type of spectrometer. You may change the description or add text to it.

Then enter the **1H spectrometer frequency** that corresponds to the strength of your magnet. Note that this frequency is typically, but not necessarily, one of the values 400.13, 500.13, 600,13 etc.

If the spectrometer supports power check then the dialog box shows the **enable power check** check box. Power check is enabled per default if a Cortab table exists, and disabled if no Cortab table exists (see command **cortab**).

Power check is a very useful TopSpin software tool to protect probes against pulses with too high power levels coming from the amplifiers. It compares peak power values. The power limits of the probes and the power of the pulses used by the pulse program are stored in TopSpin in order to be checked by power check:

- The maximum power levels for the probe are defined by edhead reading the values from the PICS of the probe. For order probes without PICS edhead has to be edited manually.
- The maximum amplifier output values are stored by cortab, which linearizes the amplifier for calculating all possible power levels for the pulse program as well.

Power check surveys the limits after starting the experiment but before executing the pulse program and displays an error message if any power level of the pulse program is too high for the probe.

The following error message is shown if peak power check is active, but no power limit is specified in the Cortab table for the respective nucleus:

Please note that a Cortab table must exist for the transmission path used and additionally the maximum amplifier output power must be defined for this transmission path. The respective value is stored in the corresponding Cortab table (for further information about Cortab please look up for the command **cortab**).

An icon in the Acquisition Status Bar shows if / that peak power check is active.

The **Power check** icon can be activated by left-click on

Options --> Preferences --> Acquisition Status Bar --> Include peak power check (POWCHK) indicator

in the TopSpin toolbar.

Click **Next** to continue. **cf** starts to scan for spectrometer components and shows each step in the center box (see next Figure ... where scans the components of an Avance III spectrometer).

Spectrometer Configuration Commands


```
start configuration on server
wait for server to start hardware configuration
get permission to start a new configuration
reinitialize objects
determine instrument name
create directories
read old configuration
edit parameters
send user input to server
wait for server to handle parameters
get permission to continue configuration
parse input from user
check for questions from server
check hardware
there are 6 DHCP controlled devices and 18 devices with fixed IP to check
try to connect 24 devices at the spectrometer subnet
connected: ELCB_Z100818_1629 at IP 149.236.99.20
connected: DRU1 at IP 149.236.99.89
connected: IPS0 at IP 149.236.99.254
BLA_W1345083_0005 at 149.236.99.252 is switched off or not installed
IPS0 at 149.236.99.251 is switched off or not installed
IPS0 at 149.236.99.250 is switched off or not installed
CORTABBOX_H118155_0001 at 149.236.99.242 is switched off or not installed
read configuration data from BSMS/2
Connecting to ipsoserver at IP 149.236.99.254 ... done
IPS0: connected to spectrometer subnet.
configure AQS racks
read configuration from AQS
read BIS from AQS_RX1 ... done
read all RG values from AQS_RX1 ... done
read BIS from AQS_SGU1 ... done
read BIS from AQS_SGU2 ... done
read BIS from AQS_SGU3 ...
```

Figure 3.4: Scanning for spectrometer components

Depending on the type of spectrometer and its hardware components a varying number of dialog boxes (all named **Edit Configuration Parameters**) appear which lead you through the remaining configuration. Here we explain the dialog boxes of a typical Avance III spectrometer configuration.

The first box to appear shows optional devices which cannot be detected by **cf**. Most of these devices are connected to RS (serial communication) channels. If a device is installed then the RS channel used to control it must be set accordingly. For devices which are not installed the corresponding RS channel is set to no. The entry **Lockswitch** specifies the router output channel of the amplifier to which the lock switch is connected (0 if the lock switch does not exist).

During initial configuration of a spectrometer the Bruker service engineer sets these parameters (at installation of the spectrometer). Changing these parameters is required only if a new device has been installed or if an existing device has been removed or is connected to a different RS channel. Otherwise the parameters need not be changed.
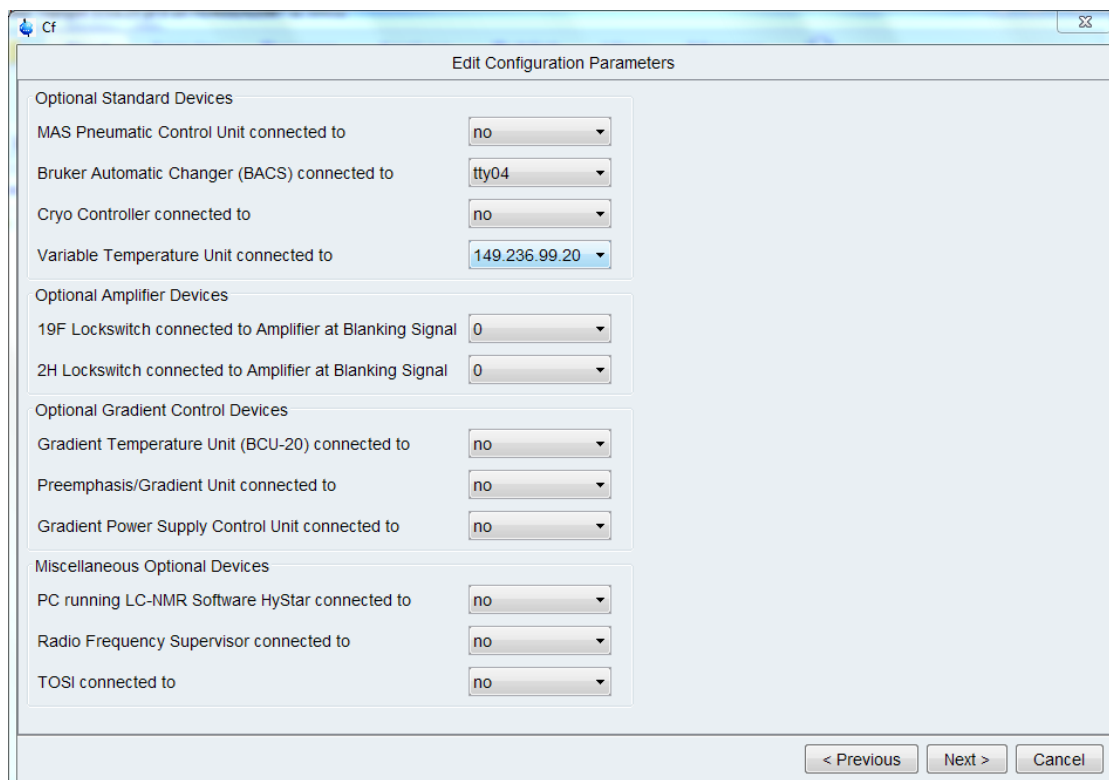
Figure 3.5: Optional devices

Click **Next** to continue.

The next dialog box shows the sample changer configuration. It is shown only if a sample changer is installed (here because its RS channel had been set to tty04 in the previous dialog box).

Figure 3.6: Sample changer configuration

Click **Next** to continue.

cf continues to configure the remaining components and shows each step in the center box (see the next figure where **cf** configures the remaining components of an Avance III spectrometer).

Figure 3.7: Configuring remaining components

When **cf** has finished the configuration of all components it shows the Summary of the spectrometer configuration in the center box. This allows you to check if your spectrometer hardware has been detected correctly. If some of the hardware is missing, please run **cf** again. If this does not change anything, run the spectrometer hardware checks.
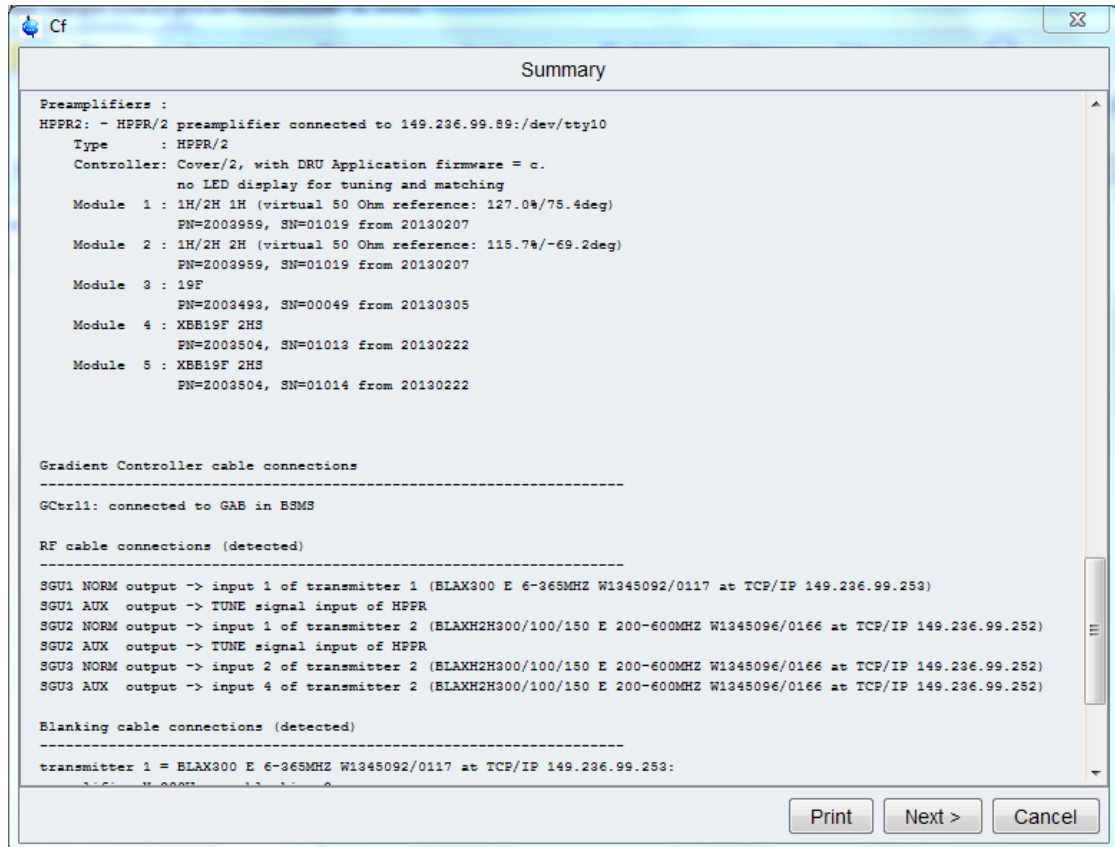
Figure 3.8: Configuration summary (uxnmr.info)

Click **Next** to continue.

The next dialog box shows the **Nuclei table**. It is displayed during initial configuration or if the basic spectrometer frequency had been changed before.

Normally, the default table is correct and you can just click *Next* to continue. If you want to get the default list, you must click **Restore**. If, for some reason, you want to change the nuclei table, you can do that here. Changes are automatically stored when you click **Next**.

Note that the nuclei table can also be changed at a later time with the command **ednuc**.

Figure 3.9: Nuclei Table

Click **Next** to continue.

cf shows the dialog window **Channel Routing**. It is displayed when the cable connections between amplifiers and preamplifiers are not detectable. Here, you can set the connections between Switchbox output and Preamplifier modules, or, if there is no Switchbox, between the Amplifiers and Preamplifier modules. As such, this step in **cf** corresponds to the command **edasp cf**. Note that in our example the Switchbox does not exist.

If the **Channel Routing** shows invalid routing connections, remove them by clicking **Clear cable connections**. Then define new lines according to you hardware cable connections and click **Save** and **Close**. If the routing table is correct, you can click **Close** to close it.
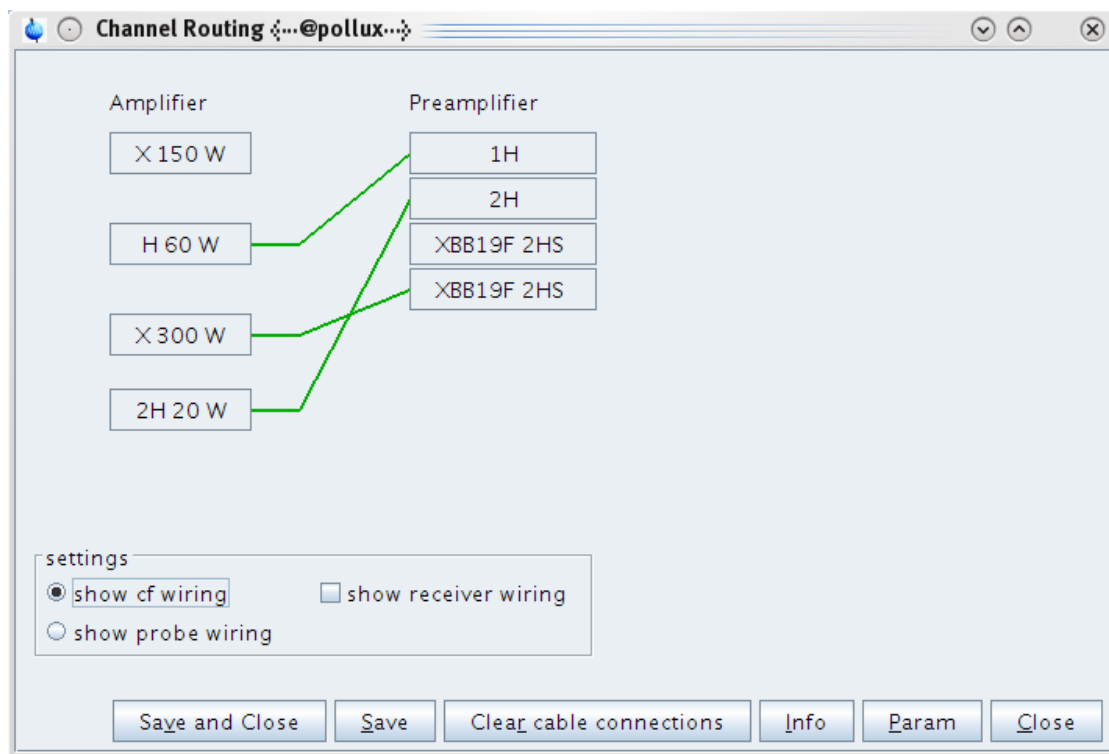
Figure 3.10: Channel Routing

Click on **Save and Close** or **Close** to continue.

Finally, **cf** shows the **Optional Setups** dialog box which shows a list of additional configuration programs (like **expinstall**, **edsolv**, ...). Click the respective button to execute these commands. After they have finished, click **Finish** to close the cf window.
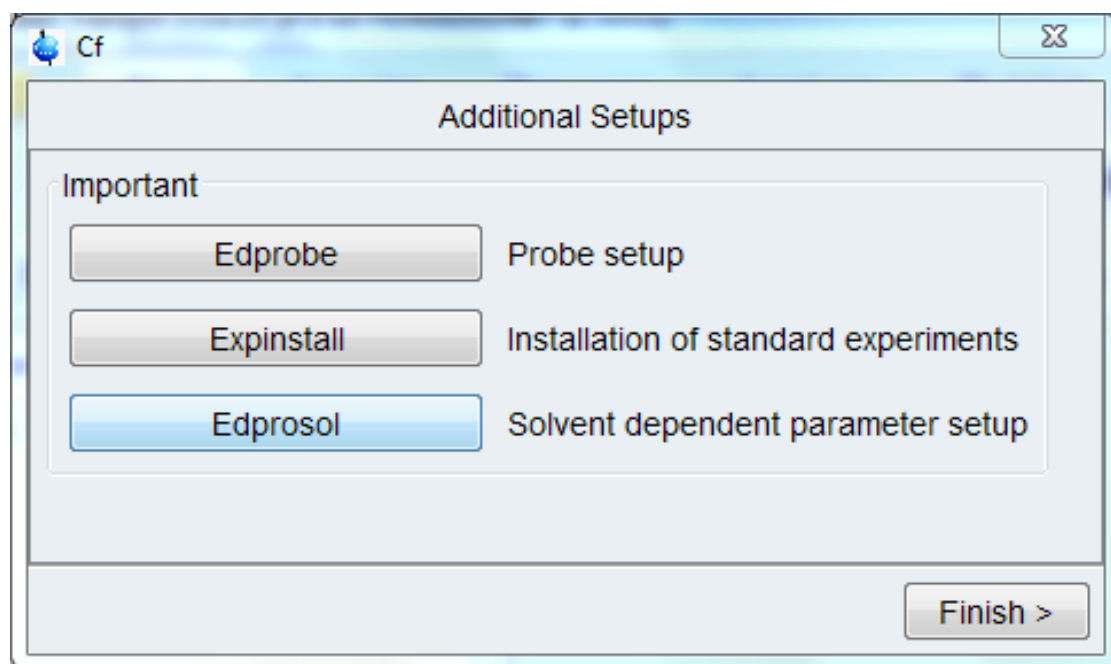
Figure 3.11: cf - Additional configuration programs

After finishing **cf**, please save the spectrometer configuration by copying the file *uxnmr.par* to floppy disk or CD and printing it out.

Creating a new configuration

To create a new configuration click on **New Spectrometer** in the Select Spec**trometer** dialog box. This opens the dialog window **Create new spectrometer configuration.**

Figure 3.12:

This dialog offers two ways to create a new spectrometer configuration:

To import an existing spectrometer configuration select **import**, use the radio box to choose between Datastation and Spectrometer, and then click **Browse** to open the Select dialog window. Browsing starts in your home directory, therefore it is advisable to store the folder containing the new configuration in your home directory. The Select dialog shows only folders, and if a folder contains a spectrometer configuration then the spectrometer type is shown in parantheses Use the Select dialog to browse to the folder which contains the existing spectrometer configuration, select it and click on **Select**.
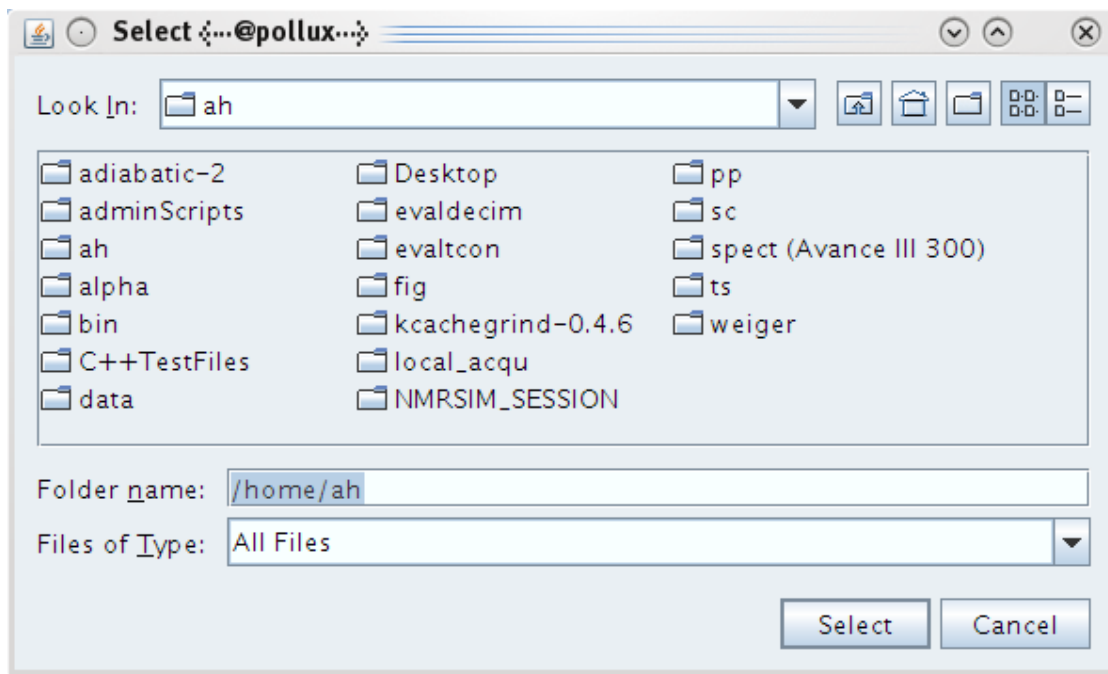
Click **OK** to open the dialog window. This is a browse dialog similar to the Select dialog described above. It opens the parent folder *TOPSPINHOME/conf/instr* and shows all folders. Chose a new folder name (no spaces allowed) and click **Save**.
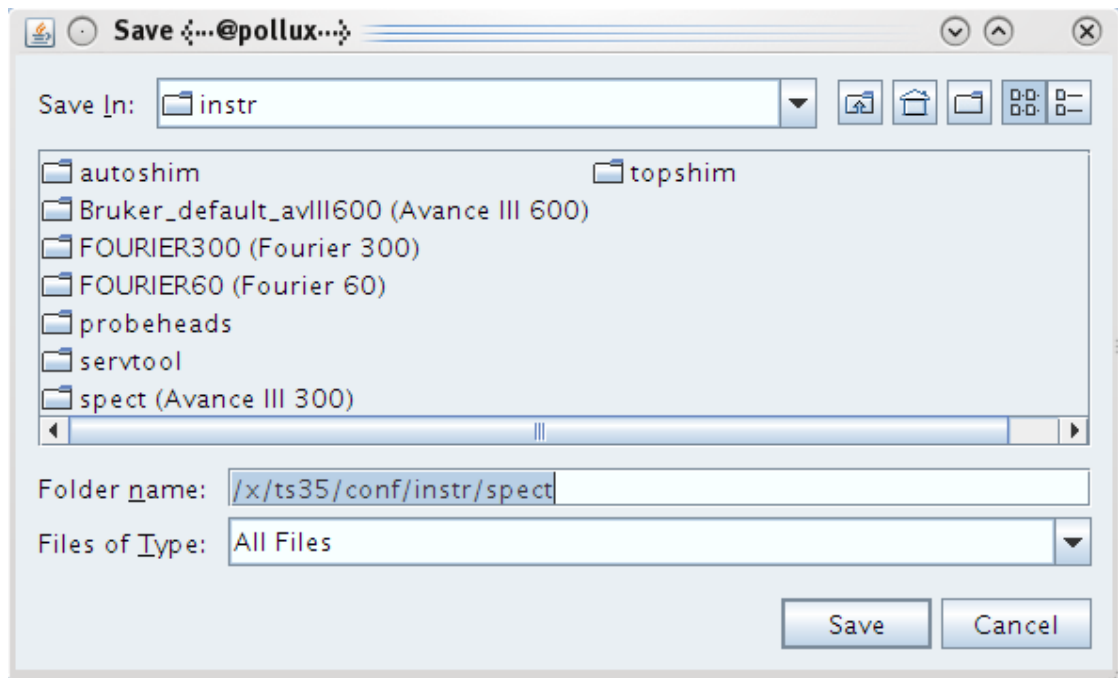
To create a new configuration from scratch select **local** and chose a name for the configuration (no spaces allowed, the name is also used for the folder containing the new configuration).

Note that if you specify an instrument name different from *spect*, this name must be specified in the hosts file:

- */etc/hosts* under LINUX
- *C:\WINNT\SYSTEM32\DRIVERS\ETC\HOSTS* under Windows

You must keep the name spect and append a white space plus the new instrument name, for example:

149.236.99.99 spect new_spectrometer

Then click **OK**. If a configuration using the same name already exists you will be asked to either delete it, or to cancel the operation.

The new configuration is now shown as selected in the Select Spectrometer dialog box. Click **Next** to continue the configuration.

See also

3.3 cortab

NAME

        cortab - Setup acquisition correction table

DESCRIPTION

The command **cortab** opens a window from which amplifier and receiver correction tables for acquisition can be created (see the next figure).
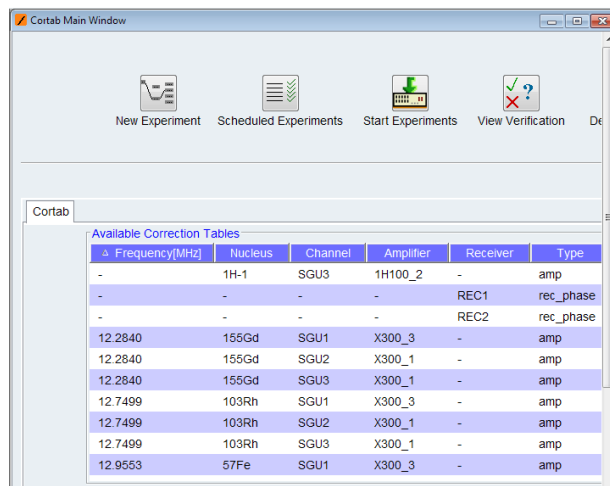
Figure 3.13: Cortab Main Window

Before this window is opened, you are prompted for the NMR Administration password. The correction tables are used to correct the non-linearity of the pulse power level versus the pulse length, and also to cancel out the influence of the receiver gain on the phase of the resulting spectrum. The amplifier linearization is performed in a range from -6 to above 75 dB and involves both amplitude linearization and phase correction, whereas the receiver phase correction is performed in a receiver gain range from 0.25 to above 1000. For the amplifier linearization, it is required that the amplifier output is connected to an external attenuator which, in turn, is connected to the receiver input. The Cortab program will inform you how to connect the cables and to which value you must set the attenuator. The result of an amplifier linearization is a correction table which is a list of correction values for the pulse amplitudes and phase values. It also contains a check sum which makes it possible to detect whether the file has been changed in any way other than by Cortab.

An example of a correction table is shown in the next table.

AMP_POWER	
# power	in dB correction in %
-6.000	0.000
-5.000	-32.644
-4.000	-38.510
-3.000	-37.361
-2.000	-35.032
-1.000	-31.672
0.000	-29.128
1.000	-24.939

2.000	-22.977
.. ..	
72.000	-0.008
AMP_PHASE	
# power in dB correction in degree	
-6.000	0.000
-5.000	-0.636
-4.000	0.622
.. ..	
72.000	4.323
# real probe input power in [W] (at -6dB)	
AMP_VALUE	
60.000000	0.00
# NAME_OF_TABLE:	amp2_1H_2
# DATE_OF_ORIGIN:	2010/01/05-14:23:22.356
##END	
# hash MD5: 99 A6 79 CA 61 B1 75 1D 2E D1 6C 79 84 FD 08 2F	

Table 3.1: Correction Table

Once a correction table exists, it is automatically used by acquisition commands like **zg** (this can be verified in the **edasp** window of the acquisition data set). If you want to check the accuracy of the amplifier linearization, Cortab allows you to do that by running a so called verification test. This test is always performed after the linearization procedure but can also be done in a separate experiment.

> **i** Please note that TopSpin can also handle correction tables created with XWIN-NMR. However, if you want to use software power check features, amplifier correction tables generated with TopSpin or XWIN-NMR 3.5 are required.

Amplifiers abbreviated 'BLA' - in case of AVI and AVII systems - and the multiplication unit of the Signal Generation Unit ('SGU MULT') requires a Cortab correction. Because the SGU appears earlier in the RF path, its correction table must be created first. The result of the SGU correction is then automatically used for the amplifier correction.

The receiver phase linearization, however, ("REC Phase Linearization") is independent and may be performed even if other correction tables already exist. It creates an additional table where theoretical and experimentally determined receiver gain values are listed (rec<x>_calibration.txt).

Amplifier linearization can be performed in two possible ways:

- The amplifier output is connected to the external attenuator.
- The amplifier output is connected to the HPPR preamplifier (as in an NMR experiment) and the preamplifier output is connected to the external attenuator.

The output of these two possibilities is not exactly the same. In general, the second one, involving the preamplifier, is used because it is a better simulation of the NMR experiment. However, some preamplifier modules the must not be used when a high power amplifier (> 500 W) is linearized. The Cortab module will inform you during the set-up of the experiment whether the preamplifier should be used.

Spectrometer Configuration Commands

The Cortab window allows you to set up the linearization experiments for various nuclei, perform these experiments and view the results. Correction tables that already exist are displayed in a list ('Available Correction Tables'). The list also contains a key word ('amp', 'pow', 'mma' , or 'rec') for each correction table, indicating whether the respective table belongs to an amplifier (amp), an SGU (mma), or a Receiver (rec). In case of an amplifier table the respective maximum amplifier output power (measured at the input of the probe) is also displayed. The 'pow' key does not represent a correction table but indicates that merely a maximum output power value is defined for a certain routing without the existence of a correction table. If an amplifier table is generated for the respective routing 'pow' will be replaced by 'amp'. In the reverse case, if an amplifier table is deleted, 'amp' will be replaced by 'pow' provided that the amplifier table contains a valid output power value.

A regular Cortab session involves the steps listed below. Each step is specified by the corresponding button in the main Cortab window.

New Experiment

Clicking this button opens a dialog box where you can define one or more Cortab experiments. Cortab experiments should be defined for each nucleus that you intend to use in your NMR measurements, an exception being the Receiver Phase Linearization which is frequency independent and needs to be performed only for one single (arbitrarily chosen) nucleus.

The following functions are available:

- BLA Linearization + Verification (this generates an 'amp' table)
- BLA Power Definition without Linearization (this generates an 'pow'table)
- SGU MULT Correction + Verification (this generates an 'mma' table)
- BLA Verification
- SGU MULT Verification
- REC Phase Linearization + Verification (this generates a 'rec' table)
- REC Phase Linearization

In case a Cortab box is installed some further experiments are possible (see description of Cortab box features below).

BLA and SGU (or REC) experiments must be executed in two separate series because they require different hardware connections. First you must set up one series for the SGU (and REC) functions and execute this series (see **Start Experiments** below). Then you can set up and execute a second series for the amplifier linearization.

---

i

Please note that the Cortab correction is mainly determined by the BLA linearization (see the *Table 3.1 [▶ 60]*). However, it is strongly recommended to perform the SGU linearization first (in case of AVI or AVII systems) in order to achieve optimum results. If you have a Router and you want to perform a MULT correction experiment, it is recommended that you connect the respective router output (e.g. RO1) to the external attenuator rather than the SGU output. This improves the quality of the resulting correction tables. The verification procedure is optional. It allows you to check the quality of the respective linearization results. It simply repeats the linearization but does not create a correction table. Instead, the existing correction table is used and the resulting amplitude and phase values are stored together with the calculated 'ideal' values in the verification files (see **View Verification** below). Thus, the verification files are directly correlated with the quality of the correction tables used.

Note that you can also run a verification procedure without doing a linearization first. This can be useful to compare the corrected with the uncorrected amplifier characteristics.

- **New Experiment => Select**

Opens the routing table (like **edasp**) where you can determine the appropriate hardware routing for the chosen experiment. In case of a REC phase correction, however, the complete routing is determined by the software and no user interaction is necessary. Please note that if you want to perform an SGU calibration, the complete and correct routing has to be chosen, though neither an amplifier nor a preamplifier are used for this experiment. After clicking the **SAVE'** button, the selected experiment (function) is added to the scheduled series. Instructions are displayed regarding the required hardware connections and the recommended value for the external attenuator is shown. Carefully follow the outlined instructions (e.g. to find the preamplifier must not be used) to prevent damaging the hardware units of the spectrometer. In one Cortab series, you can add:

experiments for various nuclei

one nucleus with various routings

linearization and verification

You cannot, however, set an SGU (or even a REC) correction and a BLA linearization in one series because they require different hardware connections. After having scheduled the experiment, you are prompted for the maximum amplifier output power. This value corresponds to the power (in Watts) measured at the output of the respective HPPR preamplifier module (i.e. at the input of the probe) when you are pulsing with maximum power and using the routing you have already specified. If this power is already known to the system, it will be displayed, otherwise zero is shown. A valid output value is crucial for any kind of output power limitation (e.g. to reduce the possibility of probe damage) of the acquisition software. In some special situations, e.g. if you want to use both the broadband input and the selective input of a probe for the same nucleus, it may be necessary to define a second amplifier power value here. In that case you may activate the „Additional power entry" button and enter an additional power value.

Note that in this situation the power values are associated with the respective preamplifier modules used, i.e. the amplifier value considered by the software depends on whether you use the XBB or the selective preamplifier module. However, if the respective power value is not known, you can either use zero (ignoring the power limitation possibilities of the acquisition software) or measure it with a wattmeter. In the following dialog window (Quick Experiment Setup) you will be able to (optionally) select a variable number of additional nuclei (together with their amplifier output power, as mentioned above) that are to be linearized with the same routing as the experiment you have already determined. This saves time because you only need to set up the routing table once, even if you want to generate a linearization table for multiple nuclei. The nuclei are selected by applying a user defined search pattern using the tokens ',' (separator) and '-' (nucleus range); for example, the search pattern "1H, 31P-29Si" would select both 1H and all the nuclei in the frequency range between 31P and 29Si. The system automatically stores the last ten search patterns so that they can be reused for further setups (e.g. with a different routing). It is also possible to edit the search patterns by editing the respective file *quick_exp_setup.txt* in the *cortab/etc* directory.

- **New Experiment => Finish**

Closes the dialog box and returns to the main Cortab window

Start Experiments

Clicking this button executes all experiments in the currently scheduled series. They are performed in the order in which they appear in the list. If a correction table already exists for the current experiment, it is renamed by appending the current date to its name. Its file will be overwritten if a further experiment with the identical setting is started at the same date. Please note that you must, temporarily, switch off the peak power limitation in order to run Cortab. To do that, run **cf** deactivating POWCHK. After Cortab has finished, run **cf** again activating POWCHK. At the beginning of the linearization, you may be prompted for the total external attenuation used. Cortab uses this value to generate scaled correction tables. This means the correction table of each SGU unit is scaled to the correction table of the 'weakest' SGU (provided that both nucleus and amplifier are the same). This may be of interest for

some experiments, but causes a small loss of the maximum output voltage of the 'stronger' SGUs. The scaled correction table files reside in the subdirectory *scale* and have the extension *.scale* and, as such, they are inactive. They can be activated by removing the file extensions and copying them into the Cortab directory.

Each linearization experiment involves the following steps:

- A receiver gain test

   This determines the optimum receiver gain for the current experiment. If this is successful, the experiment continues with the next step. If this is not successful, the experiment aborts and an information window will appear telling what to do (usually adjust the external attenuator). Just follow the instructions.

- The actual Cortab experiment, for example the BLA amplifier linearization.

In principle, you are ready now. The correction table has been created and it will automatically be used by the acquisition. If, however, you want to check the Cortab result, you can do that with the **View Verification** button (or - for a quick preview - by simply moving the mouse over the respective table entry in the Cortab Main Window), or simply double-click on an entry of the 'Available Correction Tables' list as described below.

View Verification

This button is only active when an entry is selected in the Available Correction Tables list of the Cortab window. Clicking this button (or double-clicking a list element) will then open a window where the results of the selected experiment are shown. The window consists of the following fields:

- Date of the experiments
- A list of dates at which Cortab experiments were performed.
- Name of the experiments

   When a date is selected, a list of result filenames created at that date is displayed. the respective file extensions denote the creation time of each test. For example:

   Amp_Power_res<1048> - theoretical and experim. amplifier output voltage

   (created 10:48)

   Amp_Phase_res<1048> - theoretical and experim. amplifier phase values

   (created 10:48)

When you click an entry, the contents of the corresponding file are shown. Note that files like Amp_Power_res do not contain the actual correction values but rather the theoretical 'perfect' values and the experimental values acquired on the base of the existing correction table. The smaller the difference between theoretical and experimental values, the better is the corresponding correction table. Therefore, the result files contain two simple statistical key values which make it easier to estimate the overall quality of the correction file (average deviation and standard deviation from the theoretical values). The smaller these values are, the better is the correction table used for the verification.

---

i Please note that verification files can be deleted to save disk space without affecting the correction tables they correspond to.

- A data field where the contents of the selected file are displayed.

Furthermore, the following push buttons are available:

- **View Graph**

   Opens a window with graphical and printable representation of the Cortab result.

- **OK**

  Closes the **View Graph** window and returns to the main Cortab window.

Edit Power

Clicking this button allows to modify the maximum output power of a selected correction table (see the next figure).
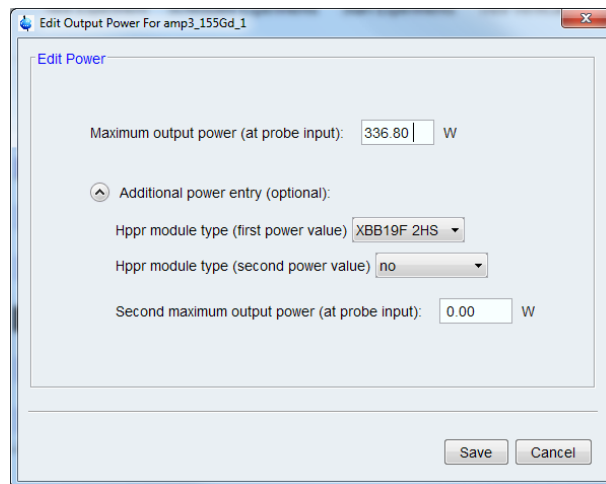
Figure 3.14: Edit output power

A window is opened where the new output power value may be added. There is also a possibility to add an additional power value in case different preamplifier modules are used for the same amplifier output.

> Please note that the output power always has to be measured with the identical transmitter path used for the real acquisition, i.e. the preamplifier must always be included for this measurement.

Scheduled Experiments

Clicking this button opens a dialog box where the currently scheduled experiments are shown. The dialog box contains the following buttons:

- **Remove All** : removes all experiments from the scheduled list.
- **Remove** : removes the selected experiment from the list.
- **OK** : exits the dialog box and returns to the main Cortab window.

Cortab experiments use two spectrometer channels simultaneously:

- The pulse channel: selected by the user in the Cortab New Experiment, i.e. the channel that is routed to the amplifier (or SGU) that is being corrected.
- The observe channel: automatically selected by the Cortab program. If the pulse channel is f1, the observe channel is f2. Otherwise the observe channel is f1.

USING A CORTAB BOX

If a Cortab box is installed some further experiments are available:

- BLA Linearization + Verification + Power Measurement (generates the 'amp' table and measures the maximum output power of the amplifier).
- BLA Power Measurement (measures the maximum output power of the amplifier).

Additionally, in the Cortab box Routing Scheme in the New Experiment Window (see the next figure), it has to be determined which amplifier is connected to which input of the Cortab box. After that it is then possible to set up and schedule experiments for all amplifiers selected in the Cortab box Routing Scheme since the Cortab box will switch the amplifier outputs automatically for each experiment.

Please note that, in case of high power amplifiers, the preamplifier my have to be circumvented to perform a BLA Linearization, but on the other hand it is always mandatory that the preamplifier is in the transmitter path for a correct power measurement, because the power measurement always has to be performed using a transmitter path identical to a 'normal' acquisition setup.

Depending on the type of amplifier and preamplifier used this can lead to a situation where it may not be possible to perform a BLA Linearization + Verification + Power Measurement procedure. Instead, one would have to perform a BLA Linearization + Verification procedure without preamplifier, and then insert the preamplifier to perform a BLA Power Measurement. The Cortab software will recognize such a constellation and display a respective error message.
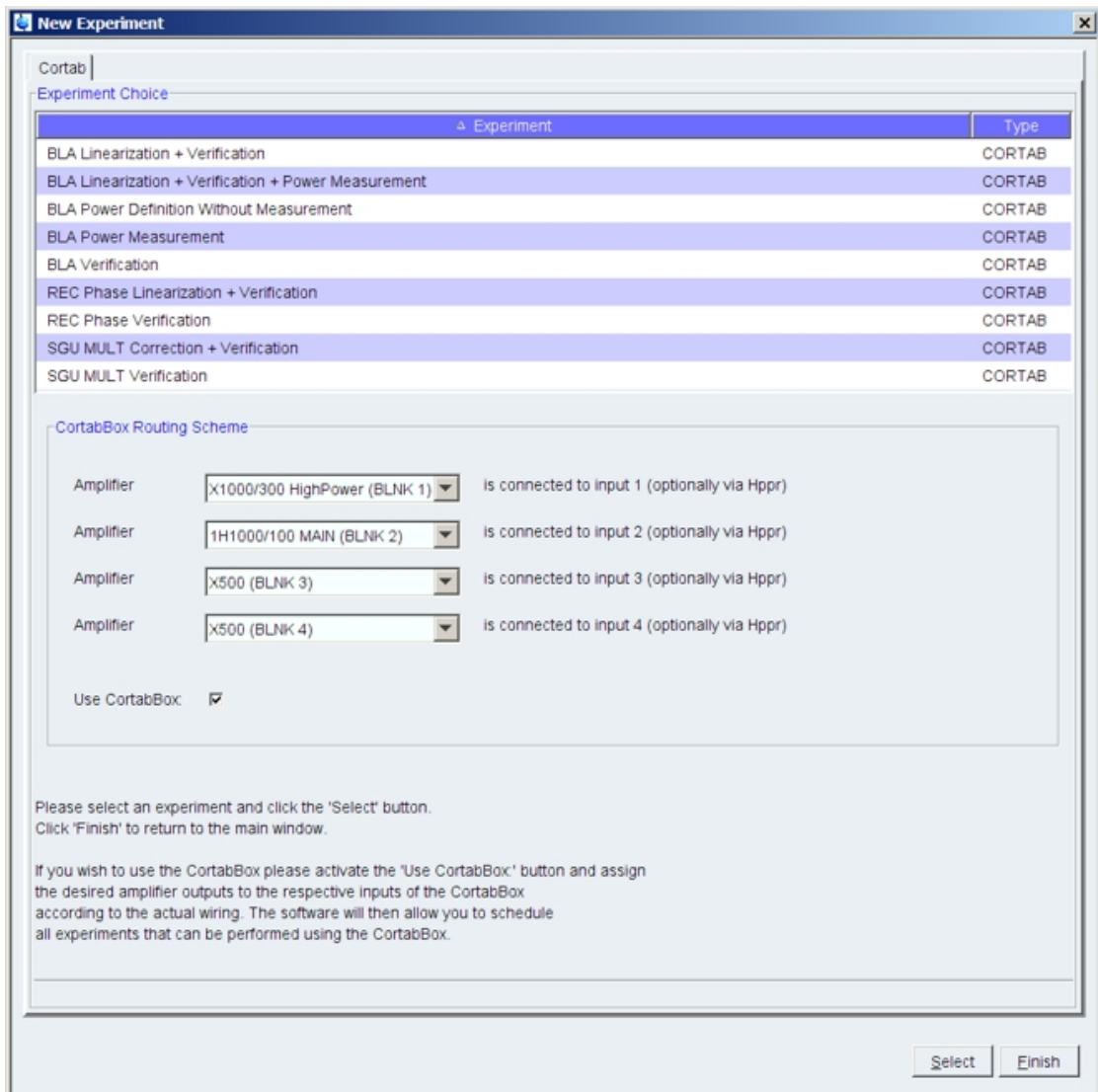
Figure 3.15: Cortab - New Experiment

INPUT FILES

- <tshome>/exp/stan/nmr/lists/pp.hwt
    - *zg_ctb* - cortab pulse program
- <tshome>/exp/stan/nmr/par.avance
    - XCTB - cortab parameter set (high resolution NMR)
- <tshome>/exp/stan/nmr/par.solids
    - XCTB - cortab parameter set (solid state NMR)

OUTPUT FILES

- <tshome>/conf/instr/<instrum>/cortab
    - *amp<fcu no.>_<nucl.>_<ampl. RO>* - amplifier correction tables where RO is the router output number.
    - *amp<fcu no.>_<nucl.>_<ampl. RO>.hp* - amplifier correction tables for high power amplifiers
    - *raw/amp<fcu no.>_<nucl.>_<ampl. RO>.raw* - raw data files; for internal use only
    - *scale/amp<fcu no.>_<nucl.>_<ampl. RO>.scale* - correction table standardized to 'weakest' SGU
    - *etc/amp_table* - table where amplifier output power is stored; for internal use only
    - *etc/quick_exp_setup.txt* – table where search patterns for Quick Experiment Setup are stored
    - *etc/rec<x>_calibration.txt* - table where theoretical and experimental receiver gain values are listed
    - *mma<fcu no.>_<nucl.>* - SGU MULT correction tables
    - *rec<x>_phase* - receiver phase correction table (corresponding to receiver <x>)
    - verification/<nucleus>/FCU<no>/<ampl. descr.>/<date>/
    - *Amp_Power_res* - theoretical and experimental amplifier output power (verification data)
    - *Amp_Phase_res* - theoretical and experimental amplifier phase values (verification data)
    - *Mult_Power_res* - theoretical and experimental SGU MULT output voltage (verification data)
    - *Mult_Phase_res* - theoretical and experimental receiver phase values (verification data)

SEE ALSO

zg [▶ 165]

3.4   ednuc

NAME

ednuc - Edit nuclei table

DESCRIPTION

The command **ednuc** opens the nuclei table showing a list of nuclei with their names, receptivity, spin and basic frequency (see the next figure).

Figure 3.16: ednuc - Nuclei table

The buttons of the nuclei table have the following functions:

- **Add/Edit**

  Edit a table entry or add a new one. A dialog is opened (see the next figure) where you can enter or change values. If you enter a new **Nucleus**, an entry is added to the list, otherwise the current entry is edited.

- **Delete**

  Delete the selected entry/entries.

- **Restore**

  Restore the original nuclei table. All changes you made will be undone. This must be done once, if you have changed the basic frequency with **cf**.

- **Save**

  Save any changes the nuclei table. You will be prompted for the NMR administration password.

- **Close**

  Close the nuclei table.

Figure 3.17: ednuc - Add new nucleus

> Note that double-clicking a table entry also opens the dialog you see in the figure above but only allows you to change the frequency.

INPUT FILES

- <tshome>/conf/instr/<instrum>/
  - *nuclei* - nuclei table
- <tshome>/exp/stan/nmr/lists/
  - *nuclei.all* - complete nuclei table (input and **Restore**)

OUTPUT FILES

- <tshome>/conf/instr/<instrum>/
  - *nuclei* - nuclei table

SEE ALSO

edsolv [▷ 74], *edhead [▷ 111]*, *edlock [▷ 91]*, *edprosol [▷ 69]*l

3.5   edprosol

NAME

edprosol - Edit probe and solvent dependent parameters

DESCRIPTION

The command **edprosol** opens a dialog box in which you can set the probe and solvent dependent (Prosol) parameters (see the next figure).

This is typically done during the installation of your spectrometer for all probes and solvents you want to use. However, you can also run **edprosol** again at a later time to set the Prosol parameters for additional probes and/or solvents.

Setting the Prosol parameters involves the following steps:

- Select the probe, solvent(s) and nucleus by clicking:
  - **Probe's Name**

    By default, the current probe (as defined with **edhead**) is selected. Prosol parameters must be defined for each probe separately.
  - **Solvent**

    The default value is **All.** If you keep that value, the same Prosol values will be stored for all solvents. If, however, you select a specific solvent, the Prosol parameters will be stored for that solvent only.
  - **Nucleus**

    By default, the nucleus of frequency channel f1 of the current dataset (NUC1) is selected. Prosol parameters must be set for each nucleus separately.
- This step is optional. You can enter two comment lines; for example the conditions under which the pulses/power levels have been determined (filters etc.).

> **i** Note that this feature is not visible in the next figure.

- Select the channels for which you want to define the parameters. You can do this by clicking one of the following radio buttons:
  - **F1+F2** : Prosol parameters for the f1 and f2 frequency channel.
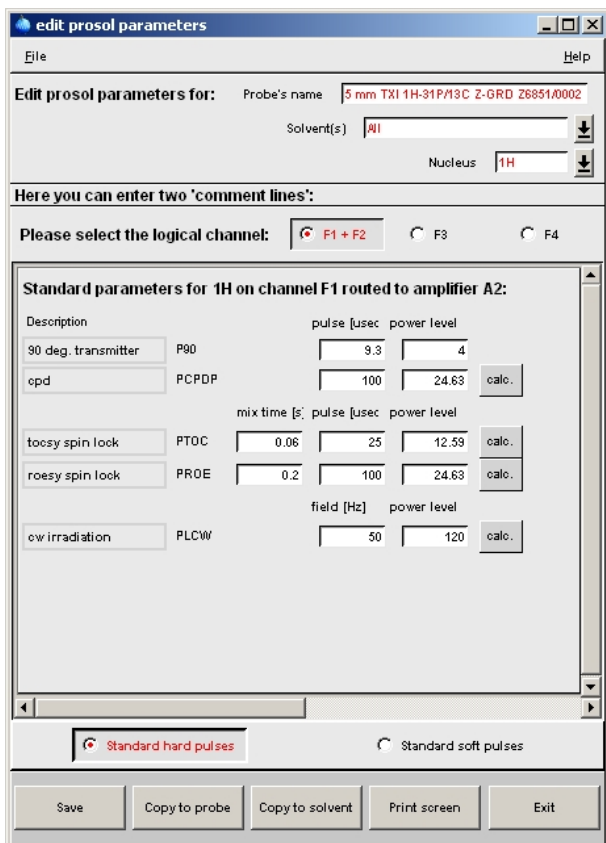
Figure 3.18: Edit prosol parameters

- **F3, F4**, **F5** etc. only appear if these channels exist and your spectrometer is configured as such.

- **Global** : a few Prosol parameters that count for all frequency channels.

By default, **F1+F2** is checked. Most experiments only use channel f1 or f1 and f2. If you are using f3, f4 etc. you have to set the Prosol parameters for these channels as well.

> **i**
>
> Note that f1, f2, f3 etc. refer to the logical frequency channels as you will see them in the routing table (**edasp**). They should not be confused with the terms that are used to indicate the directions of multidimensional dataset.

- Select the type of pulses that you want to set. For this purpose the following radio buttons are available at the bottom of the window:

  - **Standard Hard Pulses**

    Shows a list of standard hard pulses and allows you to set their pulse lengths, power levels and mixing times (tocsy and roesy only).

  - **Standard Soft Pulses**

    Shows a list of standard soft pulses and allows you to set their pulse lengths, power levels, phase alignments and shape files.

    By default, **Standard hard pulses** is checked which is sufficient for most experiments.

Set the pulse lengths and power levels for the selected frequency channel(s) and pulse type(s). You should start with the 90° hard pulse which must have been determined before you start **edprosol**. Then you can define all other pulses for the current channel. You can enter the pulse length and click the **calc** button to determined the corresponding power

level. Alternatively, you can enter the power level and click the **calc** button to determine the corresponding pulse length. This, however, only works if the pulse length is set to 0 at the time **calc** is clicked.

ℹ️ Note that the relation used by **calc** is determined by the 90° hard pulse.

- Save the Prosol parameters by clicking the **Save** button at the bottom of the dialog box. To the right of the **Save** button, you will find the following buttons:
  - **Copy to Probe**

    Open a list of all probes. You can select the probe(s) for which you want to store the Prosol parameters and then click one of the buttons

    Save to all solvents or **Save to selected solvents**.
  - **Copy to Solvent**

    Open a list of all solvents.You can select the solvent(s) for which you want to store the Prosol parameters and then click **Save** to store it.
  - **Print Screen**

    List the currently selected Prosol parameters on the printer.
  - **Exit**

    Quit the **edprosol** dialog box.

The **edprosol** dialog box provides a few additional options if you switch to expert mode. You can do that by clicking **File => expert mode**. Right above the parameter table, an arrow button will appear where you can select the amplifier connected to the currently selected channel. The default amplifier is normally correct. At the bottom of the window, two extra radio buttons will appear:

- **User-Defined Hard Pulses**

  Shows a list of user-defined pulses and allows you to set their pulse length and power level.

- **User-Defined Soft Pulses**

  Shows a list of user-defined pulses and allows you to set their pulse length and power level.

  User defined hard and soft pulses require a separate (user defined) relations file. It can be setup from the Windows Explorer or from a Linux shell in the directory

      <tshome>/conf/instr/<instrum>/prosol/<probeID>/relations

  Just copy the standard relations file and modify it to your needs.

The **edprosol** window allows you to set up a probe dependent tune file. You can do that by clicking **File => Edit tunefile**. On first time execution, this command displays the example tune file that is delivered with TopSpin. When you save the file, it is stored for the probe that is currently defined in **edprosol**. Probe dependent tune files are read by the command **tune .sx**.

The Prosol parameters are interpreted by the **getprosol** command that copies them to the corresponding acquisition parameters. The default relations between Prosol and acquisition parameters are listed in table *Table 7.1 [▶ 132]*.

ℹ️ Note that **getprosol** is automatically performed as part of the IconNMR automation.

Note that the probe and solvent dependent lock parameters are set with **edlock**.

INPUT FILES

- \<tshome>/exp/stan/nmr/lists/group/

  example - example tune file (input for first **File => Edit tunefile**)

- \<tshome>/prog/tcl/xwish3_scripts

  edprosol - Tcl/Tk script that is started by **edprosol**

- \<tshome>/prog/tcl/libtix/prosol/lib/lists

  routing - default spectrometer routing

INPUT AND OUTPUT FILES

If you select Solvent(s) **All**:

- \<tshome>/conf/instr/\<instrum>/prosol/\<probeID>/

  – *nucleus.channel.amplifier* - standard prosol parameters

  – *params* - global (channel independent) prosol parameters

  – *tunefile* - probe dependent tune file (input for **File => Edit tunefile**)

If you select a specific solvent:

- \<tshome>/conf/instr/\<instrum>/prosol/\<probeID>/\<solvent>

  – *nucleus.channel.amplifier* - standard prosol parameters

  – *params* - global (channel independent) prosol parameters

  – *tunefile* - probe dependent tune file (input for **File => Edit tunefile**)

SEE ALSO

getprosol [▷ 131], *edlock [▷ 91]*

3.6 edscon

NAME

edscon - Edit spectrometer constants

DESCRIPTION

The command **edscon** opens a dialog window, where you can set certain spectrometer parameters (constants) (see the next figure).
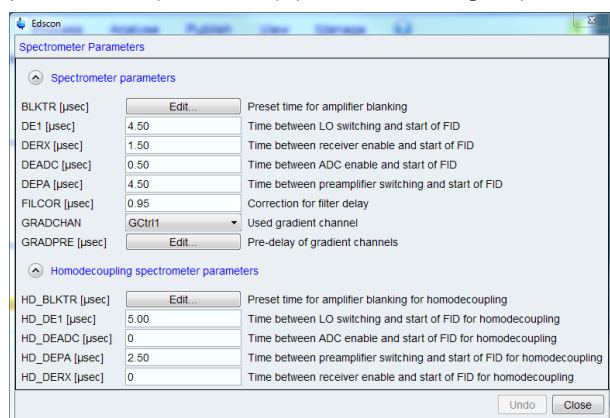
Figure 3.19: edscon

The term constant refers to the fact that these parameters count for all datasets. **edscon** must be executed once as part of the spectrometer configuration.

Changes in the **edscon** parameters can be stored by clicking **OK**. You will be prompted for the NMR Administrator password.

BLKTR is an array of amplifier blanking preset times. This means they only allow RF signals to pass during the time they are blanked. Because of the finite switching time, blanking is triggered before the start of the RF pulse. The amplifier is blanked BLKTR μsec before the pulse. It is unblanked (allows no further RF passing) at the end of the pulse.

The use of the **edscon** preset parameter BLKTR can be switched off by inserting the statement

```
preset off
```

at the beginning of a pulse program. This has the same effect as setting all elements of BLKTR to zero. In this case, the blanking steps described above occur at the beginning of the RF pulse. Nevertheless, you can enable the preset blanking for each individual channel, e.g.:

```
2μ gatepulse 1 ;enable blanking for channel f1
```

```
2μ gatepulse 1|2 ;enable blanking for channel f1 and f2
```

In this example, the blanking of the transmitter and preamplifier is triggered 2 μsec before the RF pulse.

The **edscon** dialog box also shows the so called pre-scan subdelays. These are all part of the pre-scan delay DE. This is a hidden delay (it is not specified in the pulse program) that is automatically introduced by the `go` statement.

DE consists of 5 pre-scan delays DE1, DERX, DEADC, DEPA and DE itself. The sub-delays DE1, DERX, DEPA and DEADC are global parameters which depend on the properties of the respective spectrometer hardware. DE is also set globally but can be changed temporarily by defining a value in the pulse program. DE mostly depends on the dead-time of the probe. It starts with the "go = .." instruction of the pulse program. All delays end simultaneously at the beginning of the data sampling. They act as pre-delays to the data sampling. This is different from the behaviour of TopSpin 1.3 and XWIN-NMR. With these older software versions all delays started simultaneously as post-delays to the go instruction and the data sampling started together with the end of the longest delay DE. Therefore, the values of the sub-delays have changed with TopSpin 2.0. Updating from an older software version to TopSpin 2.0 will result in an update of the edscon parameters to default values. At the end of each subdelay a certain action is performed:

- DE1: the intermediate frequency (if required) is added to the frequency of the observe channel. This corresponds to the execution of the syrec statement (default 2 μsec). The intermediate frequency is only used for AQ_mod = DQD or, if your spectrometers has an RX22 receiver, for any value of AQ_mod.

- DERX: the receiver gate is opened (default 3 μsec).

- DEADC: the digitizer is enabled (default 4 μsec).

- DEPA: the preamplifier is switched from transmit to observe mode (default 2 μsec).

Normally, the default values, which have been set during the installation of your spectrometer, can be used. Each subdelay has a maximum of DE - 1 μsec.

FILCOR is a correction to the acqt0 value which is used by the baseopt filter. It takes into account that the excitation signals as well as the FID need some time to propagate through the cables and the transmitter or receiver, respectively. This progpagation time creates an inevitable delay between the excitation and the receiving of the FID. In contrast to the value acqt0 which depends solely on the pulse program, FILCOR is a hardware correction value. Therefore, it is part of the edscon parameters and does not need to be changed by the user. Typical values are in the order of 1 μsec. or less.

In most pulse programs, data sampling is performed by the `go` statement, which automatically triggers the actions mentioned above after the corresponding pre-scan subdelay. If, however, data sampling is performed by the `adc` statement, these actions must explicitly be specified in the pulse program. Each action can be performed by a statement with the same name, in lower case letters, as the corresponding pre-scan subdelay. For example, the receiver gate can be opened with the `derx` statement. You can type **edpul zgadc** to look at an example of a pulse program using the `adc` statement. For more information on this topic click:

Help => **Manuals** => [**Programming Manuals**] **Pulse Programming Manual**

The prescan subdelays only play a role for digitally filtered data (DIGMOD = digital or digital homodecoupling). For DIGMOD = analog, the parameter DE has a different purpose. It is used to achieve a near zero first order phase correction of the spectrum. In this case, it does not consist of subdelays.

INPUT AND OUTPUT FILES

- <tshome>/conf/instr/<instrum>/
  - *scon* - spectrometer constants

SEE ALSO

cf [▶ 49]

3.7   edsolv

NAME

edsolv - Edit solvent table

DESCRIPTION

The command **edsolv** opens the solvent table (see the next figure).
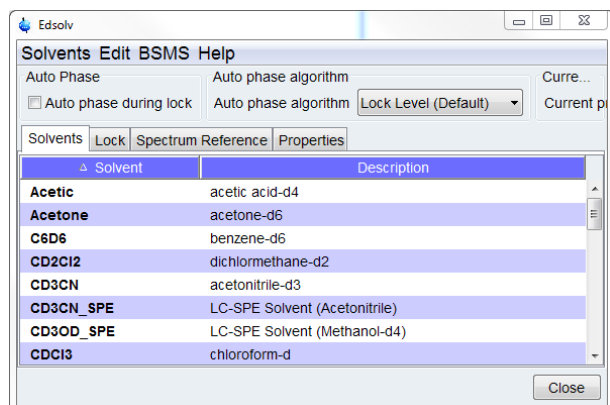
Figure 3.20: edsolv

This table contains one line for each solvent and shows the solvent name and a short description.

You can right-click in the table to copy or export the selected entries or to modify the table properties.

The buttons at the bottom of the solvent table have the following functions:

- **Add/Edit**

Edit a table entry or add a new one. A dialog is opened (see the next figure) where you can enter or change values. If you enter a new **Solvent**, an entry is added to the list, otherwise the current entry is edited.

• **Delete**

Delete the selected entry/entries.

• **Restore**

Restore the original solvent table. All changes that you made will be undone.

• **Save**

Save any changes the solvent table. You will be prompted for the NMR administration password.

• **Close**

Close the solvent table.
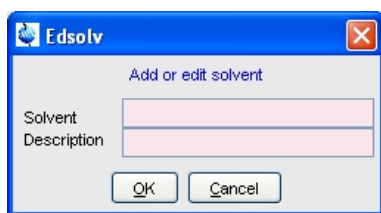
Figure 3.21: Add or edit solvent

Before you start an experiment, you must set the parameter SOLVENT to an entry from the solvent table. If you do this from **eda**, you can click the arrow button to the right of this parameter and select an entry from the solvent list.

INPUT FILES

• <tshome>/exp/stan/nmr/lists/

 – *solvents.all* - complete Bruker solvent list (input for first **edsolv** and for **edsolv =>
Restore**)

 – *solvents* - user solvent list (input for second or later **edsolv**)

OUTPUT FILES

• <tshome>/exp/stan/nmr/lists/

 – *solvents* - user solvent list

SEE ALSO

ednuc [▷ 67], *edhead [▷ 111]*, *edlock [▷ 91]*, *edprosol [▷ 69]*

3.8   edsp

NAME

edsp - Set up nuclei and spectrometer routing

DESCRIPTION

The command **edsp** allows you to set up the nuclei and the spectrometer routing for the current experiment. **edsp** is equivalent to **edasp** except for the following:

- **edasp** reads the irradiation frequencies offsets OFSH1, OFSX1, OFSX2 etc. from the current dataset.
- **edsp** reads the irradiation frequencies offsets OFSH1, OFSX1, OFSX2 etc. that were stored by the previous **edsp** or **edasp**.

As such, **edsp** allows you to transfer the frequency offset for a certain nucleus from one dataset to another. For example:

DATASET 1

   rpar PROTON all

   edasp or **edsp** =>

      BF1 500.130 MHz NUC1

      SFO1 500.135 Mhz F1

      OFSH1 **5000.00** Hz 1H

   SAVE

DATASET 2

   rpar PROTON all

   edsp =>

      BF1 125.757 MHz NUC1

      SFO1 125.758 Mhz F1

      OFSH1 1000.00 Hz 13C

      BF1 500.130 MHz NUC2

      SFO1 500.135 Mhz F2

      OFSH1 **5000.00** Hz 1H

The irradiation frequencies SFO1, SFO2 etc. are automatically adjusted to the corresponding frequency offsets.

The command **edsp** is also part of the spectrometer configuration. It must be executed only once, after the installation of TopSpin.

edsp should be executed before **expinstall** which installs the standard parameter sets.

INPUT AND OUTPUT PARAMETERS

   see **edasp**

INPUT AND OUTPUT FILES

   see **edasp**

SEE ALSO

3.9　　expinstall

NAME

   expinstall - Install pulse programs, AU programs, parameter sets etc.

DESCRIPTION

The command **expinstall** installs pulse programs, AU programs, parameter sets and various other resources for spectrometer usage. On a spectrometer, it must be performed once, after the installation of TopSpin and after **cf** has been done. On a datastation, **cf** is not needed and you can run **expinstall** immediately after the installation of TopSpin.

Configure a Spectrometer

To configure a spectrometer:

- Click **Spectrometer => Setup => Experiment installation** or enter **expinstall** on the command line.
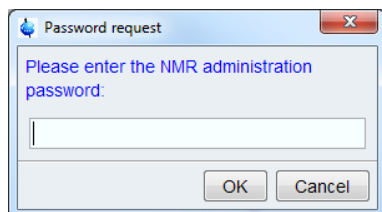
Figure 3.22: NMR admin password

- Enter NMR Administrator password as requested.
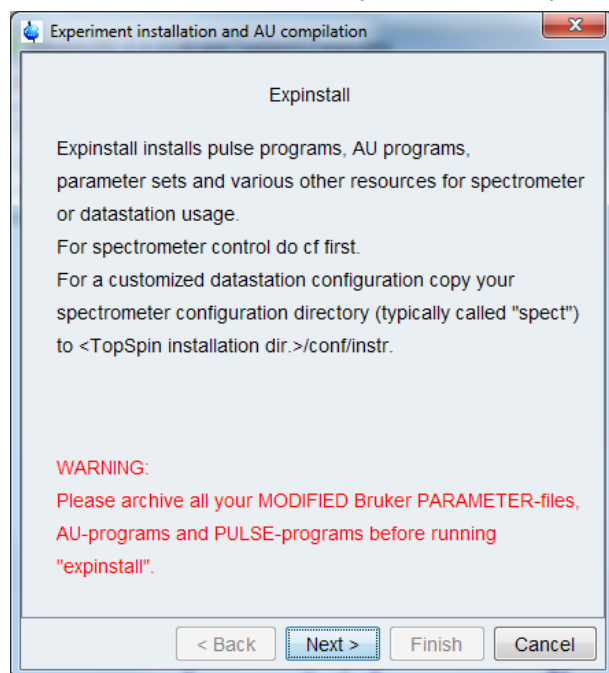
Figure 3.23: Expinstall

In the next information box:
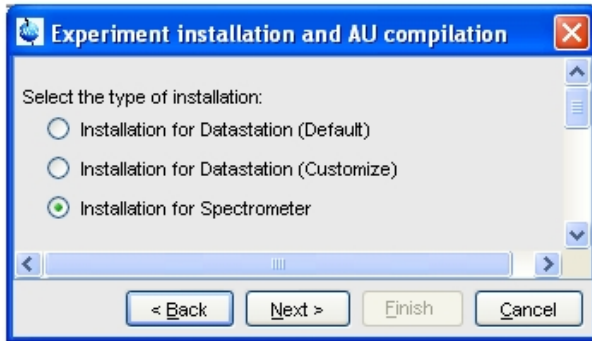
- Click **Next**

Figure 3.24: Experiment installation and AU compilation

In the next dialog box:

- Check **Installation for Spectrometer**.
- Click **Next**
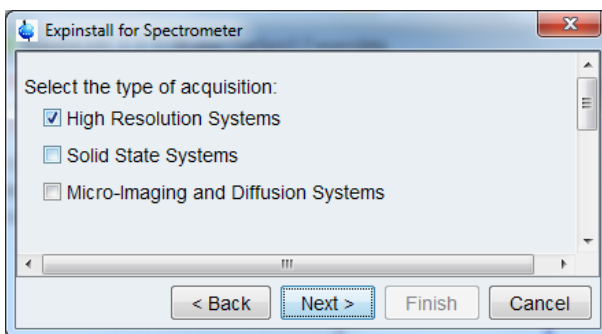
Figure 3.25: Expinstall - Select type of acquisition

In the next dialog box:

- Check **High Resolution Systems**.
- Click **Next**
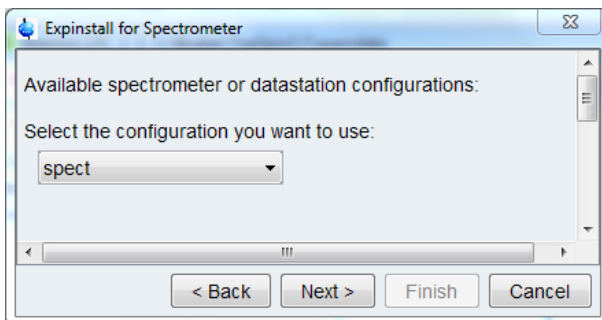
Figure 3.26: Expinstall - Available spectrometer or datastation config

In the next dialog box:

- Select the spectrometer configuration name that was defined with **cf**
- Click **Next**
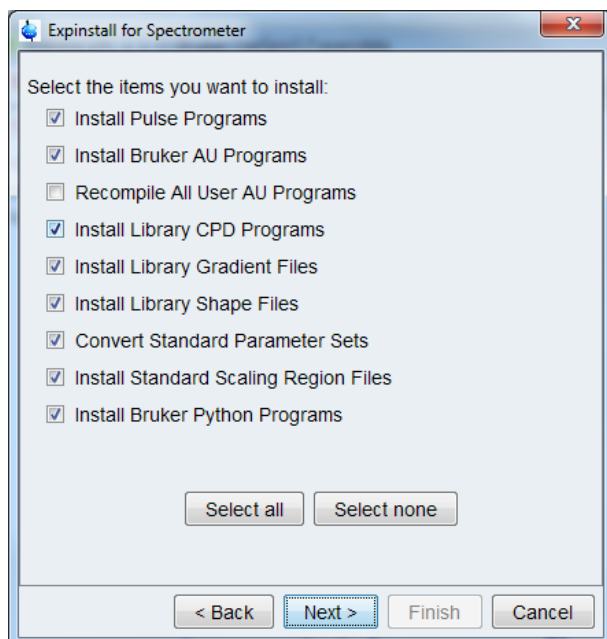
Figure 3.27: Expinstall - for spectrometers

In the next dialog box:

- Check the items you want to install, or accept the default selection.
- Click **Next**
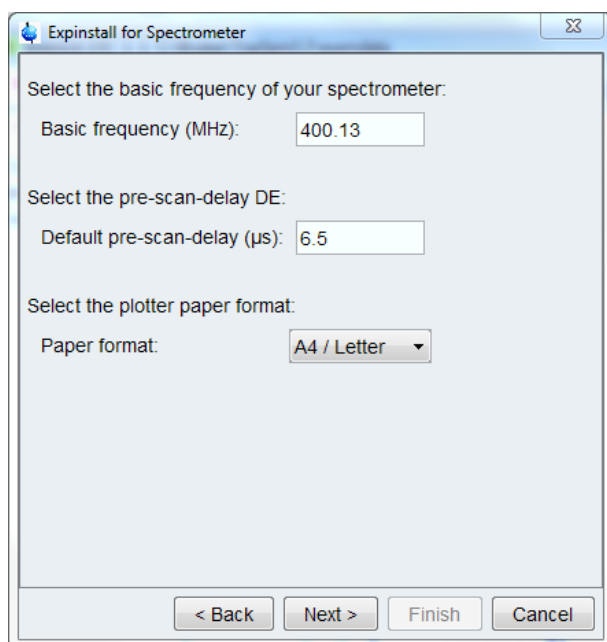
Figure 3.28: Expinstall - Select printer

In the next dialog box:

- Select the default printer and plotter and the desired paper format.
- Click **Next**
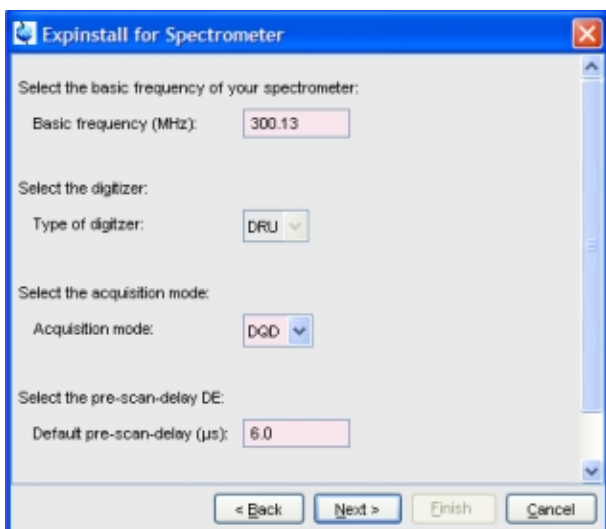
Figure 3.29: Expinstall - Basic frequency

In the next dialog box:

- Enter the desired spectrometer frequency, acquisition mode and pre-scan-delay, or accept the default values.
- Click **Next**.
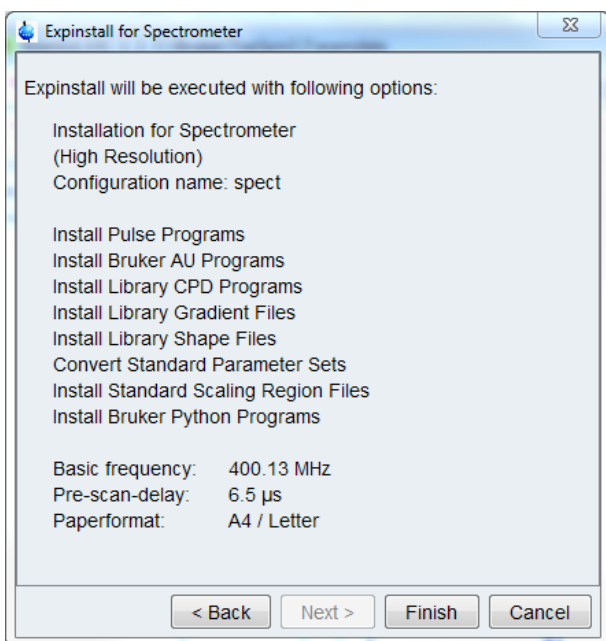
Figure 3.30: Expinstall - Summary

In the next information box:

- Check the configuration selection and, if it is correct, click **Finish**. If it is not correct, click **Back** to change the incorrect settings.

The installation of the selected items, will start now. Wait until this process has finished.

Configure a Datastation like a Spectrometer

If you want to configure your datastation like your spectrometer, you must first copy the configuration directory:

<tshome>/conf/instr/<instrum>

From that spectrometer to the data station. Here:

- <tshome> is TopSpin home, the directory where TopSpin is installed.

> **i**   Note that this can be different on the spectrometer than on the datastation.

- <instrum> is the configuration name.

See also the description of the command **nmr_save**.

After copying the configuration directory, you have to perform **expinstall** as follows:

- Click **Spectrometer => Setup => Experiment installation** or enter **expinstall** on the command line.
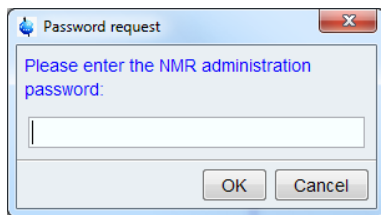
Figure 3.31: NMR admin password

- Enter NMR Administrator password as requested.
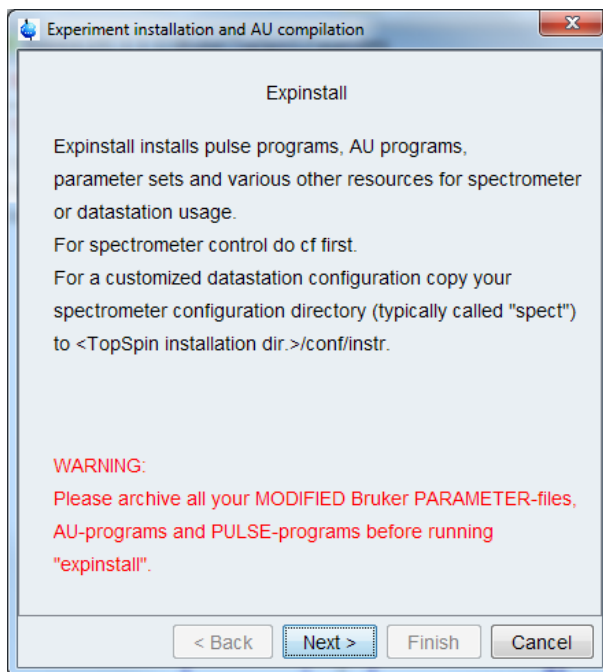
Figure 3.32: Expinstall

In the next information box:
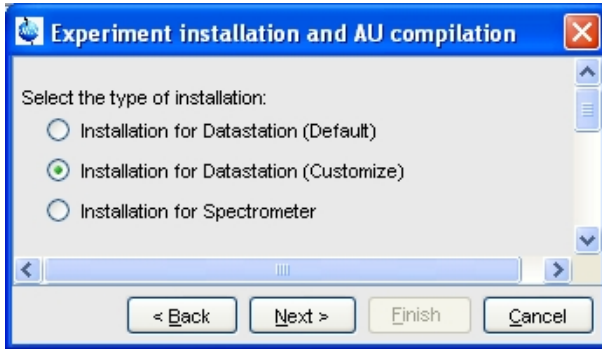
- Click **Next**.

Figure 3.33: Type of installation

In the next dialog box:

- Check **Installation for Datastation (Customize)**.
- Click **Next**.
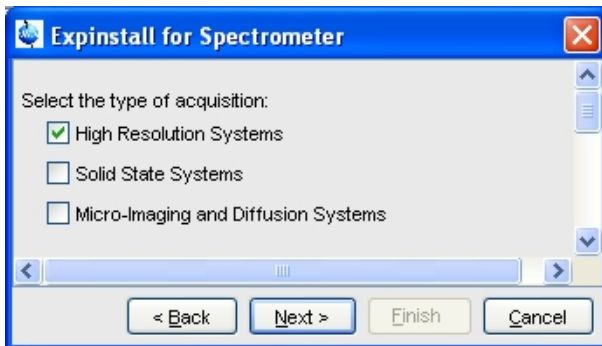
Figure 3.34: Type of acquisition

In the next dialog box:

- Check **High Resolution Systems**.
- Click **Next**.
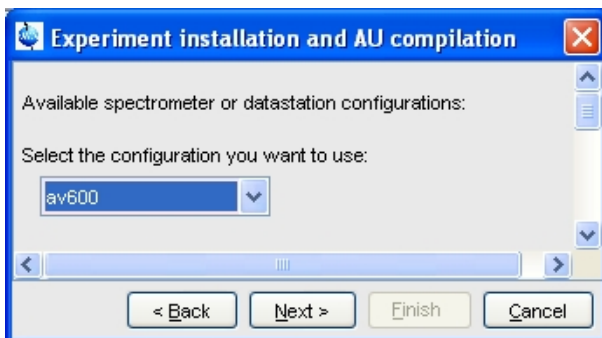
Figure 3.35: Available spectrometer or datastation configuration

In the next dialog box:

- Select the spectrometer configuration name.
- Click **Next**.
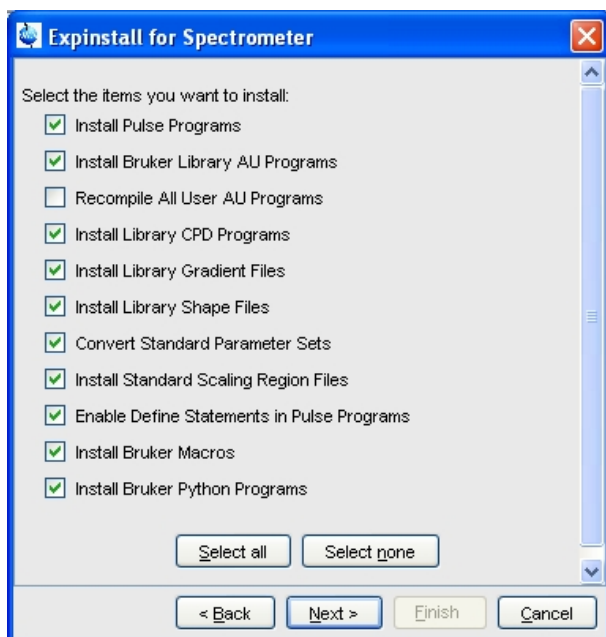
Figure 3.36: Select items to install

In the appearing dialog box:

- Check the items you want to install or, accept the default selection.
- Click **Next**.
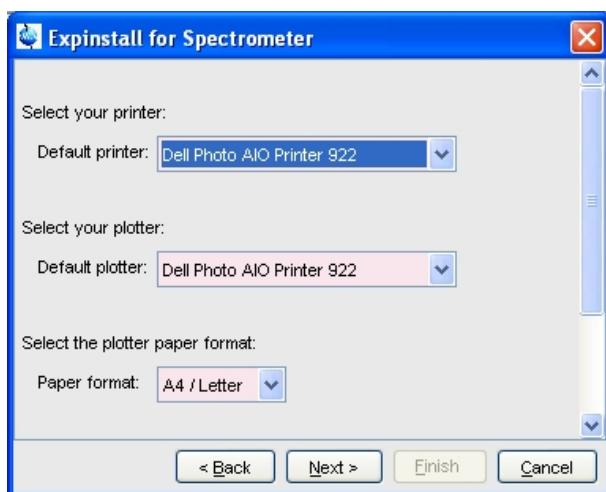
Figure 3.37: Select printer

In the next dialog box:

- Select the default printer and plotter and the desired paper format.
- Click **Next**.
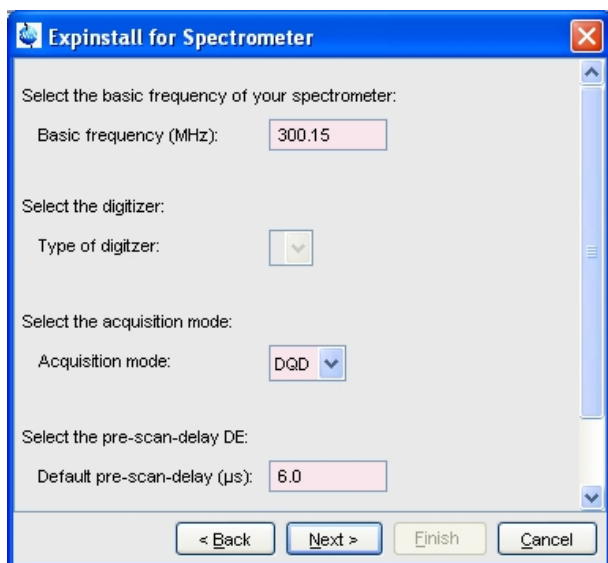
Figure 3.38: Select basic frequency

In the next dialog box:

- Enter the desired spectrometer frequency, acquisition mode and pre-scan-delay or, accept the default values.
- Click **Next**.

Figure 3.39: Expinstall - Option summary

In the next information box:

- Check the configuration selection and, if it is correct, click **Finish**. If it is not correct, click **Back** to change the incorrect settings.

The installation of the selected items, will start now. Wait until this process has finished.

INPUT PARAMETERS

If the task **Convert standard parameter sets** is selected, **expinstall** uses the following input parameters:

From the parameter sets as delivered with TopSpin:

- BF1 - BF4 - basic frequencies for channel f1 to f4.
- SFO1- SFO4 - irradiation (carrier) frequencies for channels f1 to f4.
- IN0 - increment for delay D0 (2D and 3D parameter sets only).
- IN10 - increment for delay D10 (3D parameter sets only).
- SW - spectral width in ppm.

- SPOFFS[0-7] - shaped pulse frequency offset.

OUTPUT PARAMETERS

If the task **Convert standard parameter sets** is selected, **expinstall** stores the following parameters in the parameter sets:

- BF1 - BF4 - basic frequencies for channel f1 to f4.
- SFO1- SFO4 - irradiation (carrier) frequencies for channels f1 to f4.
- SF - spectral reference frequency.
- IN0 - increment for delay D0 (2D and 3D parameter sets only).
- IN10 - increment for delay D10 (3D parameter sets only).
- SW - spectral width in ppm.
- SPOFFS[0-7] - shaped pulse frequency offset.
- DIGTYP - digitizer type.
- DR - digital resolution.
- DIGMOD - digitizer mode.
- DECIM - decimation factor of the digital filter.
- DE - prescan delay.
- FCUCHAN - routing between logical frequency channels and FCU's.
- RSEL - routing between FCU's and amplifiers .
- SWIBOX - routing between Switchbox inputs and Switchbox outputs.
- PRECHAN - routing between Switchbox outputs and Preamplifier modules.
- HPMOD - routing between high power amplifiers and Preamplifier modules.

INPUT FILES

- <tshome>/db/bruker/
    - *pp_dexam.xml* - Sources Bruker defined pulse programs
    - *cpd_dexam.xml* - Sources Bruker defined CPD programs
    - *mac.xml* - Sources Bruker defined macros
    - *pyexam.xml* - Sources Bruker defined Python programs
- <tshome>/prog/au/src.exam/* - Bruker AU programs (source files)
- <tshome>/conf/instr/<instrum>/specpar - routing parameters
- <tshome>/exp/stan/nmr/par.avance/* - Bruker parameter sets for Avance
- <tshome>/exp/stan/nmr/gp.dexam/* - gradient files for Avance
- <tshome>/exp/stan/nmr/wave.dexam/* - shape files for Avance
- <tshome>/exp/stan/nmr/scl.exam/* - scaling region files for Avance/AMX

These are the files/directories used for high resolution experiments. For other experiments other files/directories are used, e.g. *pp_dsolids.xml* and *gp.solids* for Solids pulse programs.

OUTPUT FILES

- <tshome>/exp/stan/nmr/au/src/* - Bruker AU programs (source files)
- <tshome>/exp/stan/nmr/par/* - parameter sets for your spectrometer
- <tshome>/exp/stan/nmr/lists/pp/* - pulse programs for your spectrometer
- <tshome>/exp/stan/nmr/lists/cpd/* - CPD programs for your spectrometer

- <tshome>/exp/stan/nmr/lists/gp/* - gradient files for your spectrometer
- <tshome>/exp/stan/nmr/lists/wave/* - shape files for your spectrometer
- <tshome>/exp/stan/nmr/lists/scl/* - scaling region files for your spectrometer
- <tshome>/exp/stan/nmr/lists/mac/* - TopSpin macros

SEE ALSO

cf [▷ 49], (cplbruk, cpluser, compileall, rpar, wpar)

3.10   ha

NAME

ha - Show hardware Ethernet addresses.

DESCRIPTION

The command **ha** opens the dialog window shown the addresses of the Ethernet accessible hardware devices. For each device, the Ethernet address is displayed and an **Open** button is available. Clicking this button opens the default web browser showing the respective hardware configuration. Here, you can set various configuration settings.

Figure 3.40: ha

The command **ha** can also be started from the TopSpin menu:

Click **Spectrometer => Setup => Ethernet addresses of hardware**

or

Click **Options => Spectrometer Tools => Ethernet addresses of hardware**

SEE ALSO

cf [▷ 49]

3.11   nmr_save, nmr_restore, user_save, user_restore

NAME

nmr_save - Save installation specific files

nmr_restore - Restore installation specific files

user_save - Save user specific files

user_restore - Restore user specific files

DESCRIPTION

The commands **nmr_save** and **user_save** save installation/user specific files in a backup TAR-file.

The commands **nmr_restore** and **user_restore** extract a backup TAR-file to the same or to a different installation.

All these commands open the dialog window shown in the next figure, the individual commands being available as tabs.
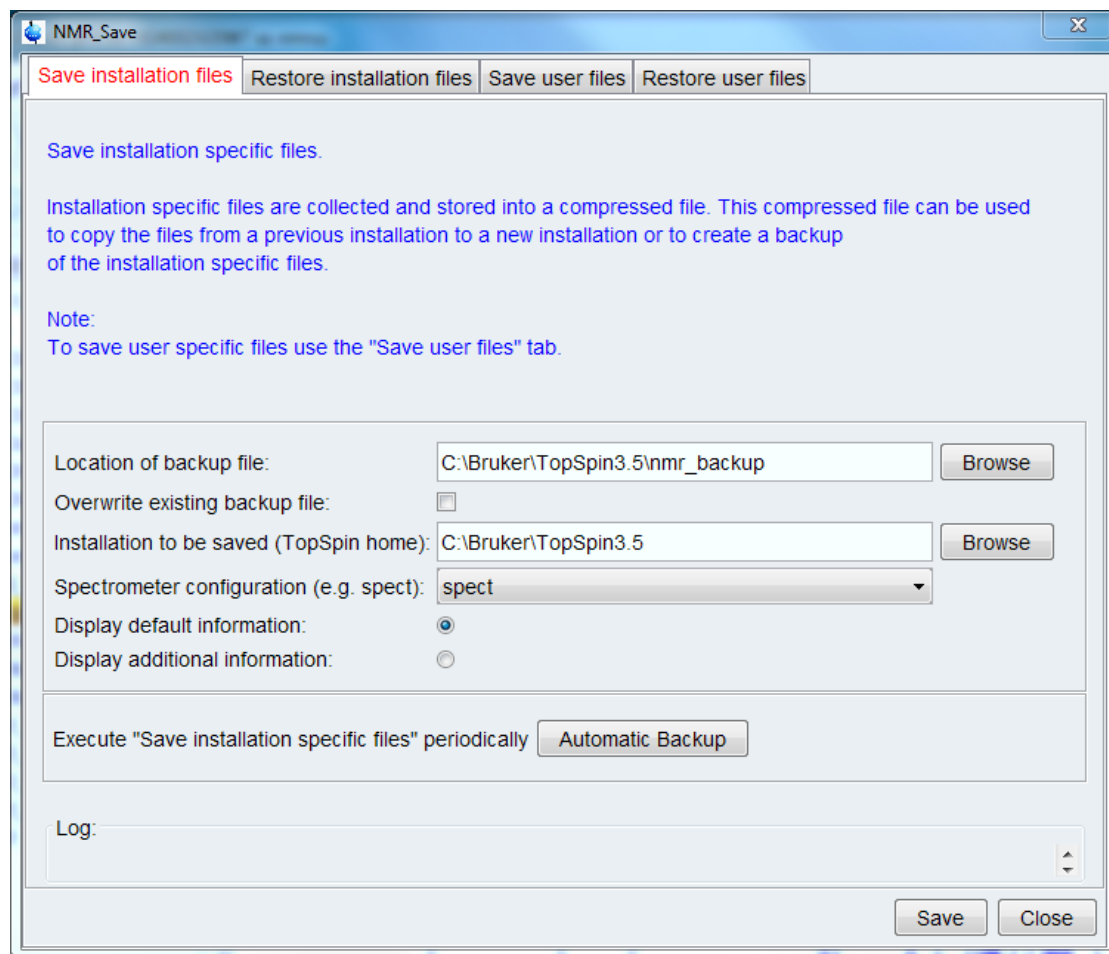
Figure 3.41: nmr_Save

Here you can specify:

- **Location of the backup file**: the storage directory of the backup file.
- **Installation to be saved** : The TopSpin home directory.
- **Spectrometer configuration**: as enter during **cf**.

Furthermore, you can select

- **Display default information**: the path of the created backup file.
- **Display additional information**: the path of the created backup file, information about directories being saved and converting/renaming information.

The button **Define cron job** opens the cron dialog where you can define a periodic save of Bruker or user files (see also command **cron**).

Here you can save or restore all TopSpin user defined files. This includes:

Spectrometer Configuration Commands

- Spectrometer configuration files (**cf**)
- Parameter sets (**rpar**, **wpar**)
- Pulse program (**edpul**, **edcpul**)
- AU programs (**edau**)
- Plot editor layouts (**plot**, **autoplot**)
- Shim files (**rsh**, **wsh**)
- Gradient shimming field maps (**gradshim**)
- IconNMR user information (**iconnmr**)
- Program Licenses (TopSpin, NMRSim, NMR Guide)
- Various lists like **scl**, **f1**, **ds** (**edlist**, **zg**, **gs**)
- TopSpin macros (**edmac**)
- Probe and solvent dependant parameters (**edprosol**)
- Lock parameters (**edlock**)
- Probe information (**edhead**)
- Nucleus information (**ednuc**)
- RF Shapes and gradients

etc.

Furthermore the files *prog/logfiles/heliumlog*,

prog/logfiles/heliumloig.err and *Bruker/licenses/license.dat* will be saved with the ending *.backup*.

> **i** Please note that these files will be stored in the folder **conf/instr/**, no longer in the original folder.

Furthermore beginning with TopSpin 3.0 the whole directory *<diskless>/crco_data/cryotool_log/* will be saved.

For more details about the commands nmr_save, nmr_restore, user-save and user-restore please refer to the respective Bruker TopSpin Installation Guides for Windows XP, Windows Vista or Linux.

INPUT AND OUTPUT FILES

- <tshome> /nmr_backup
  - *nmr_backup_<date>-<time>.tar* (**nmr_save**)
  - *nmr_backup_<username>-<date>-<time>.tar* (**user_save**)
- <tshome>/exp/stan/nmr/au/src/
  - *nmr_save* - AU program executed by **nmr_save**

INPUT AND OUTPUT DIRECTORIES

nmr_save stores various subdirectories/files of:

- *<tshome>/exp/stan/nmr/py/*
- *<tshome>/exp/stan/nmr/lists*
- *<tshome>/exp/stan/nmr/par/*
- *<tshome>/exp/stan/nmr/au/src/*

- *<tshome>/plot/layouts/*
- *<tshome>/conf/instr/*
- *<tshome>/prog/tcl/xwish3_scripts/*
- *<tshome>/exp/stan/nmr/parx/preemp/*
- *<tshome>/QTP/*
- *<tshome>/data/final/nmr/protocolfiles/*
- *<tshome>/conf/global*
- *<tshome>/prog/server/export.conf*

user_save stores various subdirectories/files of:

- *<userhome>/.topspin-<hostname>*
- *<tshome>/exp/stan/nmr/py/*
- *<tshome>/exp/stan/nmr/lists*
- *<tshome>/exp/stan/nmr/par/*
- *<tshome>/exp/stan/nmr/au/src/*

INPUT AND OUTPUT DIRECTORIES

SEE ALSO

cf [▷ 49], *expinstall [▷ 76]*, (cron)

3.12 paracon

NAME

paracon - Change the basic frequency in parameter sets

DESCRIPTION

The command **paracon** changes the basic frequency in parameter sets. This allows you to use parameter sets that were created on a spectrometer with a different frequency. It opens dialog box shown in the next figure.
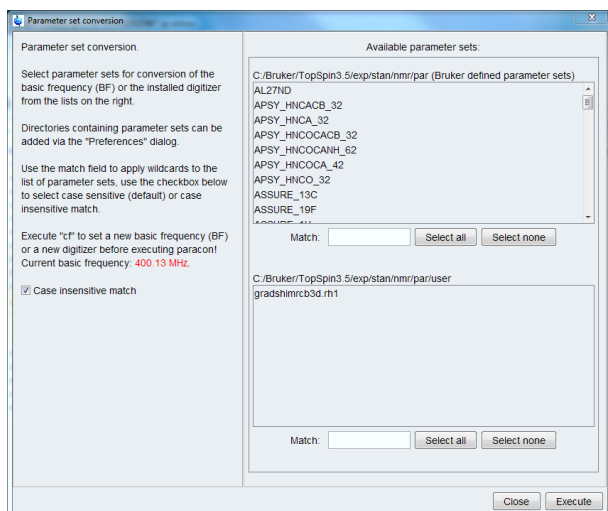
Figure 3.42: paracon

Here you can setup a list of available parameter sets. You can select Bruker and/or User defined parameter sets and uses a match string. The matching parameter sets appear in the right part of the dialog box. To start the conversion, select one or more parameter sets and click **OK**.

INPUT AND OUTPUT PARAMETERS

Acquisition parameters:

- BF1 - BF8 - basic frequency for frequency channel f1 to f8.
- SFO1 - SFO8 - irradiation (carrier) frequencies for channels f1 to f8.
- O1 - O8 - irradiation frequency offset for frequency channel f1 - f8 in Hz.
- SW - spectral width in ppm.

Processing parameters:

- SF - spectral reference frequency.

INPUT AND OUTPUT FILES

- <tshome>/exp/stan/nmr/par/*/
    - acqu - acquisition parameters
    - proc - processing parameters

SEE ALSO

expinstall [▶ 76]

4 Lock Commands

This chapter describes the commands that are involved in locking the magnetic field. Avance spectrometers use a digital lock which is provided by the BSMS unit. Locking the magnetic field is done by measuring the signal of a lock nucleus and adjusting the magnetic field whenever its frequency shifts. Common lock nuclei are 2H and 19F.

4.1 edlock

NAME

edlock - Edit the lock table

DESCRIPTION

The command **edlock** opens the lock table; a dialog box in which the lock parameters can be set (see the next figure).
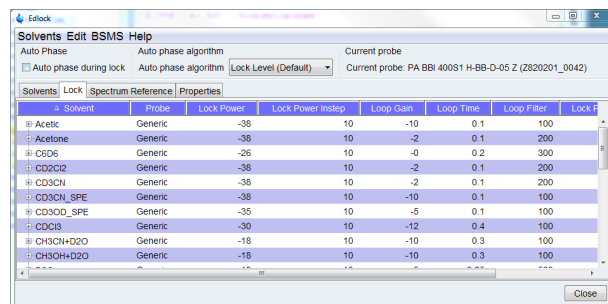
Figure 4.1: edlock

It must be executed once for each probe and each lock nucleus. The lock parameters can be determined from the BSMS keyboard or from the BSMS display (command **bsmsdisp**). The toolbar of the **edlock** dialog provides the following buttons:

Save the lock table.

Add a new solvent.

Delete the selected solvent(s).

Read the current values from the BSMS.

Copy the value of the selected parameter to all solvents.

Show the **edlock** help page.

Below the toolbar, the current probe, the field value (H0) and the basic spectrometer frequency are displayed. The latter must be specified when the spectrometer is configured with **cf**. The lock frequency is automatically calculated from the basic spectrometer frequency. You can select the lock nucleus and the observe nucleus.

Before you run **edlock**, you must define the current probe with the command **edhead**. Furthermore, you must define the lock nucleus by setting the parameter LOCNUC. This can be done with **eda** or by entering **locnuc** on the command line. In most experiments, the lock nucleus is deuterium. As such, LOCNUC is set to 2H in most Bruker standard parameter sets (see **rpar**).

When **edlock** is executed for the first time on a certain probe, a default lock table, which is delivered with TopSpin, is opened.

The table shows the lock nucleus and the current probe. The second line shows the lock frequency, the field value (H0) and the basic spectrometer frequency. The main part of the lock table shows a list of solvents and, for each solvent, the lock parameters.

Most lock parameters are used for locking the magnetic field during the acquisition. Others, however, are used for referencing the spectrum after the acquisition has finished.

Lock Parameters Used To Lock The Magnetic Field:

- Field-Correction - Difference between the Field value and the reference Field value. Typically used for **lock -noauto** on solvents with two or more lock signals (see the description of **lock**).
- LockPower - Lock power; power used to irradiate the lock nucleus (-60 to 0 dB)
- LoopGain - Loop gain; lock regulator gain (-80 to 0 dB).
- LoopTime - Loop time; lock regulator time constant (0.001 - 1.0 seconds).
- LoopFilt - Loop filter, lock regulator cut-off frequency of the lowpass filter (1 - 200 Hz).
- LockPhase - Lock phase; the phase of the lock signal.
- Distance - Chemical shift of the lock nucleus (irradiation frequency offset).

These parameters are interpreted by the commands **lock** and **lopo**.

Lock Parameters Used To Reference The Spectrum:

- Ref. - chemical shift of the reference signal (default 0).
- Width - width of the region where the reference signal is searched.
- Ref-Shift - reference shift for default calibration.

These parameters are interpreted by the command **sref**.

The loop gain, loop time and loop filter can also be set with the TopSpin commands **lgain**, **ltime** and **lfilter**, respectively. Furthermore, they can also be set from the BSMS keyboard menu. This menu can be entered by simultaneously pressing the two keys indicated with **MENU**.

The lock phase can be adjusted from the BSMS keyboard while observing the lock sweep on the lock display (**lockdisp**).

Note the difference between loop gain which can be set in **edlock** or with **lgain** and lock gain which can be set on the BSMS keyboard. The AU program **loopadj** automatically optimizes the lock phase, lock gain, loop gain, loop filter and loop time.

Note that **loopadj** optimizes these parameters for best long-term stability, but not for best lineshape, resolution or homogeneity (for more information type **edau loopadj** and look at the header of the AU program).

The Lock Power should be about 6 dB below the value where the lock signal starts to saturate. Saturation can be observed by increasing the lock power while observing the lock signal. When the signal no longer increases or starts to decrease, it is saturated.

In IconNMR automation, the lock parameters are read from the lock table and used as they are.

For information on how to determine the lock parameters click **Help => Manuals =>** [**Acquisition User Guides**] **Basic 1D and 2D Experiments**. Furthermore, you can refer to the spectrometer hardware documentation which is available on the BASH CDROM.

INPUT AND OUTPUT PARAMETERS

See DESCRIPTION above

INPUT FILES

- <tshome>/exp/stan/nmr/lists/
  - *2Hlock* - 2H lock parameters (input for first **edlock** on current probe)
  - *19Flock* - 19F lock parameters (input for first **edlock** on current probe)

INPUT AND OUTPUT FILES

- <tshome>/conf/instr/<instrum>/
  - *2Hlock* - 2H lock parameters
  - *19Flock* - 19F lock parameters
- <tshome>/conf/instr/<instrum>/prosol/<probeID>/<solvent>/
  - *bsmspar* - solvent and probe dependent lock parameters

SEE ALSO

4.2 lgain, ltime, lfilter

NAME

lgain - Set the lock regulator loop gain.

ltime - Set the lock regulator loop time.

lfilter - Set the lock regulator loop filter.

SYNTAX

lgain [<gain>]

ltime [<time>]

lfilter [<filter>]

DESCRIPTION

The command **lgain** allows you to set the loop gain, a lock regulator parameter. It takes one argument; a loop gain value between -80 and 0 dB. This value is only used when the lock-in process is done from the BSMS keyboard; i.e. by pressing **Lock On/Off** or **Autolock**. When lock-in is done with the TopSpin command **lock**, the loop gain is set to the **edlock** parameter **LGain**.

ltime and **lfilter** work like **lgain**, except that they set the regulator parameters loop time and loop filter, respectively.
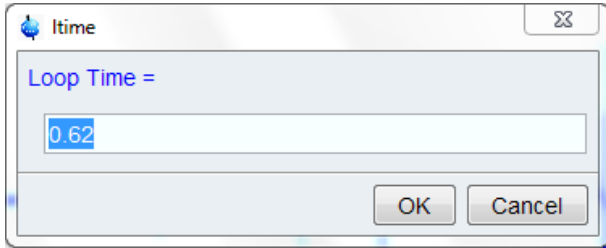
Figure 4.2: ltime
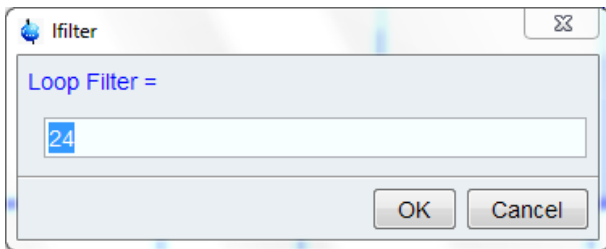
Figure 4.3: lfilter
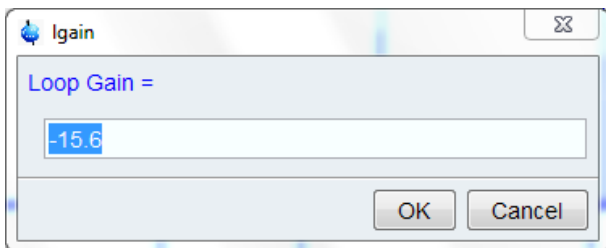
Figure 4.4: lgain

The AU program **loopadj** automatically optimizes lock gain, lock phase, loop time, loop gain and loop filter.

The regulator (loop) parameters can also be set from the BSMS keyboard menu. This can be opened by simultaneously pressing the two keys indicated with **MENU**.

For information on how to determine the lock parameters click **Help => Manuals =>** [**Acquisition User Guides**] **Basic 1D and 2D Experiments**.

Furthermore, you can refer to the spectrometer hardware documentation which is available on the BASH CDROM.

USAGE IN AU PROGAMS

> LTIME(value)
>
> LGAIN(value)
>
> LFILTER(value)

SEE ALSO

> *lock [▷ 94]*, *lopo [▷ 99]*, *edlock [▷ 91]*

4.3   lock

NAME

> **lock** - Lock the magnetic field

SYNTAX

lock [<solvent> | -acqu | -noauto]

DESCRIPTION

The command **lock** performs the lock-in procedure. It takes one argument and can be used in the following ways:

- **lock**

Opens a dialog box with a list of solvents (see the next figure).
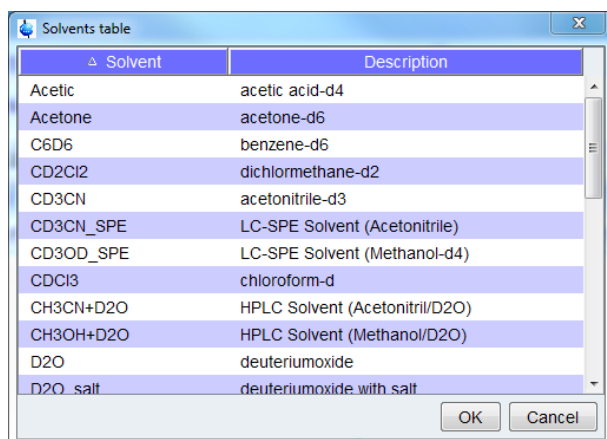
Figure 4.5: lock

When you select a solvent and click **OK**, it reads the lock parameters for that solvent and performs an **autolock** accordingly.

Right-click in the table to copy or export the selected entry or to modify the table properties.

> **i** Note that **lock** only shows the Solvents for the current lock nucleus (parameter LOCNUC), in this case 2H.

- **lock <solvent>**

Reads the lock parameters for the specified solvent and performs an **autolock** accordingly.

- **lock –acqu**

Reads the lock parameters for the solvent defined by the acquisition parameter SOLVENT and performs a **lock** or **autolock** depending on the IconNMR configuration) See Configuration Locking/Shimming options):

– If the flag **lock noauto** is set, **lock** is performed.

– If the flag **lock noauto** is not set, **autolock** is performed.

– If the flag **lock fieldupdate** is set, the BSMSfield value is updated (If it has not been updated during the last 24 hours.) (see also **edlock**).

> **i** Note that IconNMR executes **lock -acqu**.

- **lock –noauto**

Reads the lock parameters for the solvent defined by the acquisition parameter SOLVENT and performs a **lock**.

Lock Commands

Lock reads the lock parameters from the lock table which has been set up with the command **edlock**.

Lock parameters	Description	BSMS keyboard
Field	field (H0)	FIELD
LockPower	lock power	LOCK POWER
LockPhase	lock phase	LOCK PHASE
-	lock gain	LOCK GAIN
LoopGain	loop gain	Menu [1]
Looptime	loop time	Menu [1]
LoopFilt	loop filter	Menu [1]
Distance	lock frequency shift	LOCK SHIFT

Table 4.1: Lock Parameters

[1]Can be modified by simultaneously pressing the two buttons indicated with **MENU**, turn the knob until Lock appear, press the button 2nd and turn the knob again.

The **autolock** procedure involves the following steps:

- Irradiation of the lock nucleus with frequency **Lock Freq** + **Distance.**
- Acquisition of the lock nucleus FID.
- Fourier transform and magnitude calculation of the acquired FID.
- Determination of the position of the lock signal in the spectrum.
- Adjusting the Field such that the lock signal is exactly on resonance.
- Optimization the lock power and lock gain.

ℹ️ Note that the lock irradiation frequency (**Lockfreq** + **Distance**) is solvent dependent. The value of **Distance** is the chemical shift of the lock nucleus in the current solvent. As such, the irradiation frequency is approximately on resonance and **lock** needs to make only minimum field adjustments (capture range 1000 units). The advantage of this procedure is that the signal of the reference substance (e.g. TMS) appears at approximately the same position for each solvent.

The command **lock -noauto** simply reads the lock parameters and performs a lock-in accordingly. It does not adjust the field (capture range 10 units) and is typically used for samples with multiple lock signals (see below).

Lock Procedure For Solvents With Multiple Lock Signals

The Autolock procedure is suitable for solvents with a single lock signal. For solvents, however, with multiple lock signals, e.g. mixed solvents, Autolock is not recommended because it simply locks on the strongest signal.

For solvents with multiple lock signals, the following procedure can be used:

- Perform Autolock as described above for a reference solvent. For example, for CDCl3, enter **lock CDCl3**. This will set the Field for this solvent on the BSMS. Lets assume this Field value is 5000.
- Enter **edlock** and, in the lock parameter table, click 〰 to update the Field for each solvent.
- Insert the sample containing the solvent with multiple lock signals, e.g. with MeOD.
- Enter **lopo <solvent>**, e.g. **lopo MeOD** to set the lock parameters on the BSMS.

- Adjust the Field on the BSMS keyboard to put the desired lock signal exactly on resonance. Lets assume the Field value is 5100.

- Press the BSMS **Lock On/Off** key to lock-in.

- Enter **edlock** and, in the lock parameter table, enter the difference between the two Field values, in our case 100, for the used solvent in the column **Field correction**. Save the lock parameter table.

ⓘ   Note that **lock** reads the regulator parameters loop gain, loop time and loop filter from the lock table. However, these are only used after lock-in has been performed, i.e. after the **lock** command has finished. During the lock-in process fixed regulator values are used.

The lock signal can be viewed in the lock display window which can be opened with the command **lockdisp** (see the description of this command).

The lock-in procedure can also be performed from the BSMS keyboard by pressing the **Autolock** or **Lock On/Off** key. In that case the lock parameter values that are currently stored on the BSMS unit are used. These can be modified from the BSMS keyboard as indicated in table 4.1 or read from the lock table with the command **lopo**. As such,

lopo => Autolock is equivalent to **lock**

lopo => Lock On/Off is equivalent to **lock -noauto**

For information on how to determine the lock parameters click **Help => Manuals =>** [**Acquisition User Guides**] **Basic 1D and 2D Experiments**. Furthermore, you can refer to the spectrometer hardware documentation which is available on the BASH CDROM.

INPUT PARAMETERS

- Set by the user with **eda** or by typing **solvent** etc.:
    - SOLVENT - sample solvent (input for **lock -acqu** and **lock -auto**)
    - LOCNUC - lock nucleus
- Set by the user with **edlock**:
    - See also the table *Table 4.1 [▷ 96]*.

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
    - *acqu* - acquisition parameters
- <tshome>/conf/instr/
    - *probe* - current probe as defined with **edhead**
- <tshome>/conf/instr/<instrum>/
    - 2Hlock - lock table for nucleus 2H
    - 19Flock - lock table for nucleus 19F
- <tshome>/conf/instr/<instrum>/prosol/<probeID>/<solvent>/
    - *bsmspar* - solvent and probe dependent lock parameters
- <tshome>/exp/stan/nmr/lists/
    - *solvents.fieldupdate* - list of solvents for which **lock -acqu** performs an automatic Field update (set up from IconNMR configuration)
    - *solvents.noauto* - list of solvents for which **lock -acqu** performs **lock -noauto** (set up from IconNMR configuration)

USAGE IN AU PROGRAMS

- LOCK

  Executes the command **lock -acqu**.

SEE ALSO

edlock [▷ 91], *lopo [▷ 99]*, *lockdisp [▷ 98]*, *lgain, ltime, lfilter [▷ 93]*

4.4   lockdisp

NAME

lockdisp - Open the lock display window.

DESCRIPTION

The command **lockdisp** opens the lock display window in which the lock signal is shown (see the next figure).
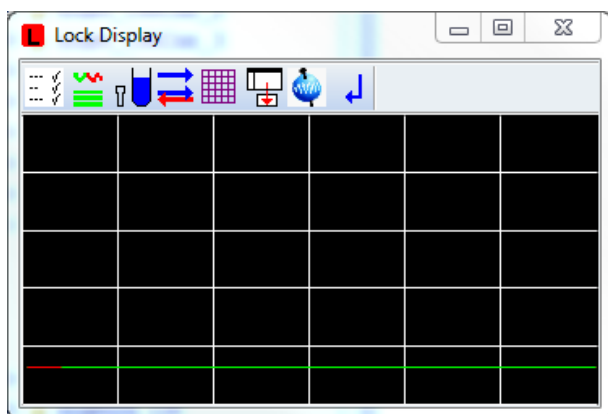
Figure 4.6: Lock Display

At the top of the lock window the following buttons are available:

Open the User Preferences window [**set**].

Toggle lock monitor mode.

Toggle lock display mode between **single** and **dual** color.

Switch grid mode between **both**, **vertical**, **horizontal** and **off.**

Make the lock display external.

Put focus into TopSpin window.

Close the lock display window.

The lock signal is the NMR signal of the lock nucleus which (usually 2H).

The lock display can be used in two different ways:

- When the field is unlocked.

  In this case the lock frequency is swept and lock signal shows a pattern of wiggles with a maximum intensity at the resonance frequency. This pattern can be used to set the field value and lock phase. The field value can be adjusted from the BSMS keyboard until the maximum lock signal lies at the center of the lock display window. The lock phase can be adjusted until the intensity is about the same on both sides of the center.

- When the field is locked.

  In this case the lock frequency is kept at the resonance position of the lock nucleus and is only adjusted in responds to changes in the magnetic field. The lock signal is a horizontal line in the upper part of the lock display window. Furthermore, the position of this line can be used for shimming the magnet. When the homogeneity of the field improves, the lock signal increases, i.e. the line moves upward.

SEE ALSO

lock [▸ 94], *edlock [▸ 91]*

4.5 lopo

NAME

lopo - Set the lock parameters.

DESCRIPTION

The command **lopo** shows a list of available solvents (see the next figure).
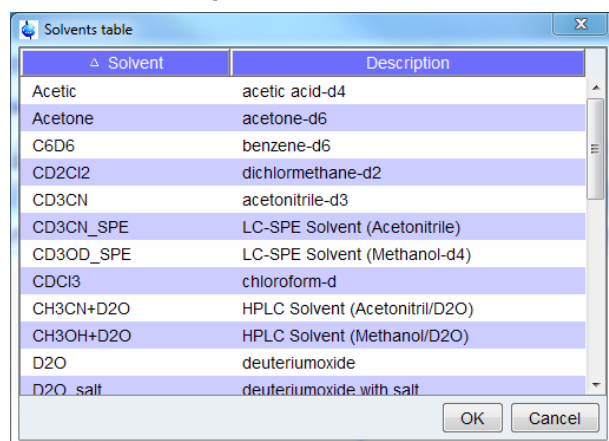
Figure 4.7: lopo

When you select a solvent and click **OK**, it sets the lock parameters according to the **edlock** table. As such, the lock power, loop gain, loop time, loop filter, lock phase and frequency shift are set to the lock table values of Lockpower, LoopGain, LoopTime, LoopFilt, LockPhase and Distance, respectively. These values are set on the BSMS unit without performing lock-in.

Right-click in the table to copy or export the selected entry or to modify the table properties.

The command **lopo** is useful if you want to observe the lock signal first. The lock-in procedure can then be performed by pressing the **Lock On/Off** or **Autolock** key on the BSMS keyboard.

---

i | Note that:

lopo => **Autolock** is equivalent to **lock -acqu**

lopo => **Lock On/Off** is equivalent to **lock -noauto**

INPUT PARAMETERS

- Set by the user with **eda** or by typing **solvent** etc.:
  - SOLVENT - sample solvent

- – LOCNUC - lock nucleus
- Set by the user with **edlock** :
  - – See also the table *Table 4.1 [▷ 96]*

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - – *acqu* - acquisition parameters
- <tshome>/conf/instr/
  - – probe - current probe as defined by **edhead**
- <tshome>/conf/instr/<instrum>/
  - – 2Hlock - lock table for nucleus 2H
  - – 19Flock - lock table for nucleus 19F

<tshome>/conf/instr/<instrum>/prosol/<probeID>/<solvent>/

bsmspar - solvent and probe dependent lock parameters

USAGE IN AU PROGRAMS

- LOPO

SEE ALSO

lock [▷ 94], *edlock [▷ 91]*, *lockdisp [▷ 98]*, *lgain, ltime, lfilter [▷ 93]*

5 Shim Commands

This chapter describes the commands which are involved in shimming the magnet. Shimming can be started with reading an appropriate shim set for the current sample and then optimizing the shim from the BSMS keyboard. In automation, the optimization is performed by an automatic shimming procedure. Furthermore, if you are using a gradient probe, you can perform gradient shimming.

5.1 autoshim

NAME

autoshim - Autoshim switching

DESCRIPTION

The command **autoshim** controls the autoshim function of the BSMS unit. It takes one argument and can be used as follows:

autoshim on

turn autoshimming on

autoshim off

turn autoshimming off

Entering **autoshim** on the command line has the same effect as pressing the AUTOSHIM key on the BSMS keyboard. The shims for which the step size is set to a value greater than 0 are optimized. It is typically used in automation.

USAGE IN AU PROGRAMS

- AUTOSHIM_ON
- AUTOSHIM_OFF

SEE ALSO

tune [▷ 104], *edtune [▷ 101]*, *rsh, wsh, vish [▷ 103]*, *delpar, delgp, delsh [▷ 190]*

5.2 edtune

NAME

edtune - Edit the tune file

SYNTAX

edtune [<name>]

DESCRIPTION

The command **edtune** opens a dialog box where you can edit an existing tune file or create a new one (see the next figure). The tune files you see listed here are delivered with TopSpin.
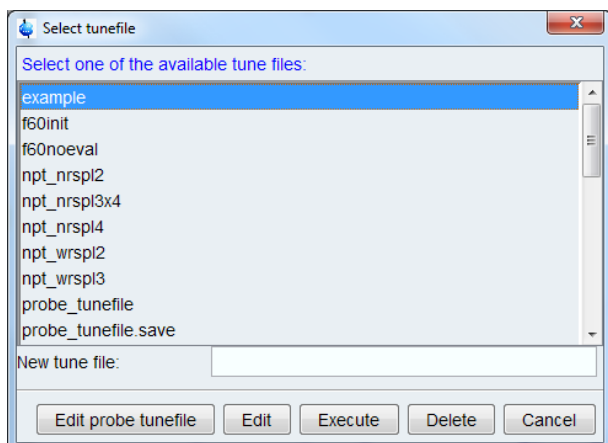
Figure 5.1: edtune

The following buttons are available:

- **Edit probe tunefile**
  - Edit the probe specific tune file as it was created with **edprosol**.
- **Edit**
  - Edit the selected tune file. The file will be opened in an editable window which offers a **Save** and a **Save as** button. The latter one allows you to save the tune file under a different name, i.e. create a new tune file.
- **Execute**
  - Execute the selected tune file. This will run the command **tune**.
- **Delete**
  - Delete the selected tune file.
- **Cancel**
  - Quit the **edtune** dialog box.

Tune files are used by the **tune** command which performs automatic shimming.

- **edtune** can take one argument and can be used as follows:
  - **edtune <name>**

    Edits the specified tune file. If the specified file does not exist, an error message will appear.

    The format of a tune file is described for the **tune** command.

INPUT AND OUTPUT FILES

- <tshome>/exp/stan/nmr/lists/group/
  - *example* - standard tune file
  - *my_tunefile* - user defined tune file
- <tshome>/conf/instr/<instrum>/prosol/<probeID>/
  - *tunefile* - probe dependent tune file (input for **edtune** => **Probe tunefile**)

SEE ALSO

tune [▷ 104], rsh, wsh, vish [▷ 103], delpar, delgp, delsh [▷ 190], gradshim

5.3   rsh, wsh, vish

NAME

rsh - Read a set of shim values

wsh - Write a set of shim values

vish - View a shim file

SYNTAX
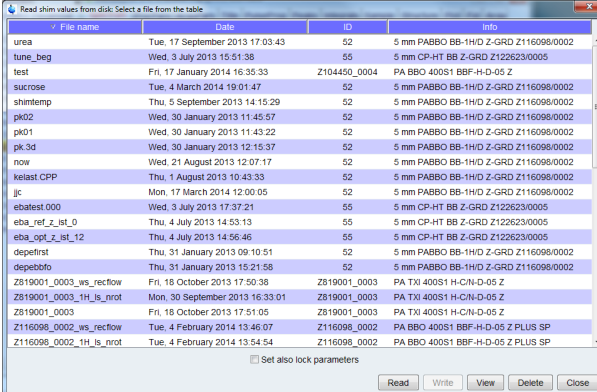
rsh [<name>]

wsh [<name>]

vish [<name>]

DESCRIPTION

The ***sh** all open the shim dialog box when they are entered without arguments (see the next figure).

Figure 5.2: Read shim values

Here, you can enter the shim file and then do one of the following actions.

The following buttons are available:

- **Read**

  Executes the command **rsh**, reading the specified shim file. When you click on a shim set, its values are loaded to the shim unit. **rsh** can also be entered on the command line with a shim file as an argument. After reading a shim file, is usually necessary to optimize the shims, especially the Z and Z^2 shim. You can do that from the BSMS keyboard or from the BSMS display (command **bsmsdisp**). **rsh** switches the autoshim function of the BSMS unit off. If you press the **AUTOSHIM** key on the BSMS keyboard or enter **autoshim on**, the shims will be continuously optimized during the experiment.

- **Write**

  Executes the command **wsh** writing the shim values which are currently installed on the shim unit to the specified shim file. **wsh** can also be entered on the command line with a shim file as an argument.

Figure 5.3: Write Shim Values

- **View**

  Executes the command **vish** opening the specified shim file with an editor. **vish** can also be entered on the command line with a shim file as an argument.

- **Delete**

  Deletes the selected shim file. If he ***sh** commands are entered the command line with an argument, then they may contain wildcards; for example:

  rsh a* lists all shim files beginning with "a"

> Note that **rsh**, **wsh** and **vish** open the dialog box with the **Read**, **Write** and **View** button activated, respectively.

INPUT FILES

- <tshome>/exp/stan/nmr/lists/bsms/
  - shim files

USAGE IN AU PROGRAMS

- RSH(name)
- WSH(name)

SEE ALSO

edtune [▶ 101], *tune [▶ 104]*, (gradshim)

5.4 tune

NAME

tune -Perform automatic shimming of the magnet

SYNTAX

tune [<tunefile> | .sx]

DESCRIPTION

The command **tune** shims the magnet in an automatic procedure according to a shim definition file. This file is called the tune file and can be set up with the command **edtune**. The command **tune** takes one argument and can be used in one of the following ways:

tune

Displays a list of available tune files (see the next figure).
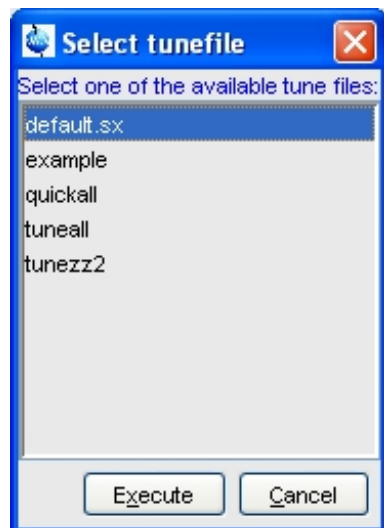
Figure 5.4: Select one of the available tune files

When you select an entry and click the **Execute** button, the corresponding tune file is interpreted and auto shimming is performed accordingly.

tune <tunefile>

Perform auto shimming according to the specified tune file.

tune .sx

Perform auto shimming according to the tune file as it is specified for the current probe. This tune file can be setup from the **edprosol** dialog box by clicking **File => Edit Tunefile**.

An example tune file is delivered with TopSpin. You can use this as it is or modify it to your needs and store it under a different name. The statements you can use in a tune file are listed below.

> **i** Note that some of these statements are settings whereas others are commands.

Settings in a tune file:

- USE_FIDAREA
  - Flag indicating to use the area under the FID envelope as a criterion for field homogeneity.
- USE_LOCKLEVEL
  - Flag indicating to use the lock level as a criterion for field homogeneity.
- LOCKDWELL n
  - The number of measurements used for determining the current lock level. The measured values are averaged to suppress the effects of the noise on the lock level. Only used when USE_LOCKLEVEL is defined.
- MAXLOCK m

– The maximum lock level. Can be used to keep the lock signal from moving out of the display during the shimming procedure. The lock level is displayed at the TopSpin status line while **tune** is running.

- `DELAY n`

  – The time (in seconds) between adjusting a shim and reading the new lock level. In the example tune file, DELAY is set to one second which is usually enough for the lock level to settle.

- `SET <shim> w c`

  – Set the maximum step size (width) and the convergence limit for the `SIMPLEX` command. These parameters can be set for each shim separately. An example is:

  SET Z1 20 3

  `TIMES n`

  `..`

  `END`

Loop structure; all statements within the loop will be executed `n` times. Nested loops are possible to a depth of five.

Commands in a Tune File

- `ROTATION ON WAIT`

  – Switches the sample rotation on, using the spin rate currently set on the BSMS unit.

- `ROTATION OFF WAIT`

  – Switches the sample rotation off.

- `RSH`, `RSH <filename>`

  – Reads a shim file. If an argument is specified, RSH will read the corresponding shim file. If not, it will read the shim file with the name of the solvent defined by the acquisition parameter SOLVENT.

- `Z s i`, `Z2 s i`, ..., `XY s i`

  – Optimizes single shims. These commands take two arguments:

  s = step size; the shim increment used as long as the lock level increases

  i = iterations; the maximum number of steps after passing the maximum

A shim is first changed **s** units in one direction. If this increases the lock level, the shim is changed again **s** units in the same direction. This is repeated until a shim change decreases the lock level. Then the direction of change is reversed and the step size is reduced. This process is continued until one of the following criteria has been met:

- The lock level has not changed significantly during the last step.
- The maximum number of iterations (i) has been performed.
- The step size has been reduced to one.

Examples of shim commands are:

  Z 10 3

  Z2 10 3

  Z3 10 3

- `SIMPLEX <shim1 shim2 etc.>`

- – Optimizes the specified shims according to the simplex algorithm which can be used for lock level and FID area shimming. SIMPLEX uses the step size and convergence limit defined by the SET statement (see above). The simplex algorithm is described in the *example* file (see INPUT FILES below).

- `AUTOSHIM ON <shim1=m, shim2=n, ...>`

  - – Switches on the automatic shimming function on the shim unit after the **tune** command has finished. This command allows you to adjust the shims continuously during the entire experiment. Only the shims that are specified as arguments to the `AUTOSHIM ON` command will be optimized. For each shim, you can specify the step size used for auto shimming. If you do not specify the step size, the default value of 5 is used. An example of this command is:

- `AUTOSHIM ON Z1=2 Z2`

  - – Since this command becomes effective after the **tune** command has finished, it can be specified anywhere in the tune file.

- `AUTOSHIM OFF`

  - – Switches off automatic shimming on the shim unit. It makes sure that automatic shimming is switched off when it was switched on before, either from the last executed tune file or manually from the BSMS keyboard.

> ℹ Note that it would not make sense to use AUTOSHIM ON and AUTOSHIM OFF within one tune file.

- `LOCKPHASE s i`

  - – Optimize the lock phase. This command takes two arguments:

    s = step size

    i = iterations; the maximum number of steps

  - – As an alternative to the automatic shimming with **tune**, you can optimize the shims manually from the BSMS keyboard. If the shims are far away from the optimum, you can read a standard shim set with the command **rsh** and then do manual shimming.

An alternative to the simplex procedure in **tune** is the AU program **simplex**.

INPUT FILES

- <tshome>/exp/stan/nmr/lists/group/
  - – *example* - tune file
- <tshome>/conf/instr/<instrum>/prosol/<probeID>/
  - – *tunefile* - tune file for the current probe (input for **tune .sx**)

USAGE IN AU PROGRAMS

- TUNE(tunefile)
- TUNESX

  Eexecutes the command **tune .sx**

SEE ALSO

edtune [▷ 101], *rsh, wsh, vish [▷ 103]*, *delpar, delgp, delsh [▷ 190]*, (gradshim)

Shim Commands

6   Probe Commands

This chapter describes the commands which are involved in probe handling. This includes probe selection, the tuning and matching procedure and sample insert and eject.

6.1   atma

NAME

atma - Automatic tuning and matching of ATM probes.

USAGE

atma [exact | course] [high].

Type **atma ?** for a complete list of options.

DESCRIPTION

The command **atma** performs tuning and matching of an ATM probe. It will automatically perform the following steps:

- Stop the sample rotation if it is on.
- Read the nucleus with the lowest frequency as it was set up with **edasp**.
- Determine the optimum sweep width and number of steps.
- Tune and match the probe.
- Repeat step 3 to 4 for all other nuclei which were set up with **edasp** in the order of increasing frequency.
- Turn on the sample rotation if it was on before **atma** was started.

The command **atma** can be used with various options, for example:

- **atma exact** will determine the optimum tuning and matching more precisely then **atma** without an argument and will therefore be slower.
- **atma course** will determine the optimum tuning and matching less precisely then **atma** without an argument and will therefore be faster.
- **atma high** will start with the nucleus with the highest frequency, and tune and match the probe for each nucleus in the order of decreasing frequency.
- **atma storeWobb <x>** will automatically save the wobble curve after optimisation. **<x>** stands for the respective procno of the used dataset.

  If more than one nucleus should be stored than the respectively next wobble curve n+1 will bei stored in the procno number **<x+1>** (see also command **wbwr**).

  Type **atma ?** for a complete list of options.

On ATM probes, **atma** can be used instead of the **wobb**. These two commands differ in the following respects:

- **atma** is fully automatic whereas **wobb** requires the user to perform the tuning and matching manually.
- **atma** automatically determines the optimal sweep width and number of steps whereas **wobb** uses the values of WBSW and WBST, respectively.

- **wobb** must be terminated with **stop** or **halt** whereas **atma** automatically finishes when optimum tuning and matching is reached. If you want to interrupt **atma**, you can do that with the command **kill**.

Automatic tuning and matching is not only convenient, it also allows you to tune and match the probe during automaton. In IconNMR, you can choose to do that before each experiment, after each sample change or after each solvent change.

For more information on the tuning and matching process, see **wobb**.

For more information on the command **atma** and ATM probes, see

Help => **Manuals** => [**Technical Manuals**] **Automatic Tuning/Matching**.

INPUT PARAMETERS

- NUC1 - NUC4 - nuclei as defined with **edasp**

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters
- <tshome>/conf/instr/<instrum>/
  - *nuclei* - nuclei table
- <tshome>/prog/wobble/
  - *acqu_go4* - wobble parameters
  - *Pulsprog_X* - wobble pulse program

SEE ALSO

atmm [▶ 110], *wobb [▶ 119]*, *wbwr [▶ 118]*

6.2   atmm

NAME

atmm - Manual tuning and matching of ATM probes.

USAGE

atmm [manWbsw]

Type **atmm ?** for a complete list of options.

DESCRIPTION

The command **atmm** is a manual tuning and matching procedure for ATM probes. It is not needed very often because ATM probes are designed for automatic tuning and matching with **atma**. Sometimes, however, the probes resonance frequency is so far away from the optimum that **atma** would take very long to finish or would fail. In practice, this only occurs for certain nuclei at broadband probes. In that case, **atmm** allows you to manually tune and match the probe for that nucleus.

atmm performs the following steps:

- It reads the nucleus with the lowest frequency as it was set up with **edasp**.
- It determines the optimum sweep width and number of steps.

- It shows the reflected power (tuning/matching curve) in the TopSpin acquisition data field.
- It opens the **atmm** control window from where you can:
  - Select the nucleus for which you want to tune and match the probe. By default, the nucleus with the lowest frequency is selected.
  - Perform course tuning/matching on broadband probes. This is the equivalent of setting the sliders on a non ATM probe to predefined numbers.
  - Perform fine tuning/matching while observing the curve in the TopSpin acquisition data field. This is the equivalent of turning the knobs or moving the sliders on a non ATM probe. See **wobb** for a description on how to reach the optimum tuning and matching.
  - Click **File => Quit** to finish the tuning/matching process.

If **atmm** is used with the argument **manWbsw**, it does not determine the sweep width and number of steps automatically but interprets the parameters WBSW and WBST, respectively. **atmm ?** will show you a complete list of arguments.

The difference between **atmm** and **wobb** is that:

- **atmm** can only be used on ATM probes.
- **atmm**, when used without arguments, automatically determines the optimum sweep width and number of steps whereas **wobb** uses the values of WBSW and WBST, respectively.
- **atmm** allows you to optimize tuning and matching from TopSpin whereas **wobb** requires you to turn the knobs (or move the sliders) on the probe.

For more information on the tuning and matching process, see **wobb**.

For more information on the command **atmm** and ATM probes, see

Help => Manuals => [**Technical Manuals**] **Automatic Tuning/Matching**.

INPUT PARAMETERS

- NUC1 - NUC4 - nuclei as defined with **edasp**
- WBSW - sweep width (only used by **atmm Manwbsw**)
- WBST - number of steps (only used by **atmm Manwbsw**)

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters
- <tshome>/prog/wobble/
  - *acqu_go4* - wobble parameters
  - *Pulsprog_X* - wobble pulse program

SEE ALSO

atma [▷ 109], *wobb [▷ 119]*

6.3   edprobe

NAME

edprobe – Edit the probe table

Probe Commands

DESCRIPTION

The command **edprobe** opens the probe table editor:
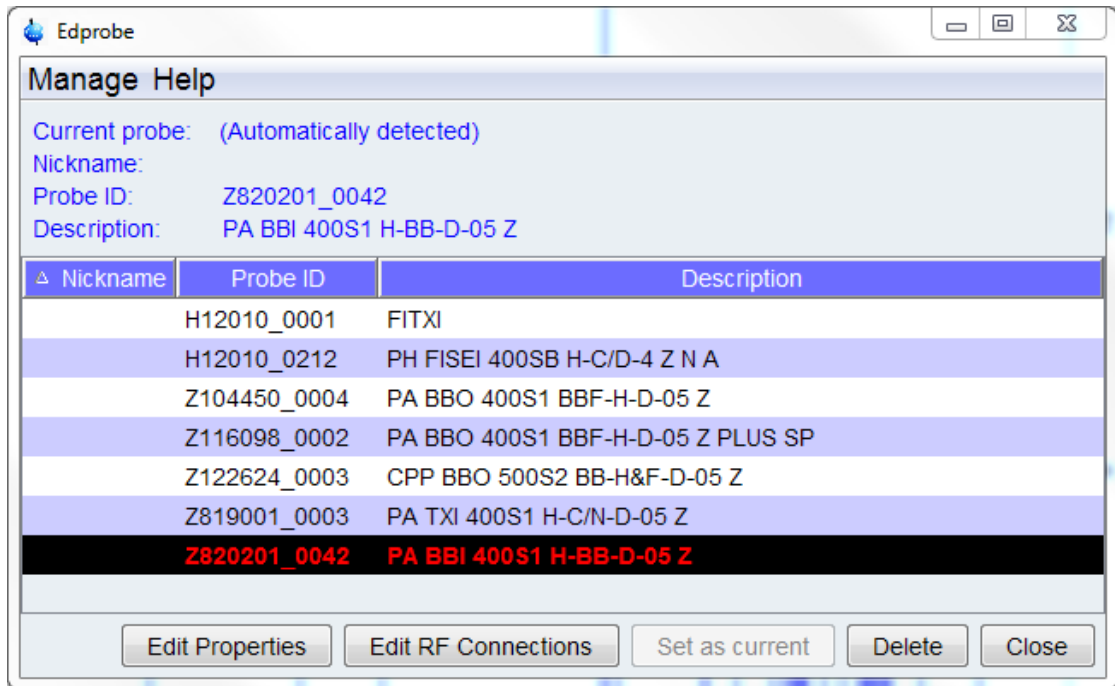
Figure 6.1: edprobe-with_PICS2

The table has columns for Nickname (optional), Probe ID and description of configured probes. The Nickname can be freely chosen to identify a probe for the user more easily.

The entry which is defined as the current probe is highlighted.

The probe table has a menu **Manage** in the upper left corner that offers the following commands:

Add a new probe

Import old probes

View Properties

Print

Flash Pics

Add a new probe

The probe dialog editor will be opened where you can define a new probe.

The name of the new probe will be the combination of Part number and Serial number.

With Template, one can select typical probe configurations.
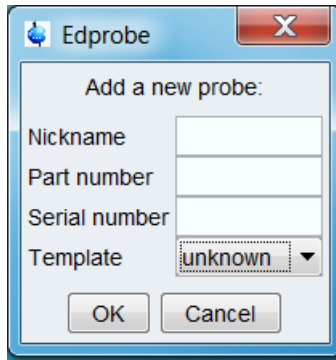
Figure 6.2: addNew

Import old probes

If older (edhead) definitions of probes are available in the current installation, they are listed in the import table. By selecting an entry, all settings of this probe are imported to the probe table.
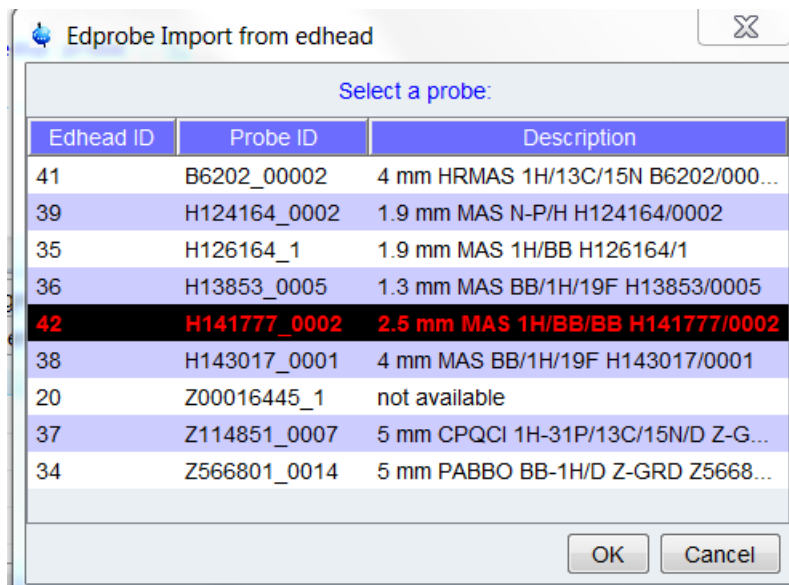
Figure 6.3: import

View Properties

Opens the Probe Properties of the selected entry for viewing only.

Print

Opens the printing dialog for the Edprobe Probes Table.

Flash Pics

Allows to flash the PICS of the current probe

Figure 6.4: flashpics

This requires that the probe is connected with a PICS cable. Only digitally signed PICS files can be used to flash the probe.

At the bottom of the edprobe dialog you can select following buttons:

Figure 6.5: buttons

Edit Properties

Opens the Probe Properties of the selected entry for editing.
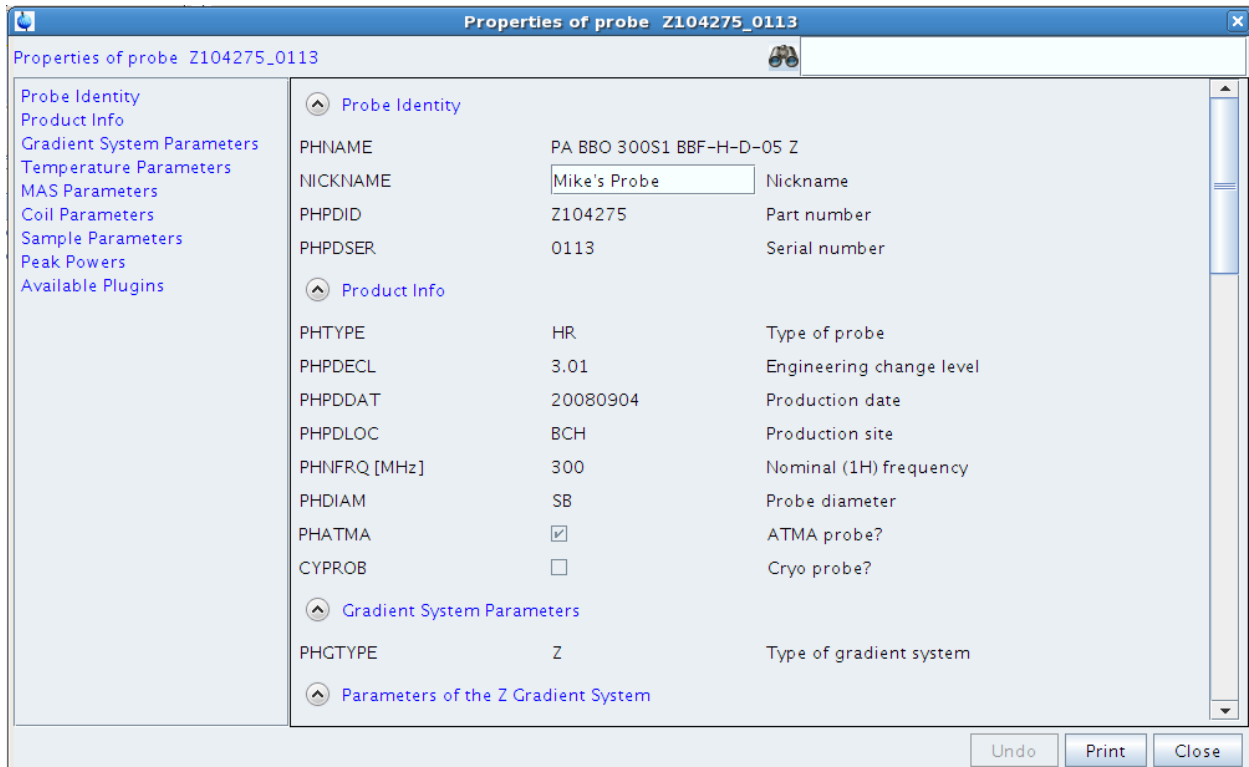
Figure 6.6: edprobe-with_PICS3

If the probe is connected with a PICS cable, only a view properties can be changed, see figure above. For probes without PICS, all properties can be adjusted, see figure below.

Figure 6.7: edprobe-properties

Edit RF Connections

It displays a new window where the wiring between the amplifier or preamplifier and the probe connectors can be defined, see figure below.

Set as current

> The highlighted entry will be defined as current probe

Delete

> The selected entry will be deleted (not possible for the currently installed probe).

Close

> Close the probe table editor.
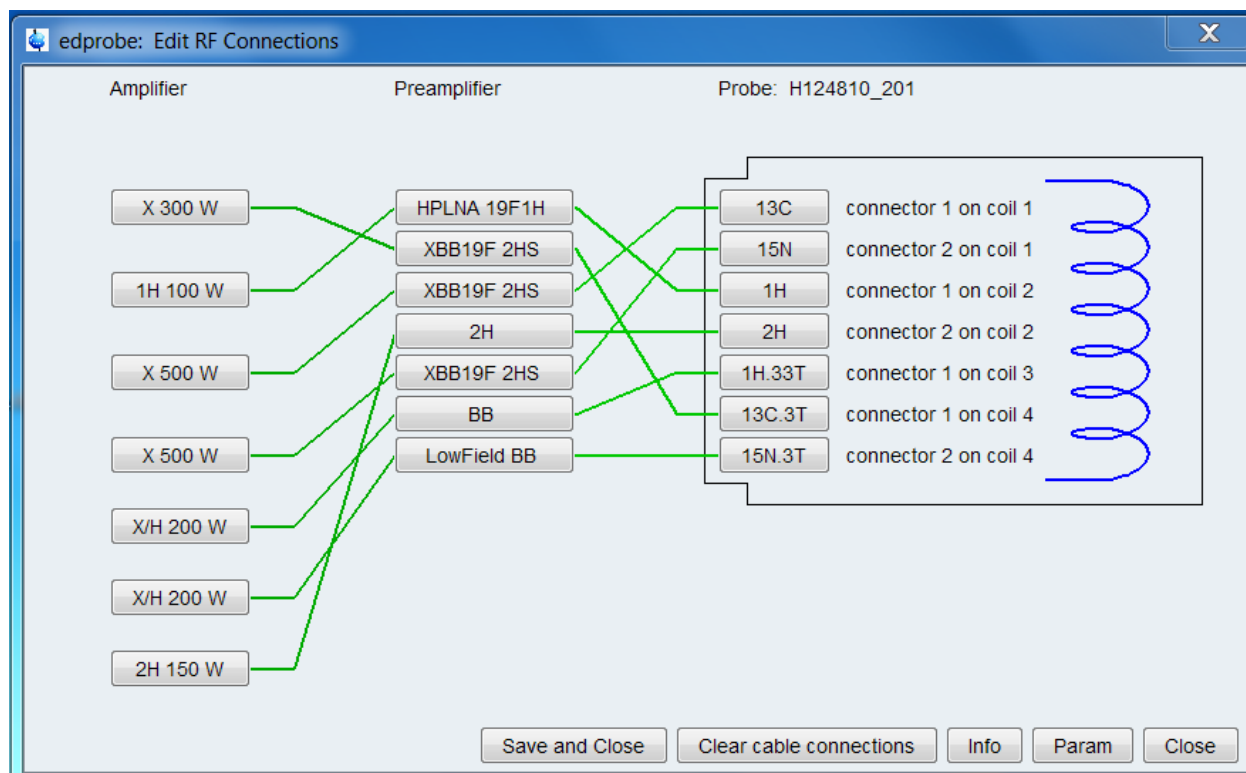
Figure 6.8: edprobe-rf

> By exiting edprobe, if current probe parameter have changed, the **Edit RF Connections** window is displayed where the preamplifier probe connection must be defined.

> The current probe as defined with **edprobe**, is interpreted by the commands **edprosol**, **edlock**, **lock**, **lopo** and **setpre** as well as during IconNMR automation. At the end of an acquisition the current probe is stored as the acquisition status parameter PROBHD (see **dpa**).

The input files of edprobe command:

> ▪ <tshome>/conf/instr/probehead - current probes name
> ▪ <tshome>/conf/instr/ probeheads/<part #>_<serial #>.par - probe parameters

SEE ALSO

> *cf [▷ 49]*, *atma [▷ 109]*, *atmm [▷ 110]*, *wobb [▷ 119]*

6.4 ej, ij

NAME

ej - Eject the sample from the magnet.

ij - Insert the sample into the magnet.

DESCRIPTION

The command **ej** ejects the sample from the magnet. It switches on the air flow and keeps it on until it is explicitly switched off, for example with **ij**.

The command **ij** inserts the sample into the magnet. This command is used when the magnet air flow is on and the sample is floating on top of it. **ij** gradually reduces the air flow to zero so that the sample is lowered into the magnet.

As an alternative to the command **ej** and **ij**, you can also press the LIFT ON/OFF button on the BSMS keyboard.

The SAMPLE DOWN indicator in the shimming panel of the BSMS keyboard is lit when the sample is positioned correctly in the probe. The SAMPLE UP indicator is lit when the sample has reached the top of the magnet. The SAMPLE MISSING indicator is lit if the sample is somehow positioned between the magnet top and the probe, either moving up or down or being stuck. The latter can happen, for instance, when the air flow is too high.

USAGE IN AUTOMATION

EJ

IJ

SEE ALSO

ro [▷ 117]

6.5 ro

NAME

ro - Switch the sample rotation on or off

SYNTAX

ro [acqu | on | off | off wait]

DESCRIPTION

The command **ro** switches the sample rotation on or off. When entered without arguments, it opens a dialog box (see the next figure).
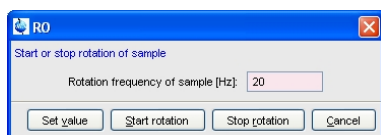
Figure 6.9: ro

Here, you can set the rotation frequency and start/stop the rotation.

ro takes up to two parameters and an be used as follows:

- **ro on**

  Switches the sample rotation on with the spin rate currently set on the BSMS keyboard (visible when you press the SPIN RATE key).

- **ro acqu**

  Sets the spin rate to the value of the acquisition parameter RO, then switches rotation on and waits for 60 seconds. If spin rate has not been reached within that time, an error message appears.

- **ro off**

  Switches the sample rotation off.

- **ro off wait**

  Switches the sample rotation off and waits until the rotation has reached zero. During the waiting time the BSMS unit cannot be accessed by other commands.

- **ro <value>**

  Sets the acquisition parameter RO to <value>.

As an alternative to the command **ro**, you can press the SPIN ON/OFF, and SPIN RATE keys on the BSMS keyboard or BSMS display (command **bsmsdisp**).

INPUT PARAMETERS

- RO - sample rotation frequency (input for **ro acqu**)

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters

USAGE IN AU PROGRAMS

- ROT

  Executes the command **ro acqu**.

- ROTOFF

  Executes the command **ro off wait**.

SEE ALSO

ej, ij [▷ 117]

6.6   wbwr

NAME

wbwr - save wobble curve to disk

DESCRIPTION

The command **wbwr** saves the wobble curve to disk.

> **i** Please note that the command is only valid in the wobble mode(first enter command wobble).

When typing **wbwr** in the command line and the user is prompted to enter a PROCNO which will later on be used to store the wobble curve in the next dialogue:
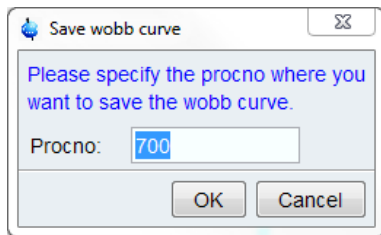
Figure 6.10: wbwr

As an alternative to **wbwr** entered on the command line, the button can be clicked in the TopSpin toolbar.

The wobble curve is stored as processed data (1r) and not as raw data (fid.). This provides the additional option of using standard processing tools like e. g. multiple display.

In addition **wbwr** can directly be entered together with an additional parameter denoting the PROCNO that shuld be used, i.e. "**wbwr 4**".

OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/<procno>/
  - 1r - the wobble curve
  - procs - processing status bar

SEE ALSO

wobb [▷ 119]

6.7 wobb

NAME

wobb - Manual tuning and matching of the probe.

USAGE

wobb [high][ext50][raw][f1|f2]

DESCRIPTITION

The command **wobb** allows you to manually tune and match the probe. It opens the wobble window showing the wobble curve (see the next figure).
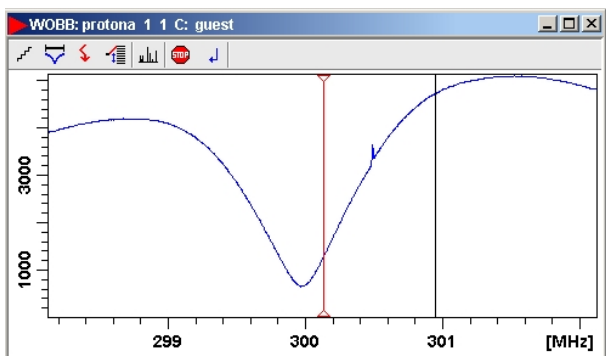
Figure 6.11: wobb

The buttons have the following functions:

Change the number of wobble steps [**wbst**].

Change the wobble sweep width [**wbsw**].

Change the wobble frequency.

Switch to the next channel/nucleus (if available).

Stop the wobble procedure.

Turn the tuning and matching knobs on the probe until the wobble curve is exactly in the middle and its minimum reaches the zero line (see the next figure).

An NMR probe must be tuned and matched because it is a resonance circuit. If its resonance frequency and impedance are the same as the transmitter frequency and impedance, respectively, the full transmitter power is transferred to the probe. However, if either or both are different, part of the transmitter power is reflected by the probe. This results in a longer 90° pulse or, for a given pulse length, a smaller flip angle.

> Note that a multi nuclear probe has a resonance circuit for each nucleus and each nucleus must be tuned and matched separately.
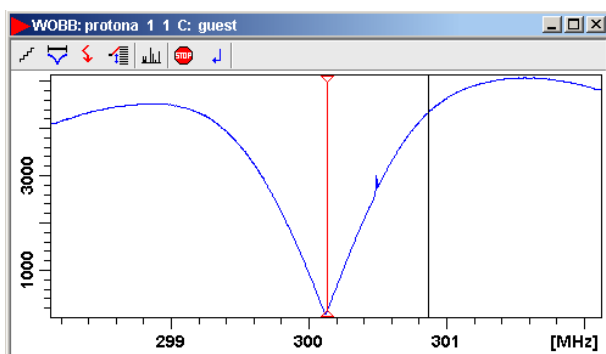
Figure 6.12: Command wobb

The command **wobb** is normally used without argument. It can, however, be used with arguments. For example:

- **wobb high**

  Starts with the nucleus with the highest frequency and continues in the order of decreasing frequency.

- **wobb ext50**

  Uses an external 50 ohm resistor as a reference.

- **wobb f1**

  Starts with frequency channel f1. It continues with the next higher frequency or, if the argument **high** is also used, with the next lower frequency.

- **wobb f2**

  Starts with frequency channel f2. It continues with the next higher frequency or, if the argument **high** is also used, with the next lower frequency.

The **wobb** command allows you to optimize both the probes resonance frequency (tuning) and impedance (matching). It sends a low power RF signal to the probe and sweeps that signal over a certain frequency range. The number of steps (frequencies) within that range is defined by the acquisition parameter WBST. The width of the frequency range is defined by WBSW. The center frequency depends on the nucleus; SFO1 for NUC1, SFO2 for NUC2 etc. The deviation of the probe impedance from the nominal 50 ohm is shown as function of the frequency in the TopSpin data field. This is the so called wobble curve. At the probes resonance frequency, the curve shows a dip, the minimum reflected power. Tuning the probe means adjusting its resonance frequency until it reaches SFO1. Matching the probe means adjusting its impedance until the reflected power reaches zero.

The entire wobble procedure involves the following steps:

- Stop the sample rotation if this is on, for example with **ro off**, or by pressing the SPIN ON/ OFF button on the BSMS keyboard.

- Setup the nucleus or nuclei for the current experiment with **edasp**.

  This will automatically set the parameters SFO1, SFO2 etc.

- Click **Acquire => Observe fid window** or enter **acqu** to switch to the TopSpin acquisition menu. If, however, tuning and matching is observed on the preamplifier, this step can be skipped (see below).

- Enter **wobb**. The wobble curve will appear in the TopSpin data field showing a dip at a certain frequency. At the center of the data field, you will see a vertical line. If you do not see the dip, it probably lies outside of the data field. In that case, you should click the

  button or enter **wbchan** to increase the sweep width. You can do this while **wobb** is running.

- When the dip is visible, you can start tuning and matching as follows:

  – Turn the tuning knob until the dip lies across the vertical line.

  – Turn the matching knob until the dip has reached a minimum. Matching influences tuning, so the dip probably moves away from the center.

  – Turn the tuning knob until the dip lies at the center again. Tuning influences matching, so the dip probably moves up again.

  – Turn the matching knob until the dip reaches a minimum again.

  – Continue this process until the dip lies exactly across the vertical line and reaches the x-axis.

- In case of a multi nuclei experiment, you have to switch to the next the nucleus. Note that **wobb** automatically starts with the nucleus with the lowest basic frequency. You can switch to the nucleus with the next higher frequency in two possible ways:

  – Press **Channel Select** at the HPPR. This will automatically select the nucleus with the next higher frequency.

  – Click   or enter **wbchan** in TopSpin. Answer the question "Do you want to change the nucleus" with **yes**.

- Repeat step *5 [▷ 119]* for the current nucleus.
- If your experiment involves more than two nuclei, repeat step *6 [▷ 119]* for each further nucleus.

- Click [STOP] or enter **stop** on the command line to finish the wobble procedure.

A probe has a tuning knob (labeled **T**) and matching knob (labeled **M**) for each resonance circuit. Most probes have two, one for 1H and one for X-nuclei. When the tuning knob reaches the end of its range before the probe is properly tuned, you should turn it to the middle of its range, adjust matching, then tuning, then matching etc. A similar procedure can be used if the matching knob reaches the end of its range.

The process of tuning and matching can also be followed on the HPPR preamplifier. Some people find that more convenient and it is necessary when the computer screen is not visible from the position of the probe. The horizontal row of LED's indicates tuning, the vertical row indicates matching. When you turn the tuning or matching knob at the probe, you will see how the number of lit LED's changes. The probe is perfectly tuned when only one LED (a green one) is lit. The same holds for matching. In practice, proper tuning and matching means that only green LED's are lit. If the LED update seems to be very slow, this might be caused by the time consuming update of the wobble curve in the TopSpin acquisition display.

In that case, you can simply click [↵] to close the wobble window.

Broadband probes usually have sliders for tuning and matching rather than turning knobs. These have the advantage that their positions are numbered. The default slider positions for each nucleus are usually written on cards that are kept with the probe. When wobbling is started with these default values, only some fine tuning and matching is required.

The probe resonance frequency and impedance is dependent on the sample. This can be a problem in automation, where several samples are measured but the probe is only matched and tuned on one of them. Bruker ATM probes support automatic tuning and matching which can be preformed on every sample during automation (see description of the commands **atmm** and **atma**).

INPUT PARAMETERS

- Set by the user with **eda** or by typing **wbst**, **wbsw** etc.:
  - WBST - number of wobble steps
  - WBSW - wobble sweep width
- Set by the user with **edasp** or **eda** => NUC1:
  - NUC1 - NUC4 - the nuclei for which the probe is tuned and matched
  - SFO1 - SFO4 - irradiation frequency

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters
- <tshome>/prog/wobble/
  - *acqu_go4* - wobble parameters
  - *Pulsprog_X* - wobble pulse program

SEE ALSO

7   Parameter Handling Commands

This chapter describes commands which are involved in parameter handling. This involves the setting of acquisition parameters, probe/solvent dependent parameters and nuclei assignment. Furthermore, the usage of predefined parameters sets is discussed.

7.1   ased

NAME

ased - Edit acquisition parameters used in pulse program

DESCRIPTION

The command **ased** opens a dialog box with the acquisition parameters which are used for the current experiment (see the next figure).
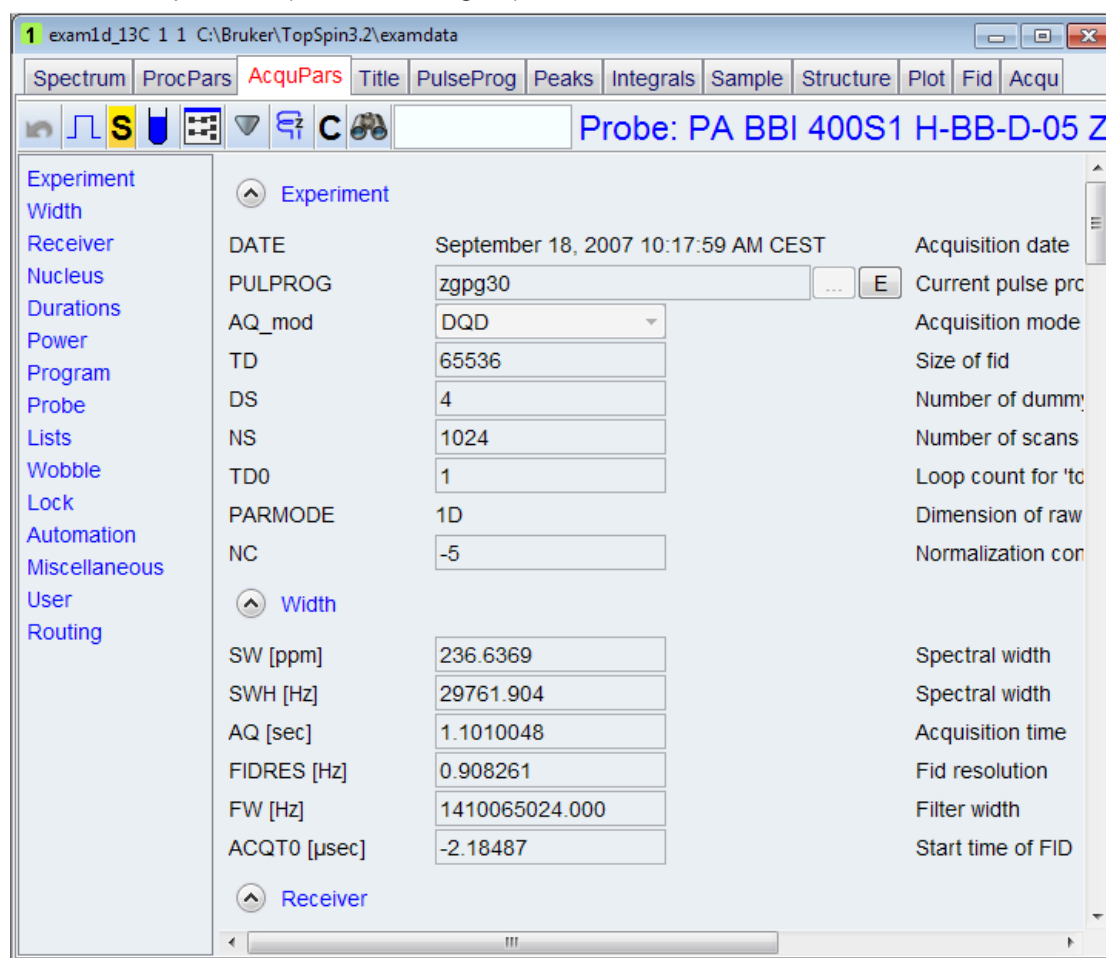
Figure 7.1: ased

This means that **ased** shows much less parameters then **eda** which shows all acquisition parameters. Entering **ased** on the command line is equivalent to clicking the **AcquPars** tab and then clicking the ⊓ button.

The following buttons are available:

Undo the last modification (unused for status parameters).

Force parse of the pulse program.

Switch to all acquisition parameters.

Set probe/solvent dependant parameters.

Open the nuclei and routing table.

Search for the parameter specified in the search field.

ased compiles and interprets the pulse program defined by PULPROG. For pulses, delays and constants, the parameter description in the right column of the **ased** window is taken from the comment section at the end of the pulse program.

INPUT AND OUTPUT PARAMETERS

- For all experiments:
  - PULPROG - pulse program used for the acquisition.
  - TD - time domain; number of raw data points.
  - NS - number of scans.
  - DS - number of dummy scans.
  - SWH - spectral width in Hz.
  - AQ - acquisition time in seconds.
  - RG - receiver gain.
  - DW - dwell time.
  - DE - pre-scan delay.
- For each frequency channel defined with **edasp**:
  - NUCx - nucleus for channel x.
  - SFOx - irradiation frequency for channel x.
- All delays, pulse lengths, power levels etc. defined in the pulse program, e.g.:
  - D[1] - relaxation delay.
  - P[1] - 90° pulse length.
  - PL[1] - power level for pulse.
  - PCPD[1] - CPD pulse length.

INPUT FILES

- <tshome>/exp/stan/nmr/lists/pp/
  - The pulse program defined by PULPROG
- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters

OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters

SEE ALSO

7.2   dpa

NAME

dpa - Display the acquisition status parameters

DESCRIPTION

The command **dpa** displays the acquisition status parameters (see the next figure).
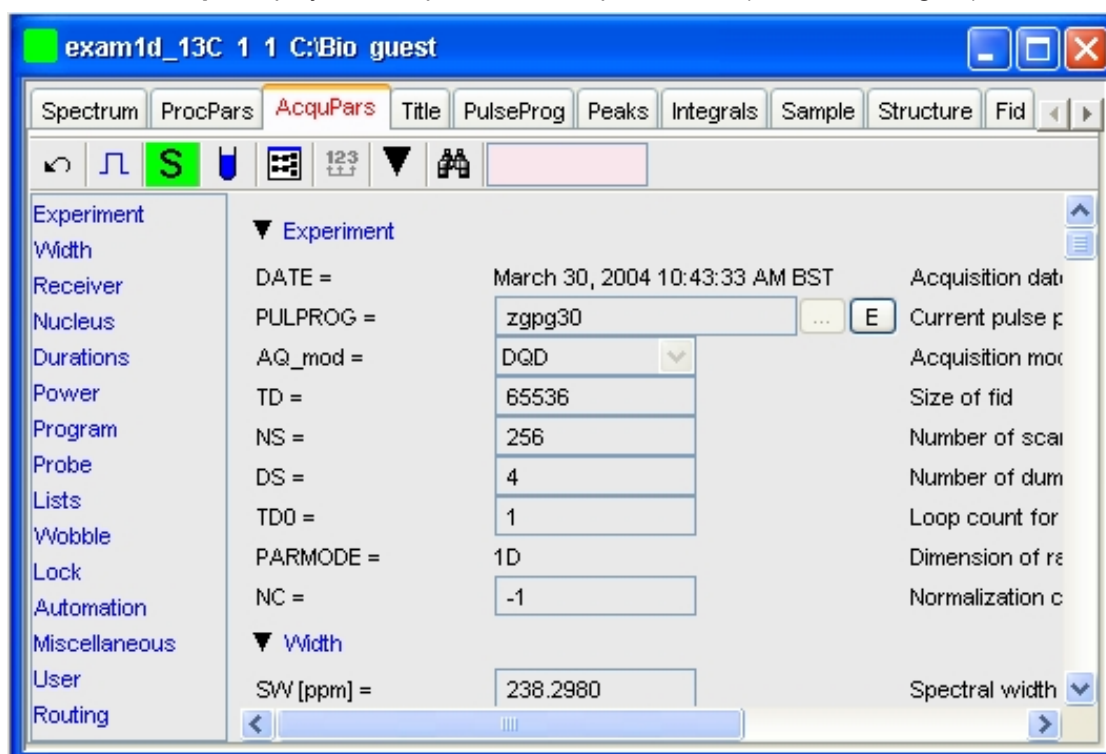
Figure 7.2: dpa

These are set by acquisition commands and represent the status of the raw data. Entering **dpa** on the command line is equivalent to clicking the **AcquPars** tab and then clicking the ⑤ button.

The following buttons are available:

↶ Undo the last modification (unused for status parameters).

⊓ Show pulse program parameters .

⑤ Switch to acquisition parameters.

⬛ Set probe/solvent dependent parameters.

⊞ Open the nuclei and routing table.

⚲ Search for the parameter specified in the search field.

Acquisition status parameters can also be viewed by entering their names on the command line. For example:

- **s ns**

  Displays the acquisition status parameter NS.

- **s td**

  Displays the acquisition status parameter TD.

On multidimensional data, the parameter values for each direction are shown.

INPUT FILES

- <dir>/data/<user>/<name>/nmr/<expno>/

  - *acqus* - acquisition status parameters

On 2D and 3D data the files *acqu2s* and *acqu3s* are used for the indirect directions, respectively (see also chapter *Parameter Files [▷ 13]*).

SEE ALSO

eda [▷ 126], *dpa [▷ 125]*

7.3 eda

NAME

eda - Edit acquisition parameters.

DESCRIPTION

The command **eda** opens a dialog box in which you can set all acquisition parameters (see the next figure).
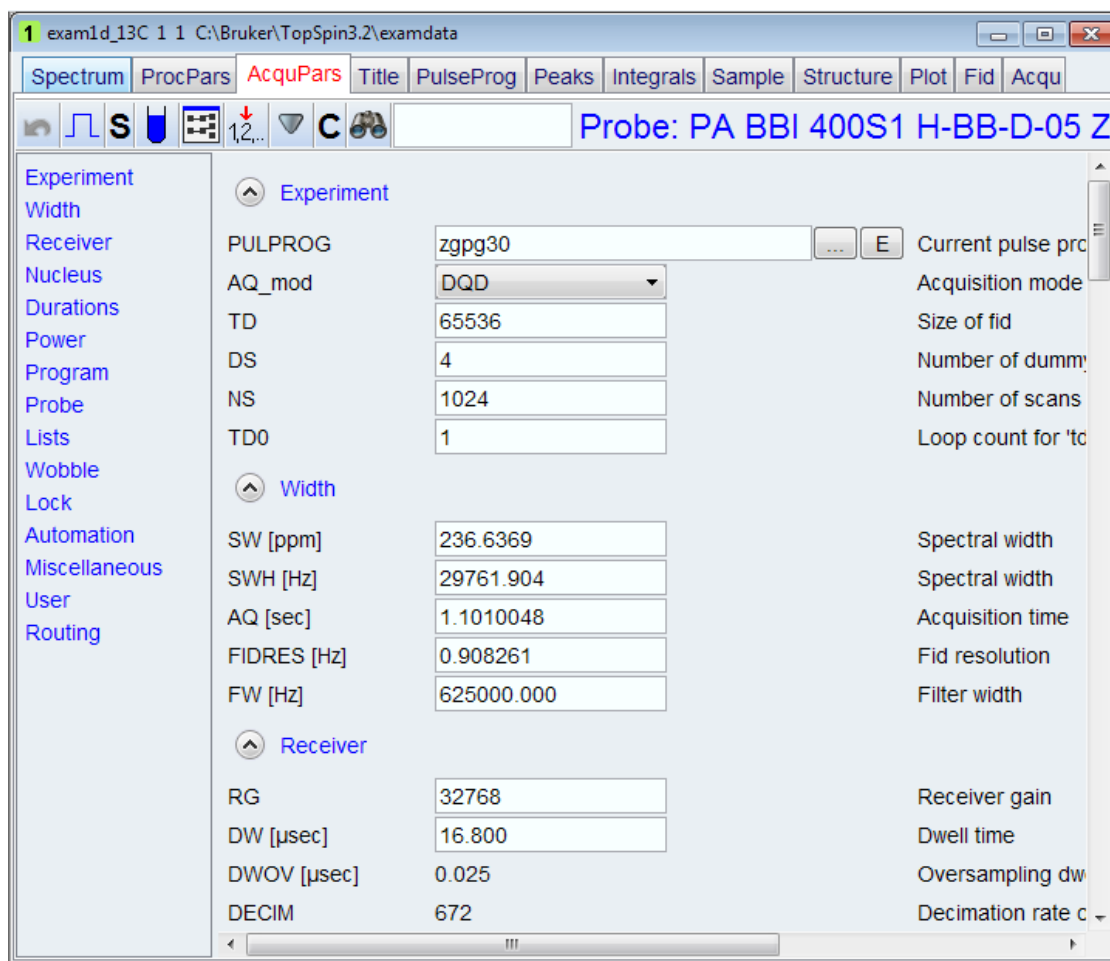
Figure 7.3: eda

Entering **eda** on the command line is equivalent to clicking **AcquPars** in the tab bar of the data window.

The following buttons are available:

Undo the last modification. Can be used repeatedly.

Show pulse program parameters.

Switch to acquisition status parameters.

Set probe/solvent dependant parameters.

Open the nuclei and routing table.

Change raw dataset dimensionality (parameter PARMODE).

Collapse/expand all parameter sections.

Search for the parameter specified in the search field.

Inside the parameter editor, you can do the following actions:

- Click an acquisition step, e.g **Experiment** at the left of the dialog box. The step becomes highlighted and the corresponding parameters will appear in the right part of the dialog box.
- Click in a parameter field, e.g. TD to set the parameter value. It is automatically stored.
- Hit the **Tab** key to jump to the next parameter field.
- Hit **Shift**-**Tab** to jump to the previous parameter field.
- Use the scroll bar at the right of the dialog box to move to parameters further up or down in the dialog box.

> ℹ️ Note that the **...** and **E** button to the right of the PULPROG parameter allow you to show the pulse program list or edit the current pulse program, respectively.

Parameters can also be set by entering there names on the command line. A dialog window will appear where you can enter the parameter value(s). For example:

td

On a 1D data set.
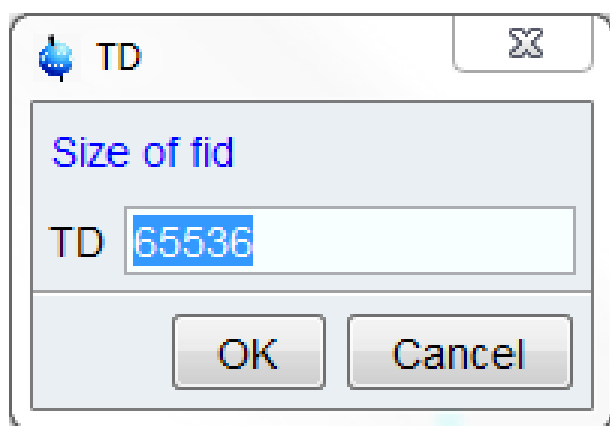
Figure 7.4: TD - Size of fid

or on a 2D data set:
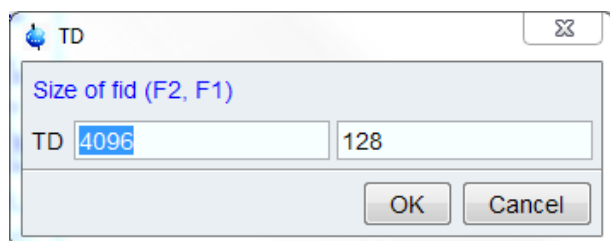
Figure 7.5: TD - Size of fid (F2, F1)

Alternatively, you can specify the parameter value as an argument on the command line, For example;

td 8k

The time domain will be set to 8k.

INPUT AND OUTPUT PARAMETERS

All acquisition parameters.

INPUT FILES

- <tshome>/classes/prop/
  - *pared.prop* - parameter properties file.
- <tshome>/exp/stan/nmr/form/
  - *acqu.e* - format file for **eda.**

INPUT AND OUTPUT FILES

- <dir>/data/<user>/<name>/nmr/<expno>/
  - *acqu* - acquisition parameters for the acquisition (direct) direction.
  - *acqu2* - acquisition parameters for F1 (2D) or F2 (3D) indirect direction.
  - *acqu3* - acquisition parameters for the F1 direction (3D).

SEE ALSO

7.4 edasp

NAME

edasp - Set up nuclei and spectrometer routing

DESCRIPTION

The command **edasp** allows you to set up the nuclei and the spectrometer routing for the current experiment. It opens the routing table showing the current nuclei selection and amplifier routing. Here you can specify the nucleus or nuclei to be used for the experiment. For the f1 logical frequency channel, you can click the NUC1 button and select a nucleus from the appearing list. The basic frequency, BF1, is automatically set. Then you can specify the frequency offset; OFSH1 for 1H or OFSX1 for X nuclei. The irradiation frequency SFO1 is automatically calculated as the sum of the basic frequency and frequency offset.

> **i** Note that the parameters NUC1, BF1 and SFO1 also appear in **eda** but cannot be set from there.

However, the frequency offset (OFSH1 or OFSX1) appears in **eda** as O1 and can be set from there. In a multi nuclear experiment, you can set up the parameters for channel f2, f3 etc. in the same way as described for f1.

The routing table also shows the spectrometer routing. The hardware elements which have been detected when the spectrometer was configured (with **cf**) will appear in this list. When you select a nucleus for a certain channel, the default routing is automatically set. Normally, this can be left as it is. Only in very special cases it is useful to change it. The routing table consists of the following stages:

Channel---FCU---Amplifier---Swibox/I ---Swibox/O---Preamplifier

You can set up the routing by connecting an element from one stage to an element of the next stage. A connection can be installed or removed by clicking the two elements involved. The routing must follow the following rules:

- Between the logical frequency channels and FCU's any combination is allowed but only one to one.

- An FCU can be connected to one amplifier only but an amplifier can be connected to multiple FCU's. Furthermore, FCU1 can only be connected to the first three amplifiers and FCU2 can only be connected to the first four amplifiers.

- The first four amplifiers are automatically connected to switchbox input in a fixed way. These connections cannot be changed.

- Between the switchbox input and switchbox output, any connection is allowed but only one to one.

- Between the switchbox output and the Preamplifier modules, any connection is allowed but only one to one. Make sure that the cables are connected accordingly.

- The output of the fifth and sixth amplifier must be connected directly to the preamplifier modules. Any combination is allowed but only one to one and only to preamplifier modules which are not connected to a switchbox output.

Avance spectrometers which are used for solid state experiments usually contain high power amplifiers with two output stages:

- A low power output: typically 150 W for 1H or 300 W for X nuclei.

- A high power output: typically 1000 W for 1H or X nuclei.

The second output appears as an extra stage in the routing table to the right of the switchbox. If you use the second output stage, the switchbox is always bypassed.

If you make a routing connection which is not allowed, you will get a message that it is illegal. You must remove it before you can save the routing table. Some connections are allowed but not recommended. In that case, you only get a warning.

At the bottom of the routing table you will find the following buttons:

- SAVE - save the routing table and quit.

- SWITCH F1/F2 - exchange the F1 and F2 nucleus including the frequency and frequency offset.

- SWITCH F1/F3 - exchange the F1 and F3 nucleus including the frequency and frequency offset.

- DEFAULT - set the default amplifier routing for the current nuclei.

- CANCEL - quit without saving any changes.

- PARAM - shows the routing parameters.

In a 2D dataset, NUC1 has a different function in the two directions. In F2, it works like in 1D experiments; it opens the routing table. In F1, it opens the nuclei list and allows you to select the nucleus for this direction. The same principle holds for 3D data; in F3, NUC1 opens the routing table, in F2 and F1 it opens the nuclei list.

The spectrometer routing is stored under the current dataset in the acquisition parameters FCUCHAN, RSEL, SWIBOX, PRECHAN and HPMOD (see the description of these parameters). These can be viewed by clicking the PARAM button in the routing table.

> **i**   Note that these parameters appear in **eda** and can be set from there.

This, however is not very common as setting them from the routing table is much more convenient. In Bruker parameter sets (see **rpar**), the routing parameters have been set according to the configured hardware. This corresponds to the routing that will be set by clicking the **default** button in the routing table.

The connections between the Switchbox output and the Preamplifier modules are fixed; they cannot be changed. However, if you start **edasp** with the argument **setpreamp**, you can change these connections or remove them all by clicking **Clear preamplifier connections**. If

you make any changes here, you must make sure that the cables at the spectrometer are connected accordingly. The command **edasp setpreamp** is implicitly called as a part of spectrometer configuration (command **cf**).

INPUT AND OUTPUT PARAMETERS

Set graphically from **edasp** by connecting routing elements:

- FCUCHAN - logical frequency channel - FCU connections.
- RSEL - FCU - amplifier connections.
- SWIBOX - Switchbox input - Switchbox output connections.
- PRECHAN - Switchbox output - Preamplifier module connections.
- HPMOD - high power amplifier - Preamplifier module connections.

INPUT AND OUTPUT FILES

- <dir>/data/<user>/<name>/nmr/<expno>/
    - *acqu* - acquisition parameters.
- <tshome>/conf/instr/<INSTRUM>/
    - *specpar* - routing parameters.

SEE ALSO

edsp [▷ 75], *eda [▷ 126]*, *cf [▷ 49]*, *expinstall [▷ 76]*

7.5   getprosol

NAME

getprosol - Get probe and solvent dependent parameters

DESCRIPTION

The command **getprosol** reads the probe and solvent dependent parameters (the prosol parameters) and copies them to the corresponding acquisition parameters. Entering **getprosol** on the command line is equivalent to clicking the **AcquPars** tab and then clicking the ▐ button. The relations between the prosol and acquisition parameters are shown in table *Table 7.1 [▷ 132]*. These are the default relations which apply to most standard high resolution experiments. Protein, DNA/RNA and LC-NMR experiments require different relations which are defined in the files, *triple*, *triple_na* and *lcnmr*, respectively (see INPUT FILES). To use relations other then default, the so called relations file must be specified in the pulse program. You can do that by editing the pulse program (with **edpul** or **edcpul**) and adding the line:

```
prosol relations=<filename>
```

before the actual pulse sequence. To look at an example, you can enter **edpul lc2** or **edpul zgesgp**.

> Note that the file *default*, for default relations can, but does not need to be specified in the pulse program.

Description	edprosol	eda
F1 90° hard pulse length	P90[F1]	P[0], P[1]
F1 180° hard pulse length	P90[F1]*2	P[2]
F2 90 ° hard pulse length	P90[F2]	P[3]
F2 180° hard pulse length	P90[F2]*2	P[4]
F1 Tocsy spin lock 60° pulse length	PTOC[F1]*0.66	P[5]
F1 Tocsy spin lock 90° pulse length	PTOC[F1]	P[6]
F1 Tocsy spin lock 180° pulse length	PTOC[F1]*2	P[7]
F1 Roesy spin lock pulse length	PROE[F1]	P[15]
Gradient 1 pulse length	P_grad1	P[16]
F1 Tocsy trim pulse length	P_mlev	P[17]
Gradient 2 pulse length	P_grad2	P[19]
F3 90° hard pulse length	P90[F3]	P[21]
F3 90° hard pulse length	P90[F3]*2	P[22]
F1 HSQC trim pulse length	P_hsqc	P[28]
F2 Roesy spin lock pulse length	PROE[F2]	P[31]
F1 90° hard pulse power level	PL90[F1]	PL[1]
F2 90° hard pulse power level	PL90[F2]	PL[2]
F3 90° hard pulse power level	PL90[F3]	PL[3]
F1 Tocsy spin lock power level	PLTOC[F1]	PL[10]
F1 Roesy spin lock power level	PLROE[F1]	PL[11]
F2 CPD power level	PLCPDP[F2]	PL[12]
F2 Second CPD (bilev) power level	PLCPD2[F2]	PL[13]
F3 CPD power level	PLCPDP[F3]	PL[16]
F2 Homodecoupling power level	PLHD[F2]	PL[24]
Gradient recovery delay	D_grad	D[16]
F2 CPD pulse length	PCPDP[F2]	PCPD[2]
F2 CPD pulse length	PCPDP[F3]	PCPD[3]

Table 7.1: Default relations between prosol and acquisition parameters

Usage Of Getprosol With Command Line Options

The command **getprosol** (get probe and solvent dependent parameters from the edprosol table) can be called with three options:

- The nucleus.
- P90, the value for 90° hard pulse length.
- PL90, the value for 90°hard pulse power level.

The existing prosol parameterset, set up by **edprosol**, is not modified.

getprosol Nuc P90 PL90 recalculates standard hard power levels with:

PLx = 20*log (Px/P90) and the standard soft power levels PLSH1 - PLSH16 by:

PLSHx = 20*log (integFac*Px/totRot * (P90/90.0) where integFac (=SHAPE_INTEGFAC) and totRot (SHAPE_TOTROT) are read from the shapes *<TOPSPINHOME>/exp/stan/nmr/lists/ wave/PNSHx*

and

<TOPSPINHOMME>/exp/stan/nmr/lists/wave/Gaus1.100 (e.g.)

INPUT AND OUTPUT PARAMETERS

See table *Table 7.1 [▷ 132]*.

INPUT FILES

- <tshome>/conf/instr/<instrum>/prosol/relations
  - *default* - relations file for most experiments
  - *triple* - relations file for Protein experiments
  - *triple_na* - relations file for DNA experiments
  - *lcnmr* - relations file for LC-NMR
- <tshome>/conf/instr/<instrum>/prosol/<probeID>/<solvent>
  - *nucleus.channel.amplifier* - prosol parameters

OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters
  - *relations* - copy of the input relations file

SEE ALSO

edprosol [▷ 69], *eda [▷ 126]*

See also

📄 getprosol [▷ 131]

8 Pulse and AU Program Commands

This chapter describes commands for the setup of pulse programs, gradient programs, CPD programs etc. Furthermore, it describes the setup of various lists which can be used in acquisition like variable pulse or variable delay lists.

8.1 edcpul

NAME
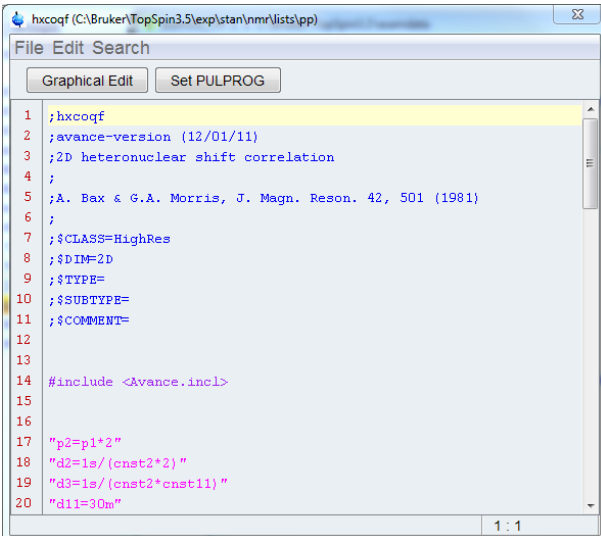
edcpul - Edit the current pulse program.

SYNTAX

edcpul [<name>]

DESCRIPTION

The command **edcpul** opens the TopSpin editor showing the current pulse program (see the next figure). The current pulse program is defined as the pulse program of the foreground dataset as defined by the acquisition parameter PULPROG.

Figure 8.1: edcpul

Bruker pulse programs are opened in **view** mode which means they cannot be modified. The dialog **File** menu offers you the following functions:

- **Open**

  This will open the **edpul** dialog where you can open any available pulse program.

- **Save as...**

  Allows you to save the pulse program under a different name

- **Print**

  This will print the pulse program

User defined pulse programs are opened in **edit** mode and can be modified. The dialog **File** menu offers you the following extra functions:

- **New**

  Allows you to open a new pulse program.

- **Save**

  Allows you to save the pulse program under the same name

Besides the File menu, the dialog offers the **edit** and **search** menu with common edit/search functions.

INPUT PARAMETERS

To be set with **eda** or by typing **pulprog** :

- PULPROG - the current pulse program (input of **edcpul**)

INPUT FILES

- <tshome>/exp/stan/nmr/lists/pp
  - Bruker and user defined pulse programs
- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters

OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters (output of **edcpul <name>**)

SEE ALSO

edgp [▷ 136], (edpul, edcpd)

8.2 edgp

NAME

edgp - Edit gradient programs

SYNTAX

edgp [<name>]

DESCRIPTION

The command **edgp** allows you to list, create or edit gradient programs. If you enter **edgp** without arguments, a list of all gradient programs is displayed (see the next figure).
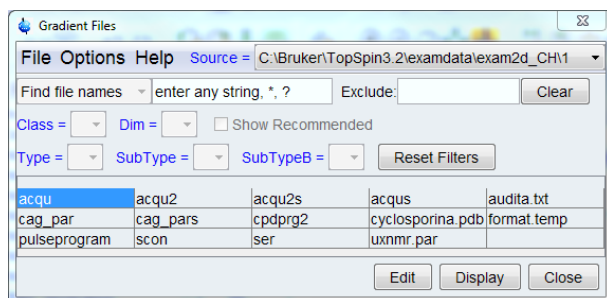
Figure 8.2: edgp

This shows a list of all Bruker and/or User gradient programs and allows you to edit or delete them.

To list Bruker gradient programs:

- Click **Options => Bruker defined**

To list User gradient programs:

- Click **Options => User defined**

To list all gradient programs:

- Click **Options => All defined**

To edit the selected gradient program:

- Double-click the gradient program or select the gradient program and click **Edit**. The gradient program will be shown in an editor and can be saved by clicking **Save**. To save it under a different name, click **Save as ...**

To create a new gradient program:

- Click **File --> New** and write the gradient program in the appearing editor

or

- Double-click an existing gradient program, modify the contents to your needs and store it under a different name by clicking **Save as ...**

To delete a gradient program

- Click **Delete** and click **OK** in the warning dialog.

If you enter the command with an argument, i.e. **edgp <name>**, the gradient program <name> is opened or, if it does not exist, it is created. The argument may contain wildcards, e.g. **edgp a*** displays a list of all gradient programs which start with a.

Bruker gradient programs must be installed with the command:

expinstall before they can be opened with **edgp**.

edgp uses the editor which is defined in the TopSpin User Preferences. To change it, enter **set**, click **Miscellaneous** and select or change the editor.

INPUT AND OUTPUT FiLES

- <tshome>/exp/stan/nmr/lists/gp/*
  - – gradient programs

SEE ALSO

edcpul [▷ 135], (edpu)l, *expinstall [▷ 76]*

8.3 edlist, dellist

NAME

edlist - Edit various lists.

dellist - Delete various lists.

SYNTAX

edlist [<type> <list>]

dellist [<type> <list>]

DESCRIPTION

The command **edlist** opens a dialog box with the **Edit...** option selected:
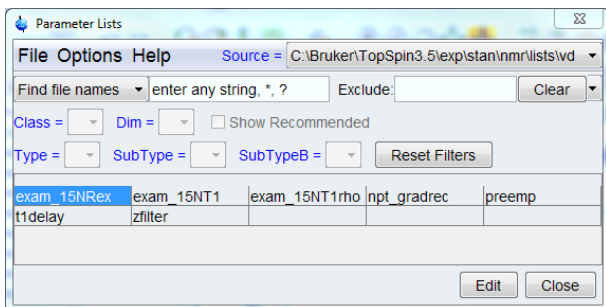
Figure 8.3: edlist

Just select the desired **List type** and click **OK**. The available files of the chosen list type appear and can be edited. To create a new list file:

- Open an existing file, edit it and click **Save as ...** to save it under a different name.

  or

- Enter a name in the field **New file**, click **OK**, enter the list in the appearing editor and click **OK**.

Lists types that can be edited with **edlist** are shown in the table *Table 8.1 [▷ 139]*. They are used by the acquisition, for example, a **vd** list is read by the pulse program statement `vd`.

A variable delay list can be set up and used as follows:

- Create a variable delay list with **edlist**.
- Set the acquisition parameter VDLIST to the name of that list.
- Specify the `vd` statement in the pulse program.

Other lists can be used in a similar way. They typically contain one value per line and the number of lines is unlimited. The type of the list that is interpreted is determined by the type of pulse program statement. The list name is determined by an acquisition parameter.

> **i** Note, however, that SREGLST is a processing parameter.

List type	Description	Defined by parameter	Read by pulse program statements
vd	delay list	VDLIST	`vd, ivd, vdidx`
vp	pulse list	VPLIST	`vp, ivp, vpidx`
va	amplitude list	VALIST	define list<power>
f1	frequency list (Avance)	FQLIST	`fq1, fq2, fq3` etc.
vt	temperature list	VTLIST	RVTLIST, VT, IVTLIST, DVTLIST
vc	counter list	VCLIST	`lo to` **x** `times c, ivc`
scl	scaling region list	SREGLST	
ds	dataset list	DSLIST	`wr #n, wr ##,` `ifp, dfp, rfp`
masr	MAS spin rate list		AU prog. multimas

Table 8.1: Lists used in acquisition

> **i** RVTLIST, VT, IVTLIST, DVTLIST:
>
> Note that these are AU program macros rather than pulse program statements.

Delay Lists

Delay lists contain any number of delay values with the corresponding units, e.g.:

> 10m
>
> 2s

where m = milliseconds and s = seconds. They are interpreted by the pulse program statements:

> `vd` - read value from the current position in the delay list
>
> `ivd` - increment the delay list position to the next value
>
> `"vdidx=n"` - set the index to position `n` in the delay list

Pulse Lists

Pulse lists contain any number of pulse length values with the corresponding units, e.g.:

> 10m
>
> 20m

where m = microseconds and m = milliseconds. They are interpreted by the pulse program statements:

> `vp` - read value from the current position in the pulse list
>
> `ivp` - increment the pulse list position to the next value
>
> `"vpidx=n"` - set the index to position `n` in the pulse list

A variable pulse list can only be used for hard pulses, not for shaped pulses.

Frequency Lists

Frequency lists contain the irradiation frequency preceded by the letter O on the first line (optional) and any number of frequency offset values on subsequent lines, e.g.:

O 500.13

3000

3150

3200

They are interpreted by pulse program statements like:

`30m fq1:f1`

Read the current value from the list defined by FQ1LIST to channel f1

`d1 fq2:f1`

Read the current value from the list defined by FQ2LIST to channel f1

`d11 fq3:f2`

Read the current value from the list defined by FQ3LIST to channel f2

> **i** Note that the `fq1`, `fq2` etc. statements must be specified with a delay. In contrast to the `vd` and `vp` statements, they automatically increment the current position in the list to the next value.

In the above example, the offset values in the list are added to the frequency specified on the first line, independent of the channel. If, however, the first line would be omitted, the offset values in the list are added to the values of SFO1, SFO2 etc. for the respective channels.

Counter Lists

A variable counter list contains any number of loop counter values, e.g.:

4

7

20

They are interpreted by the pulse program statement:

`lo to x times c`

Where `x` is a pulse program label and `c` is the value at the current position of the counter list

`ivc`

increment the counter list position to the next value

Variable Amplitude (power) Lists

A variable amplitude list contains any number of power values, e.g.:

-6.0

0.0

3.0

6.0

The entries represent attenuation values in dB.

The usage of a VA list is different from pulse and delays lists. You must define the statement by which a VA list is accessed in the pulse program. Such a statement can have any name, for example the name `vanam` is used in the examples below. The suffixes `.inc`, `.dex` and `.res` can be used to increment, decrement and reset the lists position, respectively. Furthermore, the caret operator (^) allows you to read a list value and increment the list position with one statement. The following pulse program entries illustrate the use of a variable amplitude list:

```
define list<power> vanam = <$VALIST>
```

definition of the power list

```
d1 vanam:f2 vanam.dec
```

set the power to the current value of the list and decrement the index

```
d1 vanam[2]:f3
```

set the power to the second value of the list

```
"vanam.idx = vanam.idx + 3"
```

increment the list index by 3

```
d1 vanam^:f4
```

set power to the current value of the list increment the index

As an alternative to using a list defined by the parameter VALIST, you can explicitly define a variable amplitude (power) list filename or even the list values in the pulse program. The following examples illustrates such definitions:

```
define list<power> vanam=<my_filename>
define list<power> vanam={10 30 50 70}
```

Note that the second definition does not require a list file.

Note that variable amplitude lists can only be accessed using define statements as described above. The statement va does not exist. More information on using variable amplitude lists can be found under:

Help ꞌ **Manuals** ꞌ [**Programming Manuals**] **Pulse Programming Manual**

Dataset Lists

A dataset list contains any number of dataset definitions, e.g.:

- *sucrose 1 1 C:\ guest new*
- *sucrose 2 1 C:\ guest new*
- *fructose 1 1 D:\ guest old*

where the option *new/old* is used to delete/keep a possibly existing dataset.

Caution: the usage of dataset names containing blanks (e.g. "sucrose low") in dataset lists, is not supported. The same counts for the username part of the dataset.

They are interpreted by the pulse program statements:

```
wr #n
```

Write the data to the dataset defined at the position `n` of the dataset list

```
wr ##
```

Write the data to the dataset defined at the current position of the dataset list

```
ifp
```

Increment the position in the dataset list

```
dfp
```
Decrement the position in the dataset list

```
rfp
```
Reset the position in the dataset list to the first entry

Temperature Lists

A variable temperature list contains any number of temperature values (in Kelvin), e.g.:

300

320

340

Temperature lists are interpreted by the AU program macros:

- RVTLIST

  Open the temperature list defined by VTLIST

- VT

  Read value from the current position in the temperature list and set the temperature unit accordingly.

- IVTLIST

  Increment the current position in the temperature list to the next value.

- DVTLIST

  Decrement the current position in the temperature list to the previous value.

> **i** Note that temperature lists are only interpreted by AU program macros, not by pulse program statements.

Masr Lists

MASR lists contain any number of MAS spin rate values, e.g.:

3000

3500

4000

4500

They are interpreted by the AU program **multimas**. This AU program will ask you if you want to use a list or enter the start and increment value interactively. If you choose the former (default), it will offer you the available MASR lists.

There are alternative ways of defining and using lists in acquisition. Rather than using a predefined statements like `vd`, you can define your own statement in the pulse program to access a certain list. For example, the statements:

```
define list<delay> vdnam = <$VDLIST>
```
```
define list<delay> vdfil = <my_file>
```
```
define list<delay> vdval = {0.1 0.2 0.3 0.4}
```

Allows you to use the statements `vdnam`, `vdfil` and `vdval` to read a value from a delay list. The use of lists via define statements is fully described above for the variable amplitude lists. More information on user defined lists can be found under:

Help ' Manuals ' [Programming Manuals] Pulse Programming Manual

- **edlist** can be used in one of the following ways:

- **edlist** - opens the edit/delete dialog box
- **edist <type>** - displays all entries of the list type <type>
- **dellist** opens the same dialog box as **edlist** expect that the Delete option is selected. When you select a **List Type**, and click **OK**, the available files of that type appear. You can click one or more entries to mark them for deletion. Clicking the **Delete** button deletes all marked entries.

INPUT AND OUTPUT FILES

- <tshome>/exp/stan/nmr/lists
  - *vd* - variable delay lists
  - *vp* - variable pulse lists
  - *va* - variable amplitude lists
  - *f1* - frequency lists
  - *vt* - variable temperature lists
  - *vc* - variable counter lists
  - *scl* - region files
  - *ds* - dataset lists
  - *scl* - scaling region files
  - *masr* - MASR rotation values

SEE ALSO

9   Acquisition Commands

This chapter describes all commands involved in data acquisition. This includes parameter optimization, the actual data acquisition and stopping a running experiment. Most experiments which involve a single acquisition are started with the command **zg** and run until they are finished. Other experiments involve multiple acquisitions executed with an AU program or with IconNMR.

9.1   aqguide

NAME

aqguide - Open the Acquisition Guide.

DESCRIPTION

The command **aqguide** opens the TopSpin Acquisition Guide (see the next figure). This contains a workflow for acquiring data, especially suited for new or occasional users. In **Automatic mode**, the Acquisition Guide will simply execute a acquisition command when you click the corresponding button. This requires the acquisition parameters to be set correctly. In interactive mode (**Automatic mode** unchecked), the Acquisition Guide will, at each step, open a dialog box offering you the available options and required parameters.

Experienced users normally enter the individual acquisition commands from the command line. This requires that, for each command, the acquisition parameters are properly set.
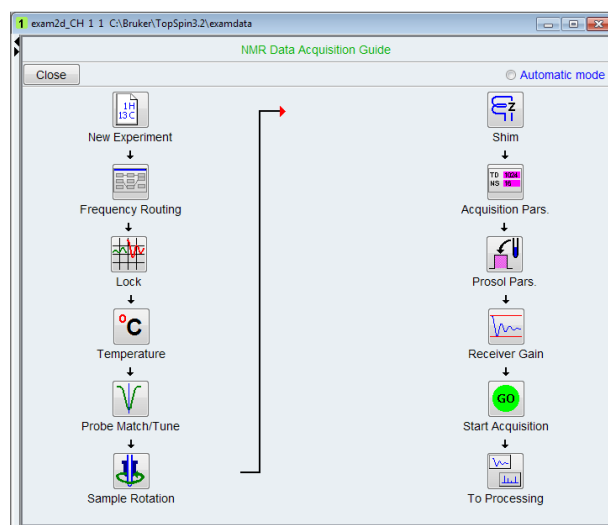
Figure 9.1: aqguide

SEE ALSO

(prguide, t1guide, solaguide)

9.2   bnmr

NAME

bnmr - Acquisition panel for basic and selective experiments.

DESCRIPTION

The command **bnmr** opens an icon box from which you can run basic or selective 1D and 2D NMR experiments (see the next figure). It is especially meant for new or occasional users. **bnmr** can be entered on the command line or started by clicking **Spectrometer => Basic/ Selective Experiments**.
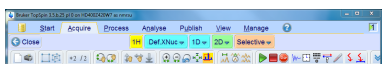
Figure 9.2: bnmr

At the top of the window, you find the buttons:

- **Close**: closes the **bnmr** window.
- **Selective**: switches to the **bnmr** window for selective experiments.
- **Tips**: allows you to switch on/off the tooltips.

The first row of icons (yellow) provide the following functions:

- **Lock**

  Allows you to select the solvent and then performs the auto lock-in (executes the TopSpin command **lock**).

- **Tune/Match**

  Switches to the acquisition menu and allows you to tune and match the probe (executes **xau buttonau match** which calls the TopSpin command **wobb**).

- **Auto Shim**

  Performs shimming according to the tune file defined for the current probe (executes the TopSpin command **tune .sx**). A probe dependent tune file can be defined from the **edprosol** dialog box.

The second and third row of icons (blue) provide 1D experiment, with or without decoupling.The fourth, fifth and sixth row of icons (green) provide 2D experiments, with or without gradients.

For a description of the individual experiments, check the **Tips** button and hold the cursor over the experiment icon to show the corresponding tooptip.

Clicking the button **Selective** switches the **bnmr** window to selective experiments (see the next figure).

Figure 9.3: Selective NMR Experiments

bnmr is an example of a button panel. You can create your own button panels with the command **bpan** (see TopSpin Users Guide).

INPUT FOLDERS

- <tshome>/exp/stan/nmr/par/
    - PROTON/* - routine proton experiment
    - C13CPD/* - routine 13C experiment
    - C13DEPT90/* - DEPT90 13C experiment
    - C13DEPT135/* - DEPT135 13C experiment
    - COSY90SW/* - COSY experiment
    - NOESYPHSW/* - NOESY experiment
    - MLEVPHSW/* - COSY experiment
    - COSYGPSW/* - gradient COSY experiment
    - INVBPHSW/* - HMQC experiment
    - INV4LPLRNDSW/* - HMBC experiment
    - INVIETGPSW/* - gradient HMQC experiment
    - INV4GPLPLRNDSW/* - gradient HMBC experiment
- <tshome>/prog/au/bin/

    Binary AU programs
    - *buttonau* - **bnmr** AU program
    - *butselau* - **bnmr** AU program for selective experiments
    - *butsel90* - **bnmr** AU program for calibrating the selective 90° pulse
- <tshome>/exp/stan/nmr/au/src/

    Source AU programs
    - *buttonau* - **bnnmr** AU program

- *butselau* - **bnnmr** AU program for selective experiments
- *butsel90* - **bnnmr** AU program for calibrating the selective 90° pulse
- <tshome>/classes/prop/English/
    - *cmdpanel_bnmr.prop* - **bnmr** properties file
    - *cmdpanel_bnmrsel.prop* - **bnmr** properties file for selective experiments

SEE ALSO

(bpan)

9.3　expt

NAME

expt - Display the experiment time

DESCRIPTION

The command **expt** calculates and displays the experiment time for the current data set (see the next figure).
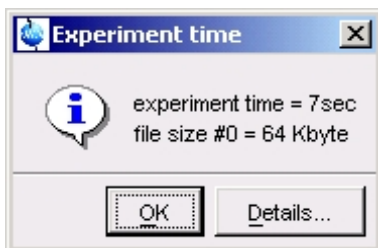
Figure 9.4: expt

If this exceeds the time you have available, you can reduce the number of scans (parameter NS) until the experiment time is acceptable.

For 2D and 3D experiments, **expt** also compares the file size of the raw data with the available disk space.

expt can also be executed by clicking the ⊡ button in the upper toolbar.

INPUT PARAMETERS

- NS - number of scans
- AQ - acquisition time in seconds

SEE ALSO

zg [▷ 165], *gs [▷ 149]*

9.4　go

NAME

go - Perform an acquisition, adding to existing data

DESCRIPTION

The command **go** starts an acquisition on the current dataset adding to possibly existing raw data. It works like **zg**, except that it does not overwrite existing data. If raw data already exist, **go** will add the new data to them. This is, for example, useful if the signal to noise of your spectrum is too low and you want to acquire additional scans. If no data exist, **go** has the same result as **zg**.

> i
>
> If you have stopped an acquisition with **halt** or **stop**, you can, in principle, continue it with **go**. Note, however, that the acquisition might have been stopped in the middle of a phase cycle and **go** starts a new phase cycle. Therefore, if you want to be able to stop and continue an acquisition, we recommend to use the commands **suspend** and **resume** (see the description of these commands).

> i
>
> Note that if you enter **halt** or **stop** during a 2D or 3D acquisition it might stop in the middle of a second or third direction increment (see above). However, this problem only occurs when you use the `wr` statement to write the data to disk. If you use the `mc` statement, the **go** command continues a 2D or 3D acquisition at the position where it was stopped. **Caution**: if you increment or decrement any pulses, delays or phases within the acquisition loop, you must do that within one of the `mc` arguments F1PH, F1QF etc., for example:
>
> ```
> d1 mc #1 to 1 F1PH(id0, ip1)
> ```

Most acquisitions are started with **zg** and run until they have finished. As such, the command **go** is not used very often. It is, however, used in some Bruker AU programs like **noediff**, **noemult**, **deptcyc** and **multicyc**.

Note the difference between the TopSpin command **go** and the pulse program statement `go` (See pulse programming manual).

INPUT AND OUTPUT PARAMETERS

See **zg**

INPUT AND OUTPUT FILES

See **zg**

USAGE IN AU PROGRAMS

- GO

SEE ALSO

zg [▷ 165], *suspend [▷ 160]*, *resume [▷ 158]*

9.5 gs

NAME

gs - Interactive parameter optimization during acquisition.

DESCRIPTION

The command **gs** opens a dialog box that consists of two panels. In the left panel, you can adjust parameters interactively during an acquisition (see the next figure). The right panel shows the FID display window where the fid is continuously updated showing the effect of each parameter change (see the figure *Figure 9.6 [▶ 151]*).

gs repeatedly executes the current pulse program but only up to the first `go=n` or `rcyc=n` statement. Therefore, **gs**:

- Does not accumulate data.
- Does not interpret the phase list.
- Does not write data to disk.

The dialog box only shows the parameters which are typically set during **gs** like irradiation frequencies and offsets, pulse lengths and power levels. By default, the irradiation frequency Offset is selected for adjustment. You can adjust the indicated parameter by putting the cursor on the slider, pressing the left mouse button and moving the mouse. Alternatively, you can click above or below the slider bar. The slider sensitivity can be changed from the **Sensitivity** field. The following parameters can be changed from the **gs** dialog box:

- SFO1 - SFO8 - irradiation frequency for channel f1- f8.
- O1 - O8 - irradiation frequency offset for channel f1- f8.
- PL[0-31] - square power levels.
- AMP[0-31] – amplitude.
- PHCOR[0-31] - reference phases.
- PH_ref - receiver phase correction.
- RG - receiver gain.
- P[0-31] - pulse lengths.
- D[0-31] – delays.

Frequency, Offset and Square Power appear for each channel that has been setup with **edasp**.
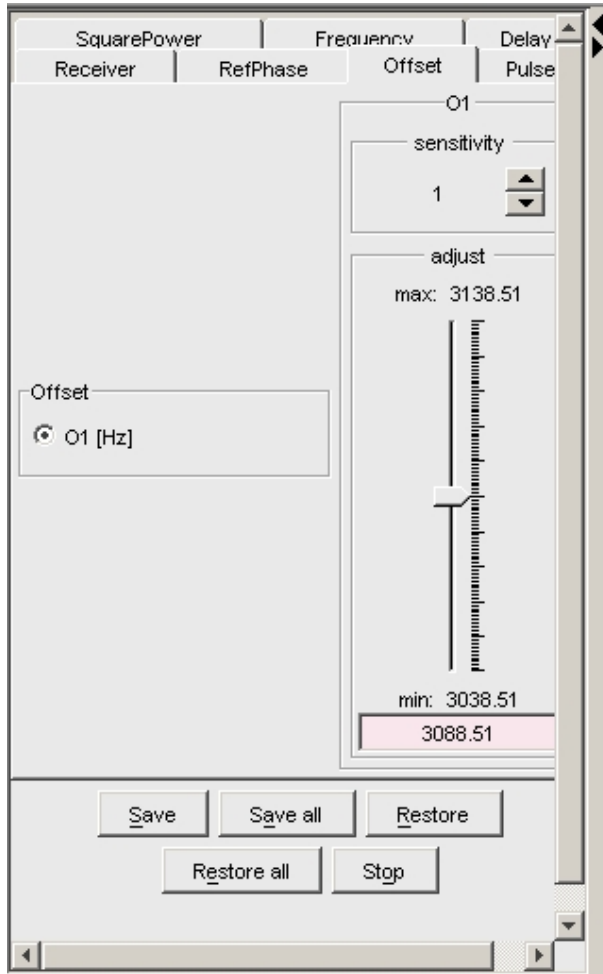
Figure 9.5: gs

ℹ Note that moving the RG slider causes an exponential change in the receiver gain. The effect of changing RG will be shown immediately in the FID display. For all other parameters in the above list, the effect of a change will be shown after one or two scans.
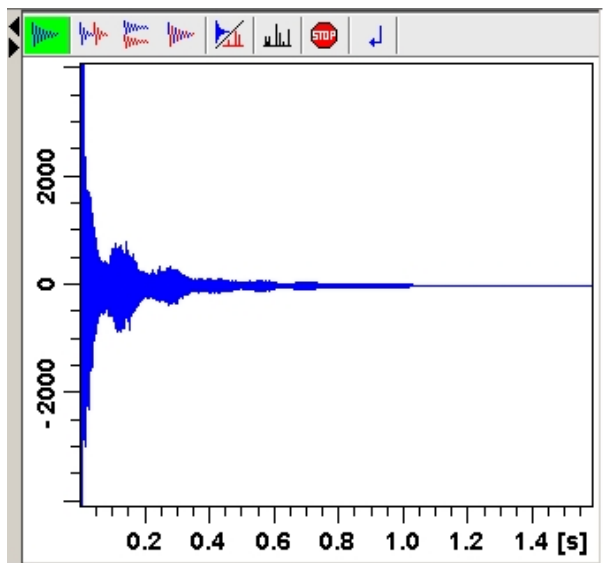
Figure 9.6: fid

The buttons of the FID window have the following functions:

► Show FID in shuffled mode.

Show FID in unshuffled mode, horizontally arranged.

Show FID in unshuffled mode, vertically arranged.

Show FID in unshuffled mode, interleaved.

Switch between FID and spectrum.

Stop the acquisition [**stop**].

Clicking the button to switch to real time FT, turns the button green and opens two extra buttons:

Switch between FID and spectrum.

Real time FT settings.

Toggle calculation of peak width at 50%, 5.5% and 1.1% height
(Shown as status parameters).

The GS dialog box also contains the following buttons:
- **Save** - save the parameter that was changed last.
- **Save all** - save all changes.
- **Restore** - restore the parameter that was changed last.
- **Restore all** - restore all changes.
- **Stop** - stop the acquisition and leave the **gs** dialog box.

If you click **Stop** without having saved your changes, **gs** shows you the changed parameters and allows you to select the ones to be saved.
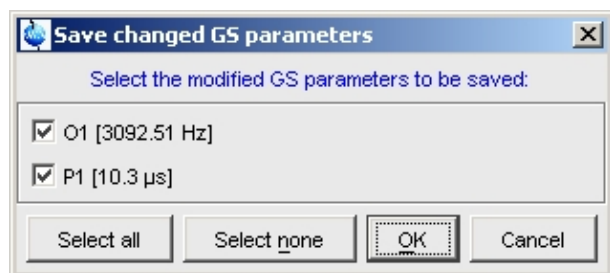
Figure 9.7: Save changed GS parameters

INPUT AND OUTPUT PARAMETERS

See the parameter list above.

INPUT AND OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters

9.6   halt

NAME

halt - Halt the acquisition after the current scan.

DESCRIPTION

The command **halt** stops the acquisition after the current scan and writes the data to disk. It is the regular command to stop a running acquisition.

> **i** Note that **halt** does not complete the current phase cycle.

As an alternative to entering **halt** on the command line, you can click the ■ button of the TopSpin toolbar or in the acquisition window toolbar. If, for some reason, a running or hanging acquisition cannot be stopped with **halt**, you can try the **stop** command. If that does not work either, you can use **kill**. This command will show all active processes and you can click the **go** module to stop the acquisition.

The command **halt** can even be entered with an optional parameter as command **halt <1...-NS>** (please note that <1...-NS> must be integer and > 0). If <1...-Ns> is defined the execution of **halt** is delayed until the total number of scans reaches a multiple of <1...-NS>. This option is useful if a fid with a complete phase cycle should be stored to disk.

As an alternative to typing **halt <1...-NS>** in the command line the button can be used from

the TopSpin toolbar with this bottom: ■

This button opens the following dialogue:

> **i** Please note that the command **halt** and its arguments does only work with DRU.
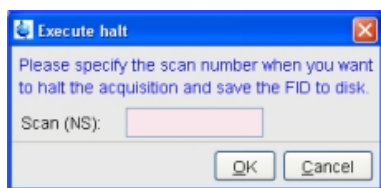
Figure 9.8: Execute halt

OUTPUT PARAMETERS

All acquisition parameters

OUTPUT FILES

In a 1D Dataset:
- <dir>/data/<user>/nmr/<name>/<expno>/
    – *fid* - raw data

– *acqus* - acquisition parameters

In a 2D Dataset:

• <dir>/data//nmr/<name>/<expno>/

– *ser* - raw data

– *acqus* - F2 acquisition status parameters

– *acqu2s* - F1 acquisition status parameters

In a 3D Dataset:

• <dir>/data//nmr/<name>/<expno>/

– *ser* - raw data

– *acqus* - F3 acquisition status parameters

– *acqu2s* - F2 acquisition status parameters

– *acqu3s* - F1 acquisition status parameters

SEE ALSO

stop [▷ 160], *zg [▷ 165]*, *go [▷ 148]*, *rga [▷ 159]*, *wobb [▷ 119]*

9.7 iconnmr

NAME

IconNMR - User interface for automation and routine spectroscopy.

DESCRIPTION

The command **iconnmr** starts the icon driven user interface IconNMR for routine spectroscopy and automation:

Figure 9.9: IconNMR

It is fully described in the IconNMR manual. The online version of this manual can opened by clicking **Help => Manuals =>** [**Automation and Plotting**] IconNMR **Automation Interface**.

9.8 ii

NAME

ii - Initialize spectrometer interface.

DESCRIPTION

The command **ii** initializes the spectrometer interface. It tries to access all hardware parts of the spectrometer that are needed for the current experiment and loads the acquisition parameters.

ii must be executed once, when the spectrometer has been switched off.

For AV III spectrometers it is possible to effectuate a restart of the hardware controllers with the command **ii restart**. This may be helpful if a hardware failure has blocked the system. This command also updates the program versions of the ipsoserver and the programs running on the hardware controllers according to the TopSpin version which is currently in use.

INPUT PARAMETERS

All acquisition parameters.

USAGE IN AU PROGRAMS

- II

SEE ALSO

zg [▶ 165]

9.9   popt

NAME

popt - Open the parameter optimization window.

DESCRIPTION

The command **popt** allows you to optimize acquisition parameters like pulses and delays (see the next figure).
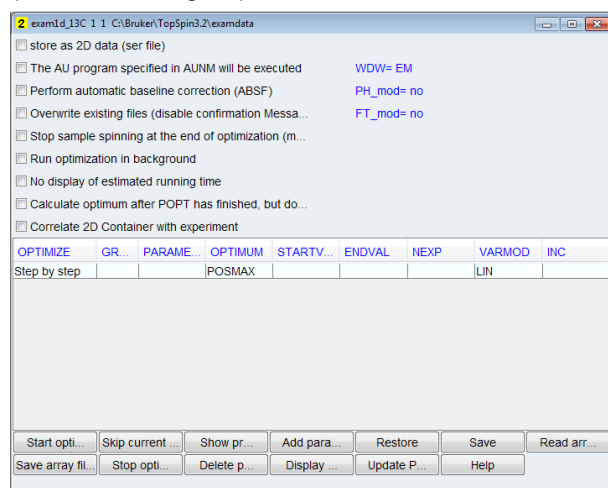
Figure 9.10: popt

Before you start an optimization, you must run one acquisition with the acquisition parameters as they are (not optimized) and process the data. On the resulting spectrum, you must define the spectral range (a peak or group of peaks) to be used for optimization. To do

that, click-hold the left mouse button at one edge of the desired region and drag the cursor to the other edge. The TopSpin display will automatically be adjusted to show the selected region only. You must now store this region by entering **dpl1**.

The **popt** dialog box allows you to create an entry for each parameter you want to optimize. By default, it shows only one entry; more parameters can be added by clicking the button **Add parameter**. Furthermore the following fields are offered:

- **OPTIMIZE** : Optimization of the parameters in the group.
    - **Step by Step**

        The parameters of a group are optimized one after the other. NEXP can be different for each parameter. The total number of experiments is the sum of the NEXP value of each parameter. The result of each parameter optimization is stored in a separate PROCNO.

    - **Simultaneous**

        The parameters of a group a optimized simultaneously. NEXP must be the same for each parameter and represents the total number of experiments. The result of the parameter optimization is stored in one PROCNO.

    - **Array**

        The parameters of a group a optimized according to an N-dimensional array (N is the number of parameters in a group). NEXP can be different for each parameter. The total number of experiments is the product of the NEXP value of each parameter. The result of the parameter optimization is stored in one PROCNO.

    - **No Optimization**

        The parameter is not optimized.

- **GROUP**: group number. Optimization starts with the lowest group number.
- **PARAMETER**: The parameter to be optimized. If **OPTIMIZE** is set to **No Optimization**, the parameter will be stored as comment in the optimization setup.
- **OPTIMUM**: Optimization criterion (see below).
- **STARTVAL**: First value of the parameter.
- **ENDVAL**: Last value of the parameter.
- **NEXP**: Number of experiments.
- **VARMOD**: Parameter variation mode: linear or logarithmic.
- **INC**: Parameter increment value. Unused for VARMOD = log.

The optimization criterion OPTIMUM can take the following values:
- POSMAX - the maximum value of a positive peak.
- NEGMAX - the maximum value of a negative peak.
- MAGMAX - the maximum magnitude value of a peak.
- MAGMIN - the minimum magnitude value of a peak.
- INTMAX - the maximum value of an integral.
- INTMIN - the minimum value of an integral.
- ZERO - zero intensity of a peak.

At the bottom of the dialog box you will find the following buttons:
- **Start optimize** - start the optimization for all checked parameters.
- **Skip current optimization** - stop the optimization for the current parameter.
- **Show protocol** - show the optimization result. A dialog appears with a list of protocol files:

- *popt.protocol* : the entire optimization result.
- *popt.protocol.999* : the optimization result in PROCNO 999.
- *popt.protocol.998*: the optimization result in PROCNO 998.
- etc.
- **Add parameter** - add a parameter entry.
- **Restore** - restore the last saved optimization setup.
- **Save** - save the current optimization setup to current dataset EXPNO.
- **Read array file** - read optimization setup.
- **Save array file as..** - save current optimization setup for general usage.
- **Stop optimization** - stop the optimization for all checked parameters.
- **Delete parameter** - delete the selected parameter.
- **Help -** open the **popt** help page.

Clicking the button **Start optimize** will start the optimization process.

> ℹ️ Note that it runs on the current dataset. For each parameter/group, a series of acquisitions will be performed. The result of this is a series of spectra (actually spectral regions) that are displayed in one screen and show the optimum parameter value. They are stored as one or more processed data files under the current dataset name and experiment number but under different processing numbers. For the first parameter/group that is optimized this is PROCNO 999, for the second parameter/group PROCNO 998 etc. As such, you must start **popt** on a dataset with PROCNO < 900. The result will also be stored in the so called protocol file (see OUTPUT FILES)

At the top of the dialog box you will find the following check buttons:

- **Store as 2D data (ser file)**.

  If checked, the result of the optimization (a series of 1D spectra) will be stored as a 2D dataset in EXPNO 899. However, if the source dataset PROCNO is greater than 100, the EXPNO of the destination 2D data will be

  PROCNO - 100.

- **The AU program specified in AUNM will be executed**.

  If checked, the AU program defined by AUNM will be executed instead of the command **zg**.

- **Perform automatic baseline correction (ABSF)**

  If checked, the output data will be baseline corrected with **absf**.

> ℹ️ Note that this command corrects the selected region (determined by the processing parameters ABSF1 and ABSF2).

- **Overwrite existing files (disable confirmation message)**.

  If checked, the output PROCNO, if it exists, will be overwritten without warning.

- **Run optimization in background**
- If checked, the foreground dataset will remain the same during the optimization. If it is not checked, the TopSpin display will change to PROCNO 999 where the optimization result is displayed.

To the right of the above check buttons, the current values of the parameters WDW (window multiplication mode), PH_mod (phase correction mode) and FT_mod (Fourier Transform mode) are displayed.

If you want to rerun an optimization, you must first return to the starting PROCNO.

INPUT FILES

- <tshome>/exp/stan/nmr/lists/popt/
  – *<name>* - parameter optimization setup for general usage (input of **Read array file..**)
- <dir>/data/<user>/nmr/<name>/<expno>/
  – *acqu* - acquisition parameters
  – *popt.array* - parameter optimization setup (input of **Restore**)
  – *popt.protocol* - parameter optimization result (input of **Show protocol**)
- <dir>/data/<user>/nmr/<name>/<expno>/pdata/<procno>/
  – *proc* - processing parameters
- <tshome>/prog/au/bin/
  – *poptau* - AU program that runs the optimization (executable)
  – *popthalt* - AU program that halts the current optimization (executable)

OUTPUT FILES

- <tshome>/exp/stan/nmr/lists/popt/
  – *<name>* - parameter optimization setup for general usage (output of **Save array file as..**)
- <dir>/data/<user>/nmr/<name>/<expno>/pdata/999
  – *1r* - processed data containing the optimization result of the first parameter/group
- <dir>/data/<user>/nmr/<name>/<expno>/pdata/998
  – *1r* - processed data containing the optimization result of the 2nd parameter/group
- <dir>/data/<user>/nmr/<name>/<expno>/
  – *popt.array* - parameter optimization setup (output of **Save**)
  – *popt.protocol* - parameter optimization result (output of **Start optimize**)
- <dir>/data/<user>/nmr/<name>/899/
  – *ser* - 2D raw data containing the optimization result

SEE ALSO

9.10   resume

NAME

resume - Resume a suspended acquisition.

DESCRIPTION

The command **resume** resumes an acquisition that has been suspended.

An acquisition can be suspended with the command **suspend**. When this is entered, the acquisition holds as soon as the pulse program statement `suspend` or `calcsuspend` is encountered. If the pulse program does not contain such a statement, **suspend** has no effect. Alternatively, an acquisition can be suspended with the pulse program statement `autosuspend` or `calcautosuspend`. They automatically hold the acquisition, without user interaction. The command **resume** continues acquisition that was suspended, either automatically or with the command **suspend**.

For more information on the `suspend` pulse program statements click:

Help => Manuals => [Programming Manuals] Pulse Programming Manual

A resumed acquisition does not start with dummy scans. This can be a problem if the recycle delay is shorter then 4 times the T1 value of the measured nucleus.

> **i** Note that the suspend information is temporarily stored on the spectrometer FCU, not on disk. As soon as you enter **stop** or **halt**, or switch of the spectrometer, the suspend information is lost and the acquisition cannot be resumed.

SEE ALSO

suspend [▷ 160], *zg [▷ 165]*, *go [▷ 148]*

9.11 rga

NAME

rga - Automatic receiver gain optimization.

DESCRIPTION

The command **rga** automatically optimizes the receiver gain. It performs an acquisition with varying receiver gain and finally sets this just below the value where no overflow occurs. In fact, **rga** repeatedly executes the current pulse program but only up to the first `go=n` or `rcyc=n` statement.

If you already know the proper value for the receiver gain, you can simply set RG in **eda** or by typing **rg** on the command line.

OUTPUT PARAMETERS

- can be viewed with **eda** or by typing **rg**:
  - RG - receiver gain

USAGE IN AUTOMATION

- RGA

SEE ALSO

gs [▷ 149], *zg [▷ 165]*, *go [▷ 148]*

9.12 stop

NAME

stop - Stop the acquisition immediately

DESCRIPTION

The command **stop** stops the acquisition immediately without writing the data to disk.

> ℹ️ Note that in most 1D experiments, the data are only written at the end of the acquisition and using **stop** would cause a complete loss of data. As such, it is used in case of emergency. To stop a normal acquisition the **halt** command is used.

Instead of entering **stop** on the command line, you can also click the ![stop button] button in TopSpin tool bar or in the acquisition window toolbar.

If, for some reason, a running or hanging acquisition cannot be stopped with **halt** or **stop**, you can use **kill**. This command will show all active processes and you can click the **go** module to stop the acquisition.

SEE ALSO

halt [▷ *153]*, *zg* [▷ *165]*, *go* [▷ *148]*, *rga* [▷ *159]*, *wobb* [▷ *119]*

9.13 suspend

NAME

suspend - Suspend a running acquisition.

DESCRIPTION

The command **suspend** allows you to hold a running acquisition. When it is entered, the acquisition holds as soon as the pulse program statement `suspend` or `calcsuspend` is encountered. If the pulse program does not contains such a statement, **suspend** has no effect. Alternatively, an acquisition can be suspended with the pulse program statement `autosuspend` or `calcautosuspend`. They automatically hold the acquisition, without user interaction. The command **resume** continues acquisition that was suspended, either automatically or with the command **suspend**.

> ℹ️ Note that the suspend information is temporarily stored on the spectrometer FCU, not on disk. As soon as you enter **stop** or **halt**, or switch of the spectrometer, this information is lost and the acquisition cannot be resumed.

Standard Bruker pulse programs do not contain any `suspend` statements. Therefore, **suspend** can only be used with user defined pulse programs which contain a `suspend` statement at a certain position.

SEE ALSO

resume [▷ *158]*, *zg* [▷ *165]*, *go* [▷ *148]*

9.14   topguide

NAME

> **topguide** - Open the *TopGuide* acquisition and processing tool

DESCRIPTION

> The command **topguide** opens a guided acquisition and processing tool: *TopGuide*. This is a new tool in TopSpin 2.0 for setting up and processing NMR experiments. As the name implies, it is an interactive guide, which guides the user through the necessary and important steps after the insert of a new sample (tuning & matching the sample, locking, shimming) and the following process of running a basic 1H 1D (reference) experiment. After these initial steps it offers a database with the basic 1D and 2D NMR experiments and helps the user to set those up.
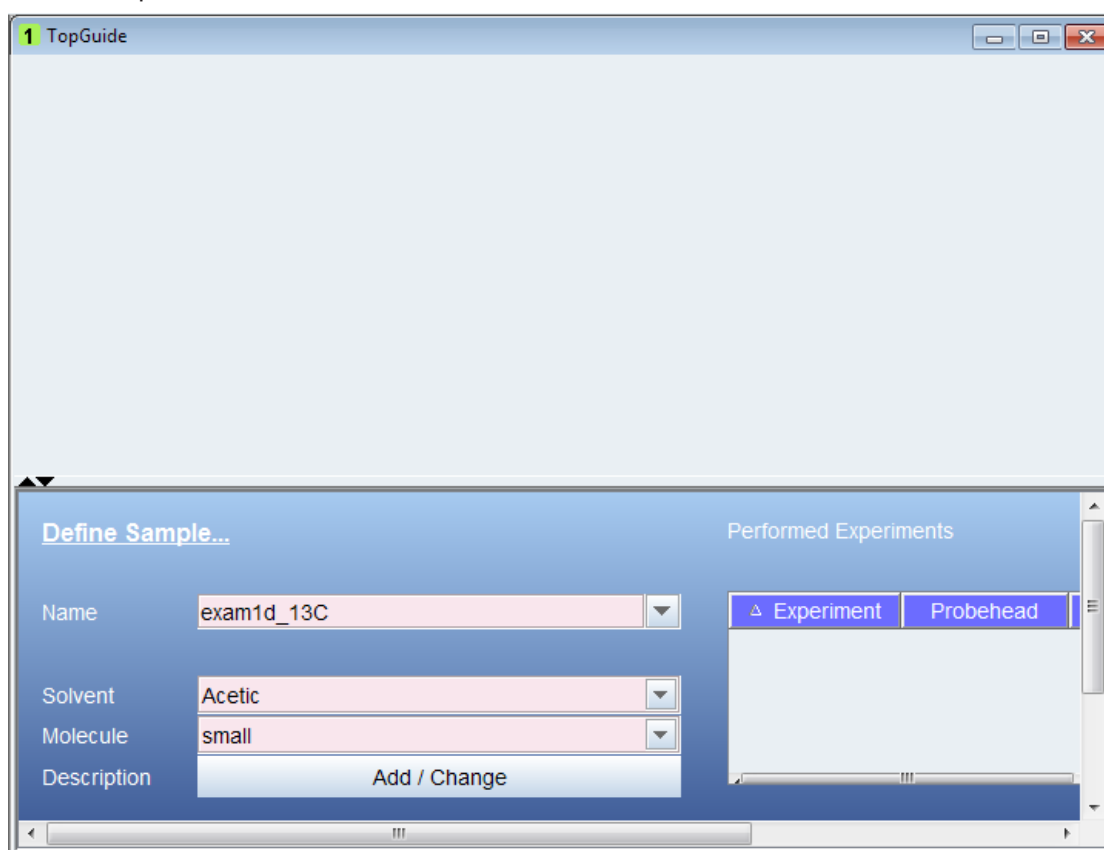
> *Figure 9.11: topguide*

> The special user interface of *TopGuide* guides the user through all the steps and decisions the user has to make. Each item within the user interface has its own help context. The help text describes the action the user has to take, gives explanations for parameters, provides explanations for procedures and/or some background information.

> *TopGuide* keeps track of the experiments which were run using *TopGuide* and stores this information in a database. For each sample (i.e. Name) the status window (Performed Experiments) shows the list of experiments which are setup (Acquisition no) or done (Acquisition yes) and if processing has been done or not (Processing yes/no). The status list provides an easy way to access these datasets: a simple left click will load the dataset into the upper part of the window, a right click pops up a context menu which gives access to either acquisition or processing parameters or some other dataset related commands.

On the right is a tool bar with 4 buttons for the basic operation. **GO** starts the acquisition on the current dataset. New in TopSpin 2.0 is a spooler mechanism (acquisition queue), which enables the user to run more than 1 experiment in a row. It is an option and the user should enable auto-queuing in the **Preference** dialog. If enabled all acquisition commands are automatically send into this queue. The **Queue** button offers the possibility to use the queue even if auto-queuing is disabled or to delay the start of an acquisition by entering a certain start time. The **Tools** button gives access to preferences and some other options and to leave *TopGuide* (**Quit**). **STOP** will stop the current acquisition and will disable the spooler. This is meant to be used as an emergency stop. If the user wants to end a running acquisition in a defined manner, he should type **halt** instead of **stop** (or **halt 8** on an Avance II system). Halt will finish the current scan and writes the acquired data to disk.

General Concept of *TopGuide*

If a new sample should be measured some initial set up steps are strongly advised: tuning & matching the probe to the sample susceptibility, locking and shimming. *TopGuide* will lead the user through these initial steps by either doing all this automatically or giving help information for manual operation. The automatic execution, of course, depends on the hardware configuration. A probe equipped with ATMA and a Z-gradient coil allows this. Without this hardware, some manual steps are necessary and the user is guided through manual tuning & matching and/or manual shimming. If a Z-gradient is present, *TopGuide* assumes that automatic gradient shimming is set up and it will be used by *TopGuide*. *TopGuide* caches the information about these steps executed, so further measurements (within the same *TopGuide* session) on the same sample will not ask for this again, unless the user changes the nucleus to be measured. If, for example, a 1H,13C-HSQC had been measured and therefore *TopGuide* knows that the probe is tuned to 1H and 13C, and you now want to run a 1H,15N-HSQC (let's assume you have a BBI probe and therefore this is possible), *TopGuide* will request to tune & match again, now to 15N instead of 13C.

The use of *TopGuide* to setup new experiments is based on a recorded 1H 1D spectrum. This proton spectrum is used to determine the signal to noise ratio (S/N) for this sample. *TopGuide* will select the highest signal in the spectrum (solvent peaks are automatically excluded), but the user may select a different peak/area for the S/N calculation. This may be of interest if the highest signal doesn't belong to the investigated compound in the case of mixtures or impurities. The determined S/N is used to estimate the number of scans (NS) for further experiments, to obtain a spectrum with reasonable S/N.

TopGuide provides a database with basic 1D and 2D experiments. *TopGuide* will only offer those experiments, which may be run on the current hardware, i.e. if the current probe doesn't have a gradient, *TopGuide* will not offer any experiment using gradients, but if the probe is equipped with a gradient, *TopGuide* will preferably offer experiments using gradients. If the current probe is a 13C/1H dual probe, *TopGuide* will not offer any 15N experiment. There are some more filter criteria based on the field strength. *TopGuide* doesn't use parameter sets like IconNMR. Therefore *TopGuide* will never show the same experiments as you would see in the listing of parameter sets, it will always show recommended experiments.

Before an additional experiment is performed, *TopGuide* will ask you to determine the spectral width (SW) to be used in the proton dimension. It will pre-select the region with peaks found in the reference spectrum. You may change this to your needs, but be aware that for homonuclear 2D measurements this SW will be applied to the indirect dimension as well. If you select a region which cuts off some signals, you may get folding of these peaks in the indirect dimension of a homonuclear 2D. The information about your selected SW is stored together with the reference spectrum and will be used in all further 1H dimensions for this sample.

The database contains reasonable acquisition and processing parameters for all experiments in it. Some parameters are fixed (typical coupling constants), some are dynamic and are chosen based on the molecular size (like relaxation times or NOESY mixing times), the field strength (TD) or the actual experiment (different SW for 13C in HSQC and HMBC).

Tools Button

The **Tools** button gives access to some preferences and options. Data Directory: the user can select the data directory where the NMR data is stored (disk unit). The initial setting depends on how *TopGuide* has been started. If you start *TopGuide* on a loaded dataset, it will keep that disk unit, if you start with no active dataset this will open a new windows and the data path defaults to the TopSpin installation directory (i.e. C:\Bruker\TopSpin under Windows or /opt/topspin under Linux).

User: if additional IconNMR users are defined, you can login as one of those.

Preferences: you can enable or disable 'auto-processing'. Depending on this setting newly acquired data is automatically processed with default processing parameters and options, or not.

Protected acquisition: this means that once an FID or SER file has been acquired it is protected by *TopGuide*, you are not allowed to overwrite it. The start of a new acquisition on a dataset with an existing FID/SER file will automatically create a new dataset (i.e. expno).

Protected processing: the same protection as described above is applied to processed data. Existing ones are not overwritten, but instead a new procno is created.

Import reference 1D: Setting up experiments with *TopGuide* requires a reference 1D spectrum, which is measured with *TopGuide* by default. If a 1D spectrum has already been recorded outside of *TopGuide* and you want to use this one as a reference for further experiments, you can import it into *TopGuide*. The import searches all expnos of the current dataset and displays the available 1D 1H spectra.

SEE ALSO

zg [▷ 165]

9.15 tr

NAME

tr - Transfer data to disk during the acquisition.

DESCRIPTION

The command **tr** transfers (writes) data to disk during a 1D acquisition. This is, for example, useful if you want to do a Fourier transform and view the spectrum before the acquisition has finished. Another reason to use **tr** is to save the currently acquired scans of a long term acquisition. This avoids losing all data in case of a power loss.

As an alternative to entering **halt** on the command line, you can click the ■ button of the TopSpin toolbar.

The command **tr** can even be entered with an optional parameter as command **tr <1...-NS>** (please note that <1...-NS> must be integer and > 0). If <1...-NS> is defined the execution of **tr** is delyed until the total number of scans reaches a multiple of <1...-NS>. This option is useful if a fid with a complete phase cycle should be stored to disk.

As an alternative to typing **tr <1...-NS> in the command line** the button can be used from

the TopSpin toolbar with this botton:

This button opens the following dialog:
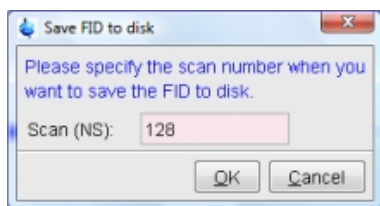
Figure 9.12: Save FID to disk

> ℹ️ Please note that the command **tr** and its arguments can only be used with DRU, not with RCU.

OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *fid* - 1D raw data
  - *acqus* - acquisition status parameters

SEE ALSO

9.16   xaua, xaup

NAME

xaua - Execute the AU program specified with AUNM

xaup - Execute the AU program specified with AUNMP

DESCRIPTION

The commands **xaua** and **xaup** execute the AU program which is specified by the parameters AUNM and AUNMP, respectively. In all Bruker parameter sets, these parameters are set to relevant Bruker AU programs. For example, in the parameter set PROTON, AUNM = au_zg and AUNMP = proc_1d. When parameter sets are used in automation (IconNMR), the AU programs specified by AUNM and AUNMP perform the acquisition and the processing, respectively.

For details on writing, compiling, and executing AU programs please refer to the TopSpin AU reference manual (click **Help** => **Manuals** => [**Automation**] **AU Programming**).

INPUT PARAMETERS

- Set by the user with **eda** or by typing **aunm** :
  - AUNM - acquisition AU program name for **xaua**
- Set by the user with **edp** or by typing **aunmp** :
  - AUNMP - processing AU program name for **xaup**

INPUT FILES

- <tshome>/exp/stan/nmr/au/src/
  - AU program source files (only input if the AU program is not compiled yet)
- <tshome>/prog/au/bin/

- – AU program binary executables
- <dir>/data/<user>/<name>/nmr/<expno>/
    - – *acqu* - acquisition parameters (input file for **xaua**)
- <dir>/data/<user>/<name>/nmr/<expno>/pdata/<procno>/
    - – *proc* - processing parameters (input file for **xaup**)

USAGE IN AU PROGRAMS

- XAUA
- XAUP
- XAUPW
XAUPW waits until the AU program has finished before the next statement is executed whereas XAUP doesn't. XAUA works like XAUPW is this respect.

SEE ALSO

(xau, edau, delau), *expinstall [▷ 76]*, (compileall, cplbruk, cpluser)

9.17 zg

NAME

zg - Perform an acquisition, overwriting possibly existing data

DESCRIPTION

The command **zg** performs an acquisition on the current dataset. A new data window will appear showing the currently acquired FID:
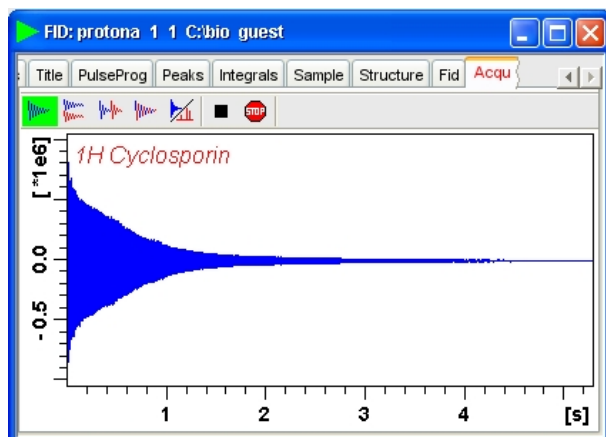
Figure 9.13: zg

Furthermore, the data window selection bar is extended as follows:

The buttons of the FID window have the following functions:

 Show FID in shuffled mode.

 Show FID in unshuffled mode, horizontally arranged.

 Show FID in unshuffled mode, vertically arranged.

Show FID in unshuffled mode, interleaved.

Switch between FID and spectrum.

Halt the acquisition [**halt**].

Stop the acquisition [**stop**].

Clicking the button to switch to real time FT, turns the button green and opens an extra button to set real time FT settings:

Switch between FID and spectrum.

Real time FT settings.

Toggle calculation of peak with at 50%, 5.5% and 1.1% height.

(Shown as status parameters)

Clicking this button opens the following dialog window:
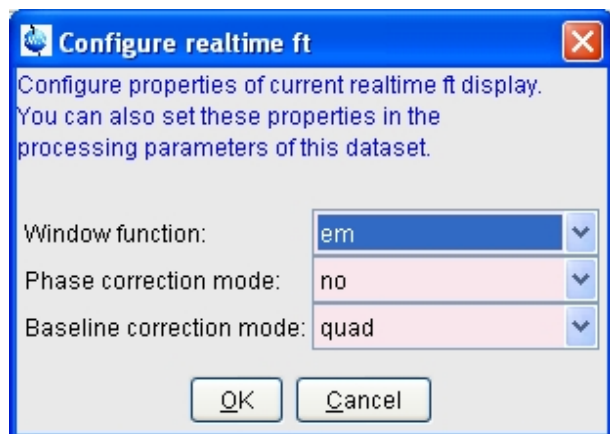
Figure 9.14: Configure realtime ft

The command **zg** performs the following sequence of actions:

- Read the acquisition parameters.
- Compile the pulse program defined by the acquisition parameter PULPROG.
- Execute the pulse program. For most pulse programs this involves:
    - The acquisition of DS dummy scans.
    - The acquisition of NS real scans, accumulating the data.
    - Storing the accumulated data to disk.
- Update the acquisition status parameters

Acquisition Sequence

Before you can start an acquisition with **zg**, you will normally have to prepare the experiment. This typically involves the following command sequence:

- **edhead** - define the current probe.
- **edprosol** - define the probe and solvent dependent parameters.
- **new** - define a new dataset and make it the current dataset.

- **wobb** or **atma** - tune and match the probe (**atma** only for ATM probes).
- **lockdisp** - open the lock display window.
- **lock** - lock the magnetic field.
- **rsh** or **gradshim** - read shim values or use gradient shimming.
- **rpar** - read a standard parameter set for the desired experiment.
- **getprosol** - get the probe and solvent dependent parameters.
- **edasp** - set up nuclei (NUC1, NUC2 etc.) and the spectrometer routing
- **ased** or **eda** - adjust the acquisition parameters to your needs.
- **zg** - acquire the data.

Pulse Program

The type of experiment is mainly determined by the pulse program. A simple 1D Bruker pulse program is `zg`, which contains the following lines:

```
1 ze                    ;zero memory, reset scan counter and phase lists,
                        ; enable dummy scans, label 1
2 d1                    ;relaxation delay with length D1, label 2
  p1 ph1                ;pulse with length P1 and phase ph1
  go=2 ph31             ;sample TD points with phase ph31, loop to 2
                        ; times NS+DS
  wr #0                 ;write the data to the current dataset
exit                    ;end of the pulse program
ph1=0 2 2 0 1 3 3 1 ;phase program (list) used for the RF pulse
ph31=0 2 2 0 1 3 3 1 ;phase program (list) used for the receiver
```

The RF pulse `p1` is executed on the default channel (f1) with the default power level (PL1). The loop to label 2 will be performed NS times to accumulate this number of scans. The `wr` statement lies outside of this loop which means the data are only written to disk once, at the end of the experiment.

Furthermore, an information window will appear showing the current scan, the total number of scans and the residual experiment time.

The purpose of dummy scans is to reach steady state conditions concerning T1 relaxation. This is necessary whenever the recycle delay of the experiment is shorter then 4 times the T1 value of the measured nucleus. Furthermore, dummy scans can be used to establish a stable temperature. This is especially important in decoupling experiments where the irradiation high power increases the sample temperature. The number of dummy scans is determined by the acquisition parameter DS. In the pulse program, the `ze` statement explicitly enables the `go` statement to perform dummy scans. In contrast, the `zd` statement disables the execution of dummy scans.

Transfer to Disk

In most 1D experiments, the acquired data are written to disk only once, at the end of the experiment. You can, however, use the command **tr** to write data to disk while the acquisition is running. This is, for example, useful if you want to do a Fourier transform and view the spectrum after a few scans. Another reason to use **tr** is to save the currently acquired scans of a long term experiment. This avoids losing all data in case of a power loss. In multidimensional experiments, the `wr` statement is part of a loop, and as such, is executed for each increment in the indirect direction(s).

> **i** Note that in many pulse programs the `wr` statement has been replaced by the `mc` statement which performs both the disk write and loop back function.

Multidimensional Data

zg allows you to perform 1D or multi dimensional acquisition. The dimensionality of the data is determined by the pulse program. If this contains an `if` and/or `st` statement, **zg** assumes a 2D or 3D dataset. Furthermore, if the pulse program contains a nested loop with loop counters `td1` and `td2`, **zg** assumes a 3D dataset. Finally, the acquisition parameter PARMODE is evaluated. If the value of PARMODE is not consistent with the pulse program, **zg** will display a warning but allow you to continue the acquisition. 1D data are stored a file named *fid* under the experiment number (expno) of the current dataset. 2D and 3D data are stored in a file named *ser* in the same directory. The pulse program is not limited to 3D and allows you to acquire data in 4 or more directions. TopSpin 2.0 and newer supports processing of nD data. Datasets up to dimension 5 have been tested by Bruker.

Stopping Acquisition

Normally, an acquisition will run until it has finished. If, however, you want to interrupt it, you can do that by clicking the ■ button or entering the command **halt**. This will finish the current scan, write the data to disk and then stop the acquisition.

> **i** Note that **halt** finishes the current scan but not the current phase cycle.

Clicking the 🛑 button or entering the command **stop** immediately stops the acquisition. It does not finish the current scan or write the data to disk. If, for some reason, **halt** and **stop** do not work, you can use the command **kill** to stop the acquisition.

When an acquisition has finished but the signal to noise is insufficient, you can accumulate additional scans. Just set the parameter NS to the number of extra scans and enter **go**. This command does not overwrite the existing data but adds the extra scans to them. **Caution**: **go** works correctly if the previous acquisition has finished normally but should not be used if this has been interrupted with **halt** or **stop** (see the description of **go**).

Preventing Data Overflow

In TopSpin 2.0 and newer, 1D raw data are, if necessary, stored as double precision (64-bit) data. If, based on the number of scans and the current accumulated signal, data overflow is expected to occur, data are transferred to the computer that controls the spectrometer, i.e. an automatic **tr** command is executed. Here the data are stored in 64-bit double precision format. The acquisition status parameter DTYPA shows whether data are stored as integers (DTYPA = 0) or doubles (DTYPA = 2). **Caution**: raw data with DTYPA 2 can only be processed with TopSpin 2.0 or newer.

> **i** Note that processed data are always stored as integer data.

Preventing Data Overwrite

zg overwrites possibly existing raw data. Each user can configure TopSpin to get a warning about existing data or not. Click **Preferences => Options**, click **Acquisition** and uncheck the entry "**Overwrite existing FID without inquiry**".

> ⓘ Note the difference between **zg** and **go**. The latter command does not overwrite existing data but adds new data to them.

After a manually performed acquisition has finished, it is normally processed with processing commands like **em**, **ft**, **apk** etc. Processing commands are described in the Processing Reference Manual.

The command **zg** is automatically executed by AU programs that contain the ZG macro. Examples of these are **au_zg***, **au_getl*** and **multizg**. Furthermore, **zg** is automatically executed when experiments are started from the automation interfaces IconNMR and **bnmr**.

> ⓘ Note that an acquisition always runs in background which means that TopSpin can do other tasks simultaneously. You can change the focus to a different data window and process data there. This is actually what happens all the time during an automation sequence performed by IconNMR.

Pulse Programming

TopSpin is supplied with a large number of pulse programs from simple 1D to sophisticated multidimensional experiments. Furthermore, you can write your own pulse programs with the **edpul** command using the Bruker pulse programs as an example. For a detailed description of the pulse program syntax click:

Help => Manuals => [**Programming Manuals**] **Pulse Programming Manual**

In TopSpin 2.0 and newer, Python programs and pulse programs can be integrated. Python programs can calculate pulse program parameters such as delays, pulses, phases and shapes using Python, and generate or execute a pulse program by executing a Python program.

(**cf . Help => Manuals =>** [**Programming Manuals**] Pulse Programming With Python).

Command Spooling/Queuing

zg, like other acquisition commands, are automatically queued.This allows you to enter multiple **zg** commands, for example on different datasets. Queued commands can be viewed in the Spooling field of the acquisition status bar. Automatic command spooling as well as the display of the spooling field in the status bar can be enabled/disabled in the User Preferences window (command **set**).

INPUT PARAMETERS

All acquisition parameters as described in chapter *TopSpin Parameters [▷ 11]* of the Acquisition Reference Guide.

OUTPUT PARAMETERS

All acquisition status parameters as described in chapter *TopSpin Parameters [▷ 11]* of the Acquisition Reference Guide.

INPUT FILES

1D Data

- \<dir\>/data/\<user\>/nmr/\<name\>/\<expno\>/
  - *acqu* - acquisition parameters

2D Data

- \<dir\>/data/\<user\>/nmr/\<name\>/\<expno\>/
  - *acqu* - acquisition parameters direct direction (F2)
  - *acqu2* - acquisition parameters indirect direction (F1)

3D Data

- \<dir\>/data/\<user\>/nmr/\<name\>/\<expno\>/
  - *acqu* - acquisition parameters direct direction (F3)
  - *acqu2* - acquisition parameters indirect direction (F2)
  - *acqu3* - acquisition parameters indirect direction (F1)

All Data

- \<tshome\>/exp/stan/nmr/lists/pp
  - \<PULPROG\> - the pulse program defined by PULPROG
- \<tshome\>/conf/instr/\<curinst\>/
  - *scon* - spectrometer parameters (created with **edscon**)

OUTPUT FILES

1D Data

- \<dir\>/data/\<user\>/nmr/\<name\>/\<expno\>/
  - *fid* - raw data
  - *acqus* - acquisition status parameters

2D Data

- \<dir\>/data/\<user\>/nmr/\<name\>/\<expno\>/
  - *ser* - raw data
  - *acqus* - acquisition status parameters direct direction (F2)
  - *acqu2s* - acquisition status parameters indirect direction (F1)

3D Data

- \<dir\>/data/\<user\>/nmr/\<name\>/\<expno\>/
  - *ser* - raw data
  - *acqus* - acquisition status parameters direct direction (F3)
  - *acqu2s* - acquisition status parameters indirect direction (F2)
  - *acqu3s* - acquisition status parameters indirect direction (F1)

All Data

- \<dir\>/data/\<user\>/nmr/\<name\>/\<expno\>/
  - *pulseprogram* - the precompiled pulse program
  - *audita.txt* - acquisition audit trail

- *scon* - spectrometer parameters
- *cpdprg1* - CPD program (output if a `cpd1:f2` statement is used)
- *cpdprg2* - CPD program (output if a `cpd2:f2` statement is used) etc.
- *fq1list* - variable frequency list (output if a `fq1` statement is used)
- *fq2list* - variable frequency list (output if a `fq2` statement is used) ect.
- *spnam1* - shaped pulse definition (output if a `sp1` statement is used)
- *spnam2* - shaped pulse definition (output if a `sp2` statement is used) ect.
- *gpnam1* - shaped gradient definition (output if a `gp1` statement is used)
- *gpnam2* - shaped gradient definition (output if a `gp2` statement is used) ect.
- *vclist* - variable counter list (output if a `vc` statement is used)
- *vdlist* - variable delay list (output if a `vd` statement is used)
- *vplist* - variable pulse list (output if a `vp` statement is used)
- *dslist* - dataset list (output if a `wr #n` or `wr ##` statement is used)
- *cag_par* - rotation matrices for gradients used for imaging
- *grdprog* - shaped gradient definition (output if a `ngrad` statement is used)

USAGE IN AU PROGRAMS

- ZG

SEE ALSO

go [▷ 148], gs [▷ 149], rga [▷ 159], wobb [▷ 119], atma [▷ 109], suspend [▷ 160], resume [▷ 158]

Acquisition Commands

10 Temperature Commands

This chapter describes commands which configure, control or monitor the temperature unit. Temperature units which are delivered with Bruker spectrometers are often referred to as a eurotherm; the make of most units. They are used to warm up or cool down the NMR sample and keep it at a certain temperature during the experiment.

10.1 edte

NAME

edte - Edit the temperature parameters

DESCRIPTION

The command **edte** opens a dialog box from which the temperature unit can be controlled (see the next figure).
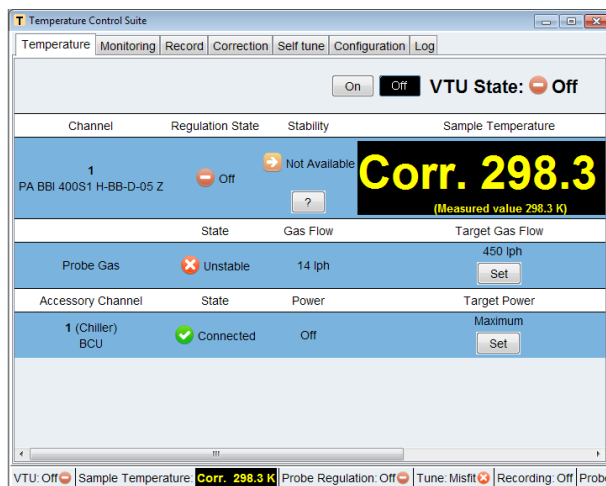
Figure 10.1: edte

The functionality of this window is described in a separate manual that can be opened from the TopSpin window by clicking:

Help => **Manuals** => [**Technical Manuals**] **Temperature Regulation**

Or from the **edte** window by clicking:

Help => **Edte Users Manual**

SEE ALSO

teget, te2get [▷ 173], *teset, te2set [▷ 174]*

10.2 teget, te2get

NAME

teget - Get the temperature from the temperature unit.

te2get - Get the second temperature from the temperature unit.

DESCRIPTION

The command **teget** gets the temperature from the temperature unit and stores it in the acquisition status parameter TE.

The command **te2get** gets the so called second temperature from a temperature unit with two regulators. The value is stored in the acquisition status parameter TE2. Temperature units with two regulators are, for example, used in BEST NMR where the first regulator controls the sample temperature and the second regulator controls the inlet capillary temperature.

OUTPUT PARAMETERS

- *TE* - demand temperate on the temperature unit.
- *TE2* - demand second temperate on the temperature unit.

OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqus* - acquisition status parameters

USAGE IN AU PROGRAMS

- TEGET
- TE2GET

SEE ALSO

edte [▷ 173], teset, te2set [▷ 174]

10.3   teset, te2set

NAME

teset - Set the temperature on the temperature unit.

te2set - Set the second temperature on the temperature unit.

SYNTAX

teset [<temperature>]

te2set [<temperature>]

DESCRIPTION

The command **teset** sets the temperature on the temperature unit. It takes one argument and can be used as follows:

teset

Sets the temperature to the value of the acquisition parameter TE. Before you enter **teset** without argument you must set TE to the desired temperature, in Kelvin, either from **eda** or by typing **te** on the command line.

teset <temperature>

Sets the temperature to the specified value.

The command **teset** is, for example, used in the AU programs **au_zgte** and **multi_zgvt**.

The command **te2set** works like **teset** except that it sets the so called second temperature to the value of the acquisition parameter TE2. This value is set on the second regulator of a temperature unit with two regulators. Such units are, for example, used in BEST NMR where the first regulator controls the sample temperature and the second regulator controls the inlet capillary temperature.

INPUT PARAMETERS

- TE - demand temperature on the temperature unit (input of **teset**).
- TE2 - demand second temperature on the temperature unit (input of **te2set**).

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters

USAGE IN AU PROGRAMS

- TESET
- TE2SET
- VT executes the command **teset <vt_list_index>**

SEE ALSO

edte [▷ 173], *teget, te2get [▷ 173]*

Temperature Commands

11 MAS Commands

11.1 MasDisp

11.1.1 Accessing the User Interface

The user interface can be accessed in different ways:

- Enter **masdisp** at the TopSpin prompt.
- Click **mas** in the TopSpin **Acquisition** tab.
- Click the **MAS spin rate** icon on the TopSpin acquisition status bar at the bottom of the TopSpin window.

11.1.2 Masdisp Overview

The user interface supplies the following tabs:

- **Main**: Main window for routine operation.
- **Monitoring**: Display of spin rate and pressure settings or readings.
- **Recording**: Writing and (graphical) inspection of record files.
- **Configuration**: Settings for displaying stability information and for window handling.
- **Log**: Display of MAS unit notifications and error messages.
- **Help**: Display of information about setting up and using the MAS unit.

A status line is located at the bottom of the window. It displays the current state (rotation on/off), the spinning stability, the current spinning speed, the system pressure, and the probe temperature as monitored and controlled by the variable temperature control unit.

11.1.3 The Main Tab of the TopSpin User Interface

11.1.3.1 Overview

Most of the MAS operation is controlled from the **Main** menu. It shows the following information (see the figure in section *Temperature Limits and Spin Rate Limits [▷ 178]*) from top to down:

- Current and desired spin rate.
- Buttons for insert, eject, spinning, and stopping the rotor.
- Supply pressure and internal operation pressure readings of the MAS unit.
- Actual pressure settings for bearing, sense of bearing, and drive.
- Frame cooling state and pressure setting.
- Settings for probe, profile, rotor type, cap material, temperature limits, and spin rate limits.

At startup the following settings are displayed: probe, profile (this defines the regulation parameters and others), rotor type, cap material, temperature limits, and lower and upper spinning frequency limits. Except for the cap material usually no user input is required because the probe is set according to the *current probe* as defined by the **edprobe**

command. The profile and rotor type are automatically chosen according to the probe. Temperature limits and frequency limits are defined by the cap material, the profile and the current probe, respectively and cannot be changed independently.

11.1.3.2 Probe and Profile Selection

For standard operation the user does not need to select the probe and corresponding profile. At startup the main tab will show the current probe as set by **edprobe**. A suitable profile (this defines the regulation parameters as well as maximum and minimum spinning frequencies) is automatically loaded. The user may change the probe by clicking on the **Exchange probe** button – the NMR super-user password is required for this. The probe cannot be changed while a rotor is spinning. The profile may be changed as well by choosing one of the available profiles from a drop-down list. Usually, the default profile will be the correct one, special profiles may be set for dedicated applications like slow spinning. Details of the current profile may be inspected by clicking the **Show profile** button.

11.1.3.3 Rotor and Cap Type Selection

The rotor is set according to the selected probe, this cannot be changed. The cap type may be selected from a drop-down menu, the default setting is *Kel-F* for 4 mm and 7 mm rotors and Vespel for smaller diameter ones. The cap material defines the allowed temperature range and maximum spinning speed.

11.1.3.4 Temperature Limits and Spin Rate Limits

The displayed temperature limits are set according to the probe and cap material settings. The display is for information only, the limits cannot be changed on their own. The same holds true for the spin rate limits, which are set according to the probe, profile, and cap material. If, e. g., a lower temperature limit is required the rotor must be equipped with suitable cap. If, e. g., slower spinning is desired a suitable profile needs to be chosen.
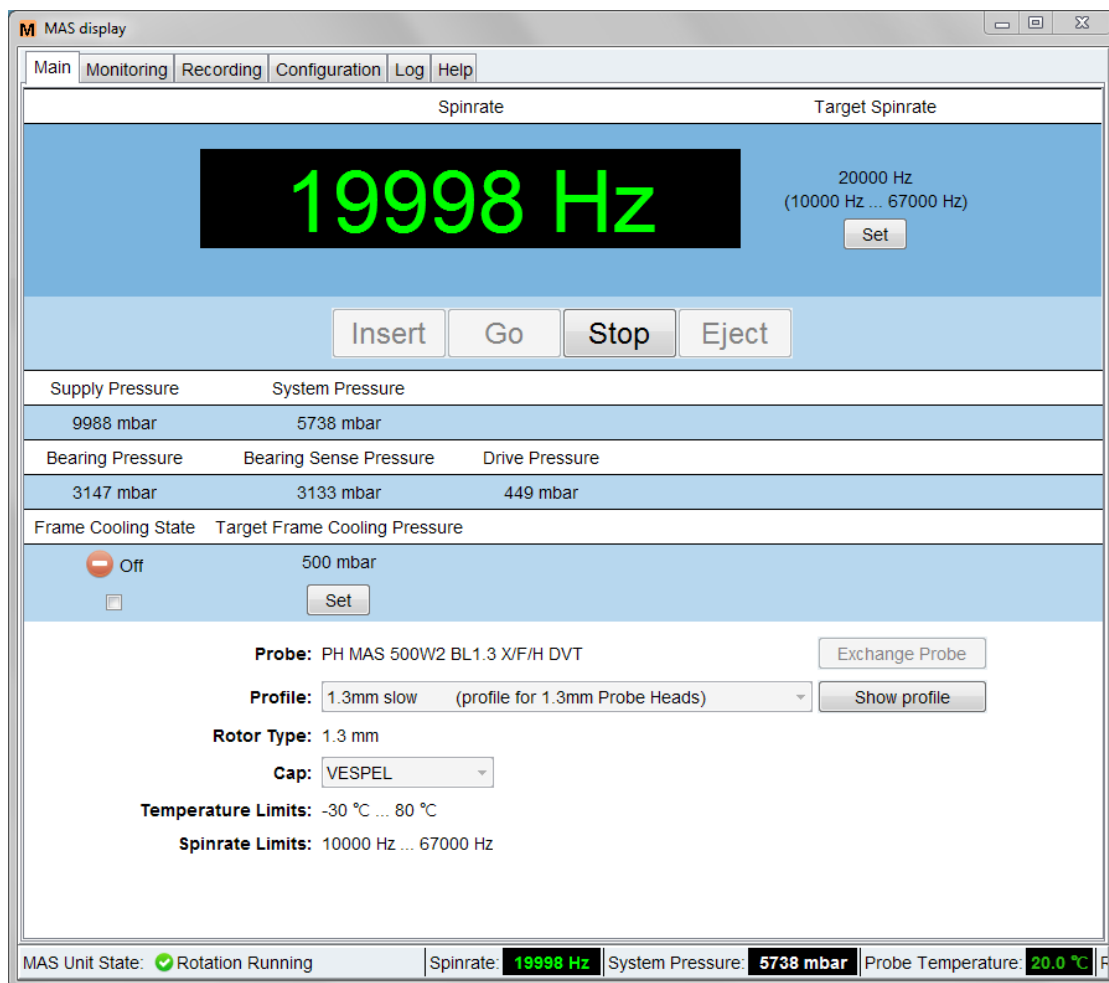
*Figure 11.1: The **Main** Tab of the TopSpin MAS User Interface.*

11.1.3.5 Rotor Handling

Rotor insertion and removal are performed by clicking the **Insert** or **Eject** button, respectively. The unit will perform the necessary steps according to the probe type like applying insert and eject air, stator flipping for probes equipped with a flip type stator etc. The **Eject** button is enabled as long as no rotation is running even when no sample is in the system. Both the **Insert** and the **Eject** button are inactive during sample spinning.

The target spin rate is shown on the top right side of the **Main** window together with the allowed spin rate range. The user may set the spin rate by clicking the **Set** button, this will open a small window in which the desired value may be typed. Changing the target frequency is always possible but not during the start and stop rotation phases. Entries outside of the allowed range cannot be set. Spinning is started or stopped by clicking the **Go** button or **Stop** button, respectively. The **Go** button is inactive before **Insert** is applied at least once. The **Stop** button is inactive if the rotor does not spin. The current spin rate is shown on the upper left side of the **Main** window, a color code is used to indicate regulation status:

- Green color indicates that the rotor is spinning at the desired spinning speed with a defined tolerance (this is set in the **Configuration** tab, see *The Configuration Tab of the TopSpin User Interface [▶ 183]*).

- Red color indicates deviations from the desired spin rate.

Such deviations occur during normal operation when the rotor is spun up or down, or the spinning speed is changed. They may as well indicate a problem like instable spinning due to an imbalanced rotor or strong fluctuations of the supply pressure. In case the spinning stability is lost a corresponding message will be displayed just below the display of the current spin rate. The user may delete this message by clicking the **Reset** button on its right side, see the next figure. The color code may be changed by the user in the **Configuration** tab of the TopSpin user interface.
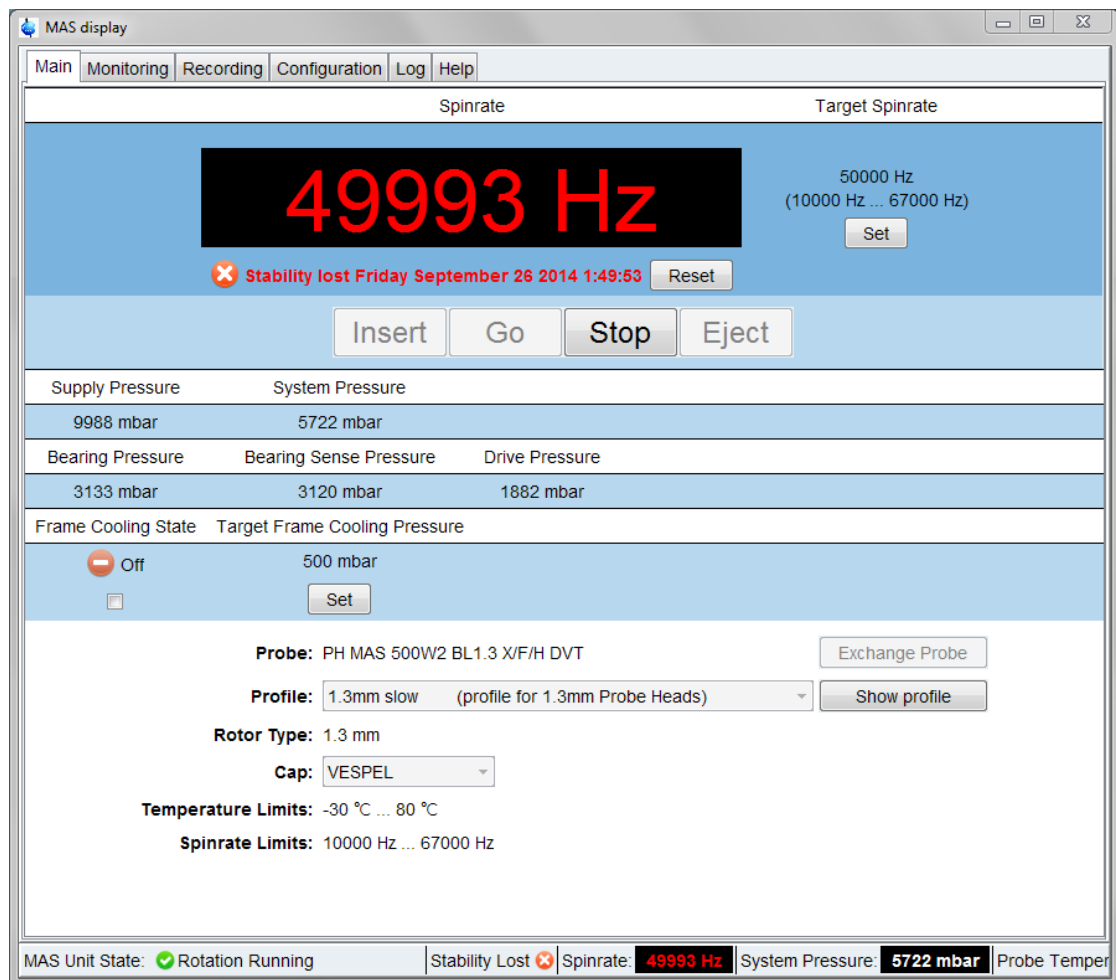
*Figure 11.2: The **Main** Tab of the TopSpin MAS User Interface in Case of Instable Spinning.*

See also

Spinning and Stopping [▷ 179]

11.1.3.6   Pressure Display

The supply pressure is the pressure of the gas supply for the MAS unit. This pressure is reduced to the system pressure in the unit which in turn is used to set bearing, drive, etc. The internal pressure has to be adjusted on the pressure regulator near the gas tank (refer to chapter Gas Buffer Connections).

The bearing and drive pressures are set by the MAS unit according to the probe, profile, and desired spinning speed. Both pressures are set by the MAS unit and displayed in the main window. The display is for information only, the values cannot be changed. The bearing sense pressure is measured at the probe directly for safety purposes: The MAS unit supervises the bearing sense pressure constantly and stops the spinning if its value drops

below a certain cutoff limit. This is intended to prevent a rotor damage caused by a loss of bearing pressure between MAS unit and probe. Such a loss may, e. g., occur if the MAS hoses or joints are not tight or damaged.

11.1.3.7 Frame Cooling

The frame cooling is a stream of ambient temperature gas which is applied to the probe's frame. This is intended as a cooling of the electric parts in high RF and high duty cycle experiments. It is as well used to keep the probe's electric parts at ambient temperature in high or low temperature experiments. Frame cooling is switched on by ticking the corresponding check box. The target frame cooling pressure may be set by clicking the **Set** button and typing in the desired value.

The frame cooling may be used for other purposes, e. g. for transfer line cooling.

11.1.4 The Monitoring Tab of the TopSpin User Interface

11.1.4.1 Purpose

This tab is provided for information purposes. It supplies a graphical display of the spin rate and pressure settings or readings as a function of time, see the figure in section *Graphical Display [▶ 181]*.

11.1.4.2 Configuration

The user may define the displayed parameters by ticking the corresponding checkboxes in the **Configure** section of the tab. The parameter list may be shown or hidden by clicking the **Down arrow** / **Up arrow** button, respectively, on top of window (left to **Configure**). The following settings can be made.

- General: Update interval in seconds.
- Parameters.
- Spin rate: Target and actual spin rate.
- Pressures: Supply pressure and system pressure.
- Bearing/Drive pressure

If the **Auto scale** check box is marked the vertical and horizontal axes will constantly be rescaled to fit to the actual spin rate and pressure settings or readings. The **Overview** mode is similar to the TopSpin overview mode for spectra display, if the corresponding box is checked the full curve will always be shown on top of the actual display.

11.1.4.3 Graphical Display

For each of the parameter groups described above a separate graph will be shown (if no parameter of the corresponding group is selected, no graph will be shown). Each display has got icons similar to the TopSpin icons for expanding/compressing the horizontal or vertical scales. If the **Overview** mode is set (see above) the full curve will always be shown on top of the current display.
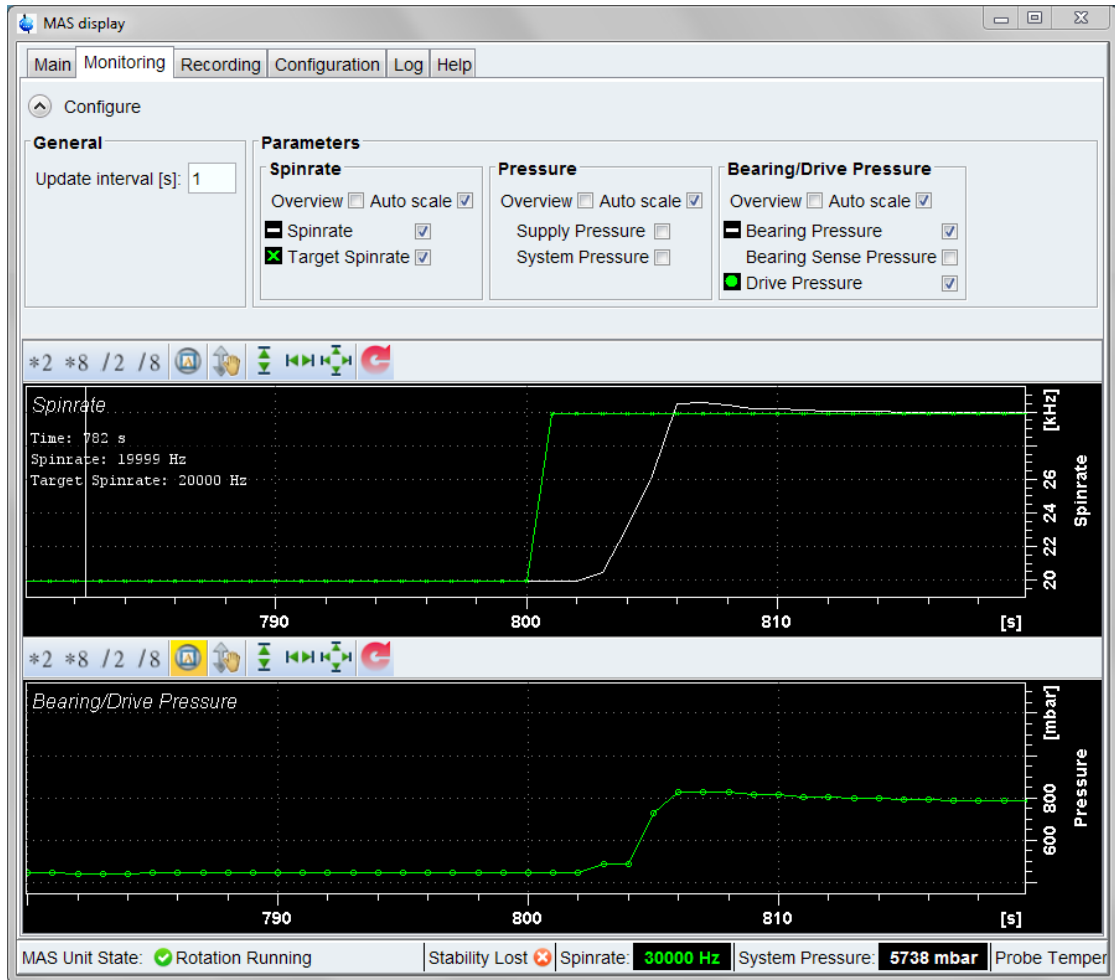
*Figure 11.3: The **Monitoring** Tab of the TopSpin MAS User Interface.*

11.1.5 The Recording Tab of the TopSpin User Interface

11.1.5.1 Purpose

The user may write and inspect record files in which spin rate and pressures settings and readings of the MAS unit are stored at defined time steps. The figure in section *Monitoring of the Recorded Parameters [▶ 183]* shows the **Recording** tab.

11.1.5.2 Defining the Recorded Parameters

The left side of the **Recording** tab is for user input and for the start of stop of the recording. The following sections are given:

- Start/Stop Recording: The user may start or stop the recording by clicking the appropriate button.

- Record files: This shows a list of previously written record files. The file content may be displayed graphically (button **Show graph**) or edited (button **Show file content**). The user may delete files by clicking the **Delete** button. The user will be prompted to confirm the deletion.

- Options. Here the recording interval and the time format are set and the user may define a suitable record file name. Record files are stored in the XXX directory.

11.1.5.3   Monitoring of the Recorded Parameters

On the right side of the **Recording** window the last record lines written to file are displayed. The user may inspect these lines e. g. to check the proper choice of parameters and time interval. The display may be cleared by clicking the **Clear** button.
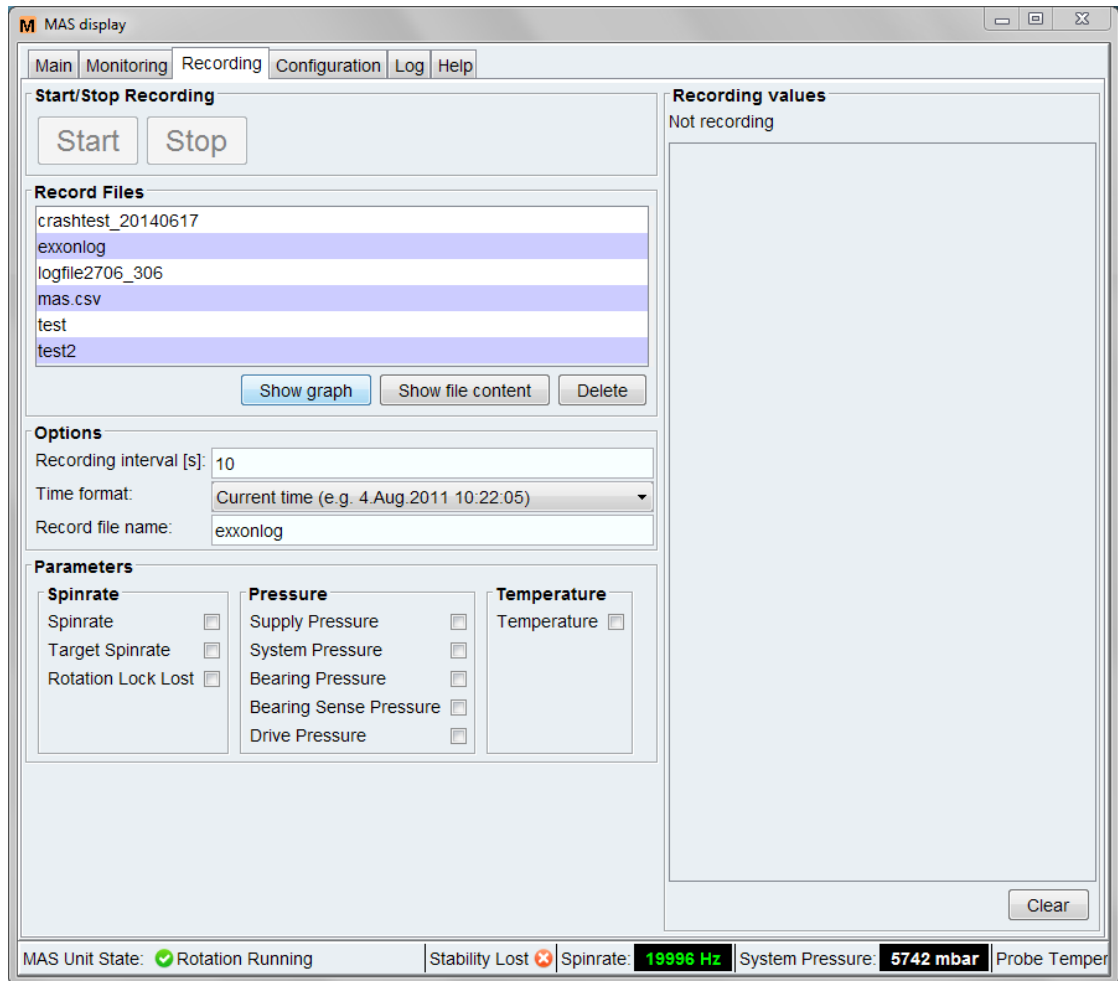
*Figure 11.4: The **Recording** Tab of the TopSpin MAS User Interface.*

11.1.6   The Configuration Tab of the TopSpin User Interface

11.1.6.1   Purpose

This tab allows some settings for the graphical appearance of the TopSpin MAS interface. It is not intended to configure the MAS unit. For the latter purpose, however, a link to the MAS unit's web page is supplied.

11.1.6.2   Settings

The following sections are available:

Stability Configuration:

- Lock tolerance: the *lock tolerance interval* defines the maximum deviation of the actual spin rate from the target spin rate which is regarded as acceptable for stable spinning. The default value is ±5 Hz, the user may change this to smaller or larger figures. It cannot be changed while a rotation is starting or stopping.

> **i** The spin lock tolerance limit is only a monitoring parameter, it has no impact on the spinning regulation. It is used for the spinning speed color code, the stability indicator in the TopSpin MAS interface status bar, and the *Rotation Lock Lost* entry of the record files.

Color

- Color for stable rotation: This is the color for the spin rate display for stable spinning, the default color is green, the user may choose a different color by clicking the **Select** button.
- Color for unstable rotation: This is the color for the spin rate display if spinning is unstable, the default color is red, the user may choose a different color by clicking the **Select** button.
- Use default colors: This button may be clicked to undo any user choices and return to the default color scheme.

Location

- External TopSpin window: The MAS user interface will show up in a separate window which may be handled independently of the main TopSpin window.
- Internal TopSpin window: The MAS interface will be incorporated into the main TopSpin window.

MAS Webpage

- MAS main webpage: Clicking the **Show** button will open a web browser and display the main MAS unit webpage.
- MAS firmware update: Clicking the **Show** button will open a web browser and display the MAS unit webpage for firmware update.
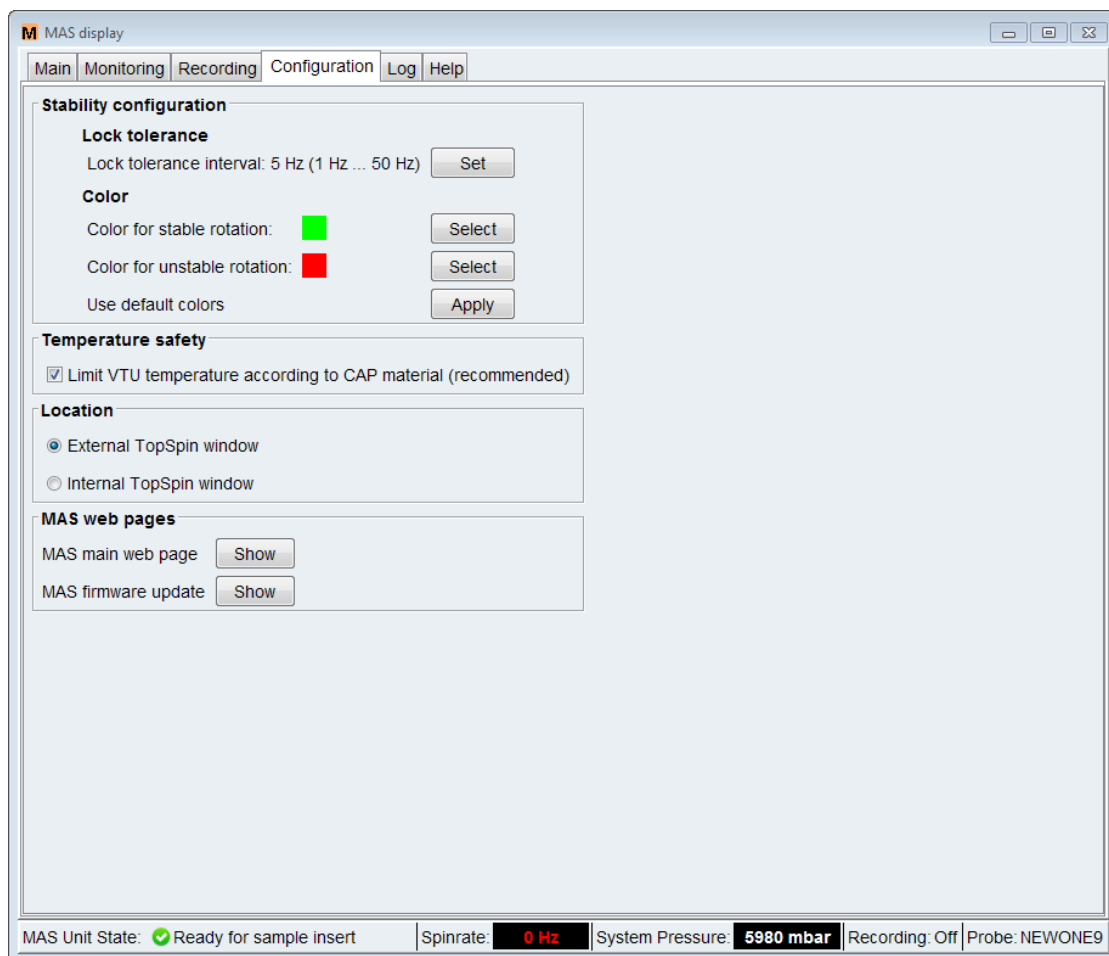
*Figure 11.5: The **Configuration** Tab of the TopSpin MAS Unit Interface.*

11.1.7 The Log Tab of the TopSpin User Interface

11.1.7.1 Purpose

This tab allows to inspect, print, or clear notifications, error messages and information supplied by the MAS unit.

11.1.7.2 Categories

The following categories are displayed:

- MAS Unit Notifications.
- Current error messages.
- Error message history: The **Reload** button updates the display to include the most recent entries.
- Info messages.
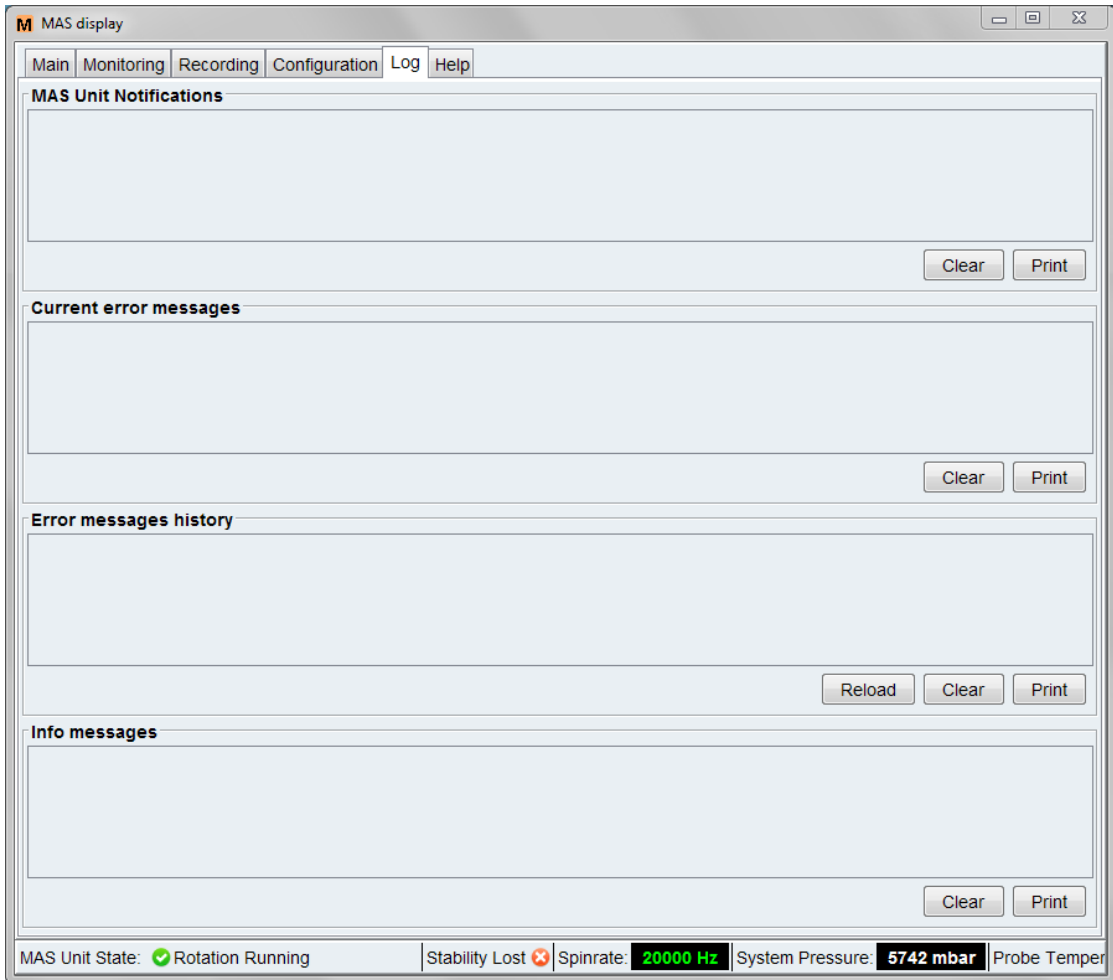
*Figure 11.6: The **Log** Tab*

To print a message category:

- Click the **Print** button.

To empty a message category:

- Click the **Clear** button.

11.1.8   The Help Tab of the TopSpin User Interface

This tab provides access to the MAS unit manual.
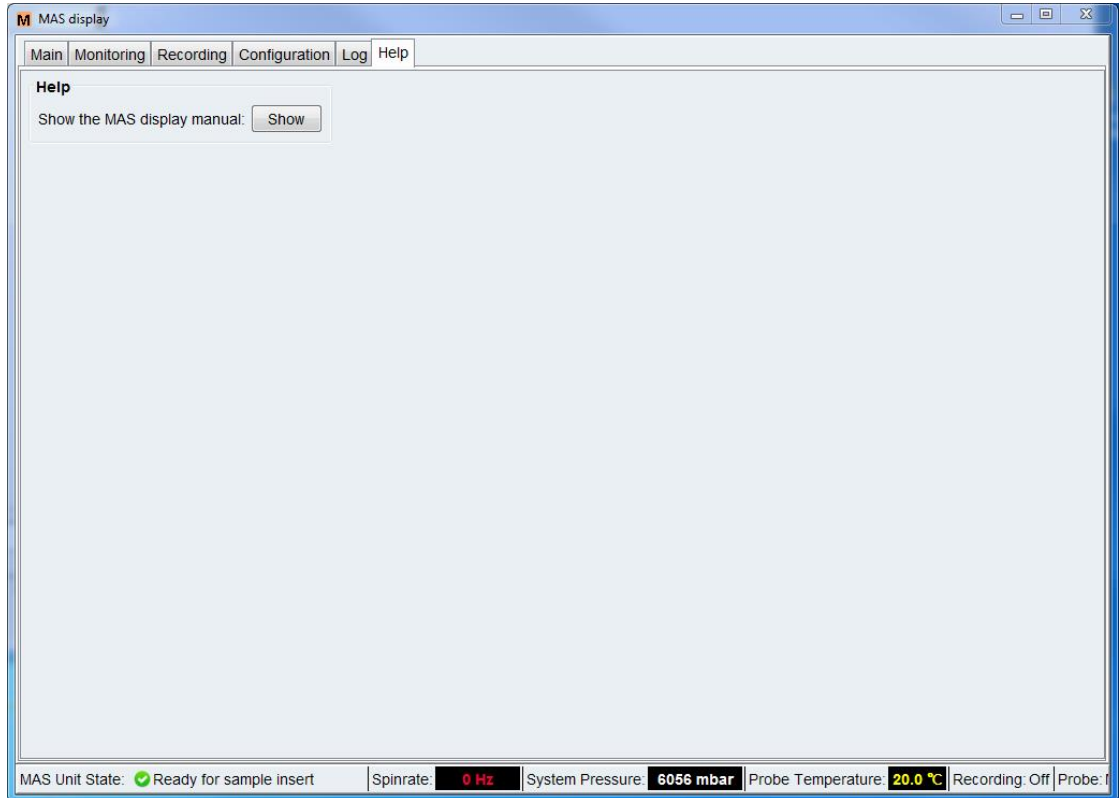
*Figure 11.7: The **Help** Tab*

12 Miscellaneous

This chapter describes TopSpin commands that are somehow acquisition related and have not been described in previous chapters.

12.1 account

NAME

account - Show acquisition accounting statistics

DESCRIPTION

The command **account** prints statistics derived from the experiments performed during a desired time.

Statistics are only available if accounting is switched on. You can do this as follows:

- Click **Options** => **Preferences** [**set**].
- Click **Acquisition** in the left part of the dialog box.
- Check the item **Enable data-acquisition protocol**.
- Click **OK**.

INPUT FOLDERS

<tshome>/prog/curdir/acqhistory - acquisition accounting files

SEE ALSO

(set, hist)

12.2 crpon, crpoff, crplock, crpobs

NAME

crpon - Switch the cryo preamplifier on for the lock and observe channel.

crpoff - Switch the cryo preamplifier off for the lock and observe channel.

crpobs - Switch the cryo preamplifier on for the observe channel.

crplock - Switch the cryo preamplifier on for the lock channel.

DESCRIPTION

The **crp*** commands listed above control the internal preamplifier of a cryoprobe. Cryo probes operate at a reduced coil and preamplifier temperature which improves the signal to noise with a factor of 4. They are available as Dual or Triple resonance probes.

The command **crpon** switches from the external (HPPR) preamplifier to the internal cryo preamplifier. This happens for both the observe and the lock channel.

The command **crpoff** switches from the internal cryo preamplifier to the external (HPPR) preamplifier. This happens for both the observe and the lock channel.

The command **crplock** switches from the external (HPPR) preamplifier to the internal cryo preamplifier. This happens for the lock channel only.

The command **crpobs** switches from the external (HPPR) preamplifier to the internal cryo preamplifier. This happens for the observe channel only.

SEE ALSO

edhead [▷ 111]

12.3 delpar, delgp, delsh

NAME

delpar - Delete parameter sets.

delgp - Delete **gradient programs.**

delsh - Delete shim files.

SYNTAX

delpar [<name>]

delgp and delsh have the same syntax as delpar

DESCRIPTION

The **del*** commands without argument open a dialog box with a list of the available entries. They are equivalent to the respective edit commands **rpar**, **edgp** and **rsh**. From the dialog box, you can select entries and click **Delete** to delete them. Alternatively, you can use the **del*** command with one argument, for example:

    delpar myparset

Deletes the parameter set **myparset**.

> ℹ   Note that only user defined entries can be removed.

INPUT FOLDERS

- <tshome>/exp/stan/nmr/par - Bruker and user defined parameter sets.
- <tshome>/exp/stan/nmr/lists/gp - Bruker and user defined gradient programs.
- <tshome>/exp/stan/nmr/lists/bsms - shim files.

USAGE IN AU PROGRAMS

- DELPAR(name).
- No AU macros are available for the other **del*** commands.

SEE ALSO

edlist, dellist [▷ 138], (delau, delpul, delcpd, delpy, delmac)

12.4   exprof

NAME

exprof - Display excitation profile of selective experiment.

DESCRIPTION

The command **exprof** displays the excitation profile of a selective experiment. The experiment must have been completely setup for acquisition, before **exprof** is executed.

The command **exprof** opens the dialog shown in the next figure.
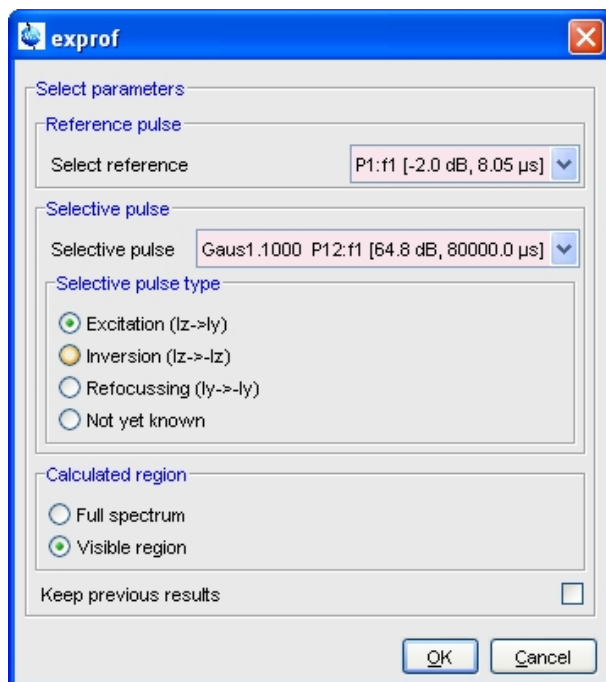
Figure 12.1: exprof

The dialog offers the following options:

- **Reference Pulse**

  The 90° hard pulse in the same channel as the selective pulse.

- **Selective Pulse**

  The selective pulse to be examined. If your pulse programs contains only one selective pulse, and it follows the Bruker style guide (P1=90°, etc.) **exprof** automatically selects the correct reference pulse.

- **Selective Pulse Type**

  The rotation type. Automatically set to the value stored in the shape file. If the shape file does not contain a rotation type, it is set to „not yet known" and must be selected in the dialog. Shapes created with TopSpin 1.3 or older, or with external programs do not contain the rotation type.

- **Calculated Region** (greyed if no spectrum exists)

  The profile can be calculated for the entire spectrum or for the currently displayed region. It always consist of 1000 points.

- **Keep Previous Results**

  If enabled, the result of previous **exprof** command(s) is kept, allowing simultaneous display of multiple profiles.

If your pulse program contains only one selective pulse and the waveform contains rotation type, you don't have to set anything in the **exprof** dialog. Just click **OK**, to confirm the dialog.

The command **exprof** requires that the *NMRSim* program, delivered on the TopSpin DVD, is installed.

OUTPUT PARAMETERS

- PULPROG - pulse program used for the acquisition
- SHAPE - array of shaped pulse parameters

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters
- <tshome>/exp/stan/nmr/lists/wave/*
  - Shape files
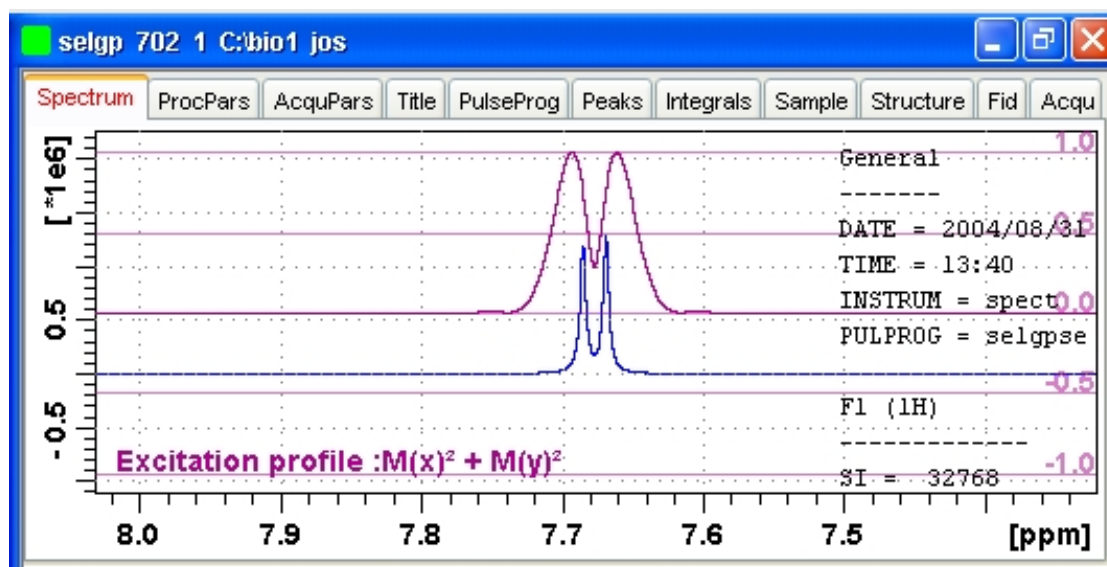
Figure 12.2: selgp

SEE ALSO

(nmrsim)

12.5   getlim1d, getlcosy, getlxhco, getljres, getlinv

NAME

getlim1d - Calc. 1D spectral width according to integral ranges.

getlcosy - Calc. 2D cosy spectral width from 1D int. ranges.

getlxhco - Calc. 2D xhco spectral width from 1D int. ranges.

getljres - Calc. 2D jres spectral width from 1D int. ranges.

getlinv - Calc. 2D inverse spectral width from 1D int. ranges.

DESCRIPTION

The **getl*** commands listed above calculate and set the optimum spectral width for the specified experiment types. The optimum spectral width is determined from one or two associated 1D datasets, which are defined as the so called second and third dataset. Before the actual experiment is performed, the second (and if necessary the third) dataset must be acquired, Fourier transformed and baseline corrected. The latter processing step implicitly determines the integral ranges. The **getl*** commands determine the spectral width such that it includes all integral ranges, in other words, all signals.

getl* commands are typically used in automation. They are called from AU programs like **au_getl1d**, **au_getlcosy**, **au_getlinv** and **au_getlxhco**. These, in turn, are called by IconNMR where the preparation and the actual experiment are defined as a so-called composite 2D experiment (see IconNMR Online help for more information).

getlim1d determines the optimum spectral width on a 1D preparation experiment and then sets the parameter SW on the current 1D dataset accordingly.

getlcosy determines the optimum spectral width on one 1D preparation experiment. Then it sets the F2-SW and F1-SW on the current 2D COSY dataset accordingly.

getlxhco determines the optimum spectral width for F2 and F1 on two different 1D preparation experiments (typically 1H and X). Then it and sets the F2-SW and F1-SW on the current 2D XH correlation dataset accordingly.

getljres determines the optimum spectral width on one 1D preparation experiment. Then it sets F2-SW on the current 2D J-resolved dataset in accordingly.

getlinv determines the spectral width on one 1D preparation experiment. Then it sets F2-SW on the current 2D INVERSE dataset accordingly.

> **i** Note that F2-SW refers to the acquisition parameter SW in the F2 direction and F1-SW refers to the same parameter in the F1 direction.

The first 1D preparation experiment is defined as the so called second dataset. The second 1D preparation experiment is defined as the so called third dataset.

OUTPUT PARAMETERS

- SW - spectral width in ppm

INPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/pdata/<procno>
  - *intrng* - integral regions
  - *curdat2* - definition of second and third dataset

OUTPUT FILES

- <dir>/data/<user>/nmr/<name>/<expno>/
  - *acqu* - acquisition parameters
  - *acqu2* - F1 acquisition parameters of a 2D dataset

USAGE IN AU PROGRAMS

- GETLIM
  executes the command **getlim1d**
- GETLCOSY
- GETLXHCO

- GETLJRES
- GETLINV

12.6   hpdisp

NAME

hpdisp - Pulse program display showing signals going to spectrometer hardware, inclusive exact timing, phase and amplitude.

DESCRIPTION

The command **hpdisp** opens a new window with a display of all hardware events on selected channels.

hpdisp presents the results of the results of the pulse program simulation using the same software which sends the commands to hardware in a real experiment.

> ℹ️ Please note that displaying each event going to the spectrometer hardware may lead to a very large number of displayed entries, even for simple pulse programs (see the next figure).
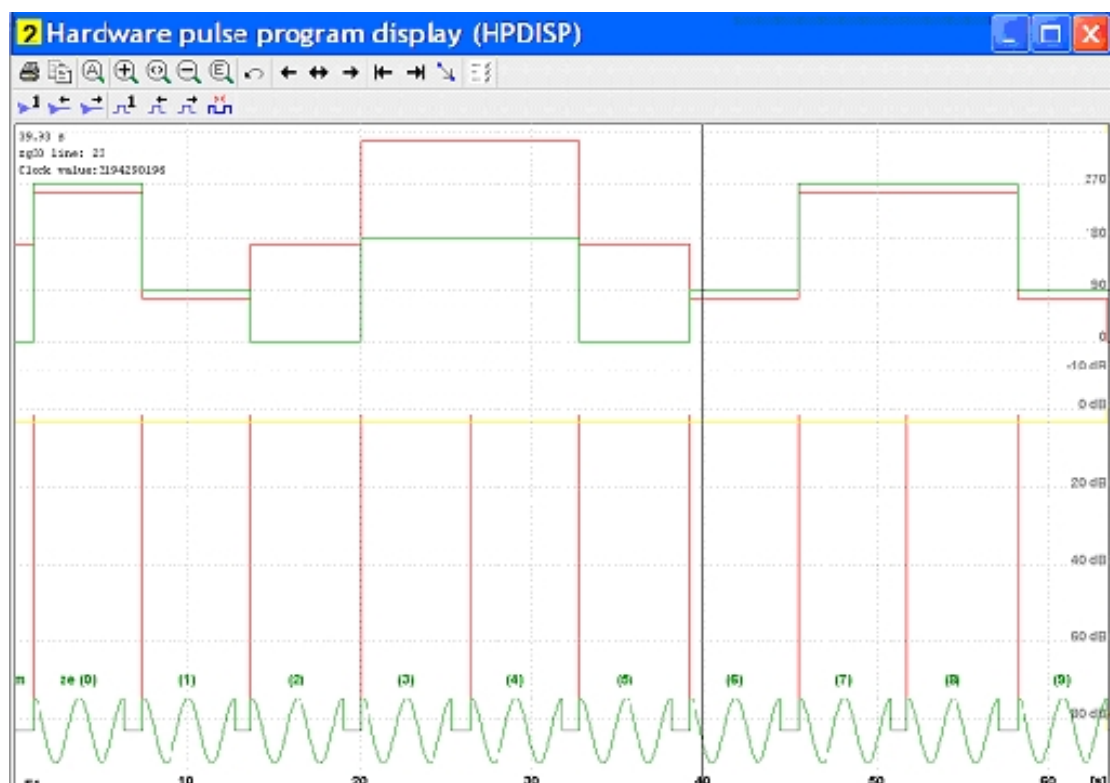
Figure 12.3: hpdisp

E.g. simple 1D experiment using adiabatic pulses for decoupling may generate millions of spectrometer events.

SEE ALSO

spdisp [▷ 195]

12.7   spdisp

NAME

spdisp - Symbolic (graphical) pulse program display.

DESCRIPTION

The command **spdisp** opens a new window with a symbolic pulse program display.
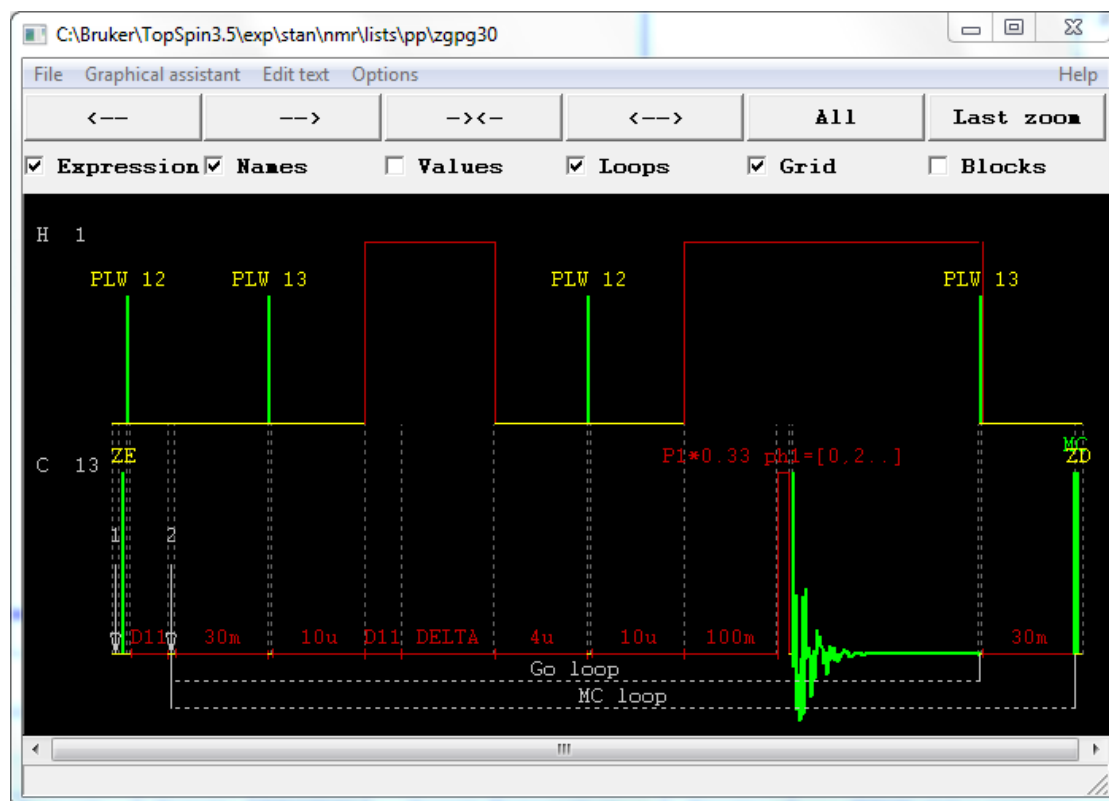
Figure 12.4: spdisp

It takes one argument and can be used as follows:

- **spdisp**

  Displays the pulse program defined by the acquisition parameter PULPROG.

- **spdisp <name>**

  Displays the pulse program <name>.

spdisp starts the pulse program display part of *NmrSim*. The functionality of this program is described in the *NMRSim* manual. This can be opened by clicking **Help => Manuals =>** [**Analysis and Simulation**] **NMRSim Experiment Simulation**.

spdisp can only start if the following conditions are met:

- The Bruker AU programs have been installed (with **expinstall**).
- The *NMRSim* program is installed (from the TopSpin DVD).

SEE ALSO

(nmrsim)

12.8 stdisp

NAME

stdisp - *ShapeTool* for handling RF shapes and gradients.

DESCRIPTION

The command **stdisp** opens the *ShapeTool* window where you can create, manipulate and analyse RF shapes and gradients (see the next figure).
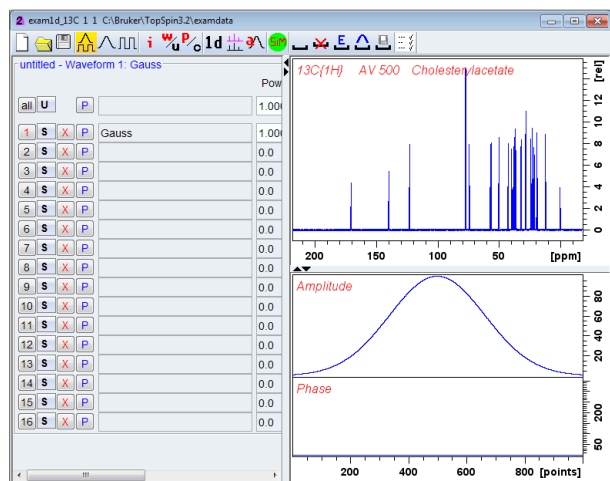
Figure 12.5: stdisp

The TopSpin menu bar will change, showing the entries **Shape** and **Manipulate** and *ShapeTool* submenus in **Analysis** and **Options**.

The functions of the buttons are available as tooltips. A detailed description of **stdisp** can be found under **Help => Manuals => [Acquisition Application Manuals]** *ShapeTool*.

SEE ALSO

st [▷ 196]

12.9 st

NAME

st - Generate, manipulate and analyze RF shapes and gradients

DESCRIPTPION

The command **st** allows you to generate, manipulate and analyze RF shapes and gradients. It is described in a separate manual which can be opened as follows:

click **Help => Manuals => [Acquisition Application Manuals]** *ShapeTool*

SEE ALSO

stdisp [▷ 196]

12.10 wvm

NAME

wvm – WaveMaker (Software solution for pulse shaping)

DESCRIPTION

WaveMaker is a software solution for pulse shaping that is based on wave form definitions within the pulse programs. This avoids any need to modify the PROSOL tables when new or non-standard shaped pulses and decoupling or mixing wave forms are required. It also adds flexibility to and improves portability of many standard and user developed experiments and pulse programs. Porting the WaveMaker based pulse programs to NMR systems of different field strength is straightforward and typically requires no further adjustments and PROSOL updates.

For detailed information please have a look at the WaveMaker – Pulse Shaping Software (User Manual). This can opened by clicking *Help → Manuals → Acquisition - Application Manuals→ WaveMaker*.

Miscellaneous

13 Contact

Manufacturer

Bruker BioSpin GmbH

Silberstreifen 4

D-76287 Rheinstetten

Germany

http://www.bruker.com

WEEE DE43181702

NMR Hotlines

Contact our NMR service centers.

Bruker BioSpin NMR provides dedicated hotlines and service centers, so that our specialists can respond as quickly as possible to all your service requests, applications questions, software or technical needs.

Please select the NMR service center or hotline you wish to contact from our list available at:

https://www.bruker.com/service/information-communication/helpdesk.html

Phone: +49 721-5161-6155

E-mail: *nmr-support@bruker.com*

Contact

List of Figures

List of Figures

List of Figures

List of Tables

List of Tables

Index

Index