

# **NMR** SUITE

## **Acquisition**

### *Reference Manual*



---

Copyright (C) 2003 by Bruker BioSpin GmbH

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means without the prior consent of the publisher.

Part-No: H9775A3

14.03.2003

Product names used are trademarks or registered trademarks of their respective holders.

Bruker software support is available via phone, fax, e-mail, Internet, or ISDN.  
Please contact your local office, or directly:

Address: Bruker BioSpin GmbH  
Software Department  
Silberstreifen  
D-76287 Rheinstetten  
Germany

Phone: +49 (721) 5161 455  
Fax: +49 (721) 5161 943  
E-mail: [nmr-software-support@bruker.de](mailto:nmr-software-support@bruker.de)  
FTP: <ftp.bruker.de> / <ftp.bruker.com>  
WWW: [www.bruker-biospin.de](http://www.bruker-biospin.de) / [www.bruker-biospin.com](http://www.bruker-biospin.com)  
ISDN: on request

---

# Contents

---

<b>Chapter 1</b>	<b>Introduction .....</b>	<b>A-3</b>
1.1	About this manual .....	A-3
1.2	Conventions .....	A-3
1.3	About dimensions .....	A-4
1.4	About digitally filtered Avance data .....	A-5
1.5	Usage of acquisition commands in AU programs .....	A-5
1.6	Starting commands from the XWIN-NMR menu .....	A-6
<b>Chapter 2</b>	<b>XWIN-NMR parameters .....</b>	<b>A-7</b>
2.1	About XWIN-NMR parameters .....	A-7
2.2	Parameter value types .....	A-10
2.3	Parameter files .....	A-10
2.4	Acquisition (eda) parameters .....	A-11
2.5	Acquisition status (dpa) parameters .....	A-52
2.6	Routing (edasp) parameters .....	A-57
2.7	Lock (edlock) parameters .....	A-58
<b>Chapter 3</b>	<b>Spectrometer configuration commands .....</b>	<b>A-61</b>
<b>Chapter 4</b>	<b>Lock commands .....</b>	<b>A-103</b>
<b>Chapter 5</b>	<b>Shim commands .....</b>	<b>A-119</b>
<b>Chapter 6</b>	<b>Probehead commands .....</b>	<b>A-133</b>
<b>Chapter 7</b>	<b>Parameter handling commands .....</b>	<b>A-149</b>
<b>Chapter 8</b>	<b>Pulse and AU program commands .....</b>	<b>A-171</b>
<b>Chapter 9</b>	<b>Acquisition commands .....</b>	<b>A-191</b>
<b>Chapter 10</b>	<b>Temperature commands .....</b>	<b>A-227</b>
<b>Chapter 11</b>	<b>MAS/Solids commands .....</b>	<b>A-239</b>
<b>Chapter 12</b>	<b>Rename and delete commands .....</b>	<b>A-259</b>
<b>Chapter 13</b>	<b>Miscellaneous .....</b>	<b>A-267</b>
<b>Chapter 14</b>	<b>NMR Suite files .....</b>	<b>A-279</b>
<b>Chapter 15</b>	<b>Bruker addresses .....</b>	<b>A-293</b>

## Command Index

## Index



---

# Chapter 1

## Introduction

---

### 1.1 About this manual

---

This manual is a reference to XWIN-NMR 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 XWIN-NMR 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

---

#### Font conventions

***zg*** - commands to be entered on the command line are in courier bold italic

***wobb-SW*** - commands to be clicked are in times bold italic

`g0=2` - pulse program statements are in courier small

*fid* - filenames are in courier

*contents* - any contents of a text file is in courier small

*name* - any name which is not a filename is in times italic

### **File/directory conventions**

<xwhome> - the XWIN-NMR home directory (default C\;Bruker or /u)

### **Header conventions**

SYNTAX - only included if the command described requires arguments

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

### **Spectrometer naming conventions**

This manual is written for Avance spectrometers. Nevertheless, a large part of it is also valid for older spectrometers like AMX, ARX and ASX. Since the end of 1999, Bruker delivers a new type of Avance spectrometers that are specified in this manual as Avance-AQS. The conventional Avance spectrometers are specified as Avance-AQX. You can easily find out which type of spectrometer you have by opening the cabinet door; one of the racks is named either AQS or AQX. Note that there are three types of Avance-AQX spectrometers: DMX, DRX or DPX. These specific names are used in this manual whenever a description only holds for one or two of them.

## **1.3 About dimensions**

---

XWIN-NMR can acquire 1, 2 or 3 dimensional data. The dimensions of a dataset are indicated with the terms F3, F2 and F1 which are used as follows:

1D data

F1 - acquisition dimension

2D data:

F2 - acquisition or direct dimension

F1 - indirect dimension

3D data:

F3 - acquisition or direct dimension

F2 - indirect dimension

F1 - indirect dimension

In 3D processed data, F2 is always the second and F1 the third dimension. 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. Note that ICON-NMR 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 an XWIN-NMR 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 XWIN-NMR macro. These are scripts created with **edmac** containing a sequence of XWIN-NMR commands. The syntax of each line is simply an XWIN-NMR command (in lowercase letters) as it would be entered on the command line.

## 1.6 Starting commands from the XWIN-NMR menu

---

This manual describes all acquisition commands as they can be entered on the command line. However, they can also be clicked from the XWIN-NMR popup menus. Most acquisition commands can be found under the *Acquire* menu. There, the corresponding command line commands are specified in square brackets.



---

# Chapter 2

## XWIN-NMR parameters

---

### 2.1 About XWIN-NMR parameters

---

XWIN-NMR parameters are divided in acquisition, processing, plot and output 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.

### prosol parameters

Probehead 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.

### 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:

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

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. Here are some examples of how you can set or view the parameter TD:

On a 1D dataset:

- ***td*** - set the parameter TD
- ***1s td*** - view the status parameter TD

On a 2D dataset:

- ***td*** - set the parameter TD in the F2 dimension (= direct dimension)
- ***2 td*** - set the parameter TD in the F2 dimension (same as ***td***)
- ***1 td*** - set the parameter TD in the F1 dimension (= indirect dimension)
- ***2s td*** - view the status parameter TD in the F2 dimension
- ***1s td*** - view the status parameter TD in the F1 dimension

On a 3D dataset:

- ***td*** - set the parameter TD in the F3 dimension (= direct dimension)
- ***3 td*** - set the parameter TD in the F3 dimension (same as ***td***)
- ***2 td*** - set the parameter TD in the F2 dimension (= indirect dimension)
- ***1 td*** - set the parameter TD in the F1 dimension (= indirect dimension)
- ***3s td*** - view the status parameter TD in the F3 dimension
- ***2s td*** - view the status parameter TD in the F2 dimension
- ***1s td*** - view the status parameter TD in the F1 dimension

Although status parameters are normally not changed by the user, a command like ***1s td*** 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. Table 2.1 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**

## 2.3 Parameter files

XWIN-NMR parameters are stored in various files in the dataset directory tree.

In a 1D dataset:

```
<du>/data/<user>/nmr/<name>/<expno>/
```

acq - acquisition parameters

acqus - acquisition status parameters

In a 2D dataset:

```
<du>/data/<user>/nmr/<name>/<expno>/
```

acq - F2 acquisition parameters  
acq2 - F1 acquisition parameters  
acqus - F2 acquisition status parameters  
acq2s - F1 acquisition status parameters

In a 3D dataset:

<du>/data/<user>/nmr/<name>/<expno>/

acq - F3 acquisition parameters  
acq2 - F2 acquisition parameters  
acq3 - F1 acquisition parameters  
acqus - F3 acquisition status parameters  
acq2s - F2 acquisition status parameters  
acq3s - 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*, *rga* and *pulsdisp*. Some, however, are only interpreted by specific commands which are then specified in the list below. Acquisition parameters can be set with **eda** or by typing their 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 **gs** by adjusting (FID display)
- can be only used on Avance-AQS spectrometers
- 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 2.6. 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*. Note that, in order to use DQD, your Avance spectrometer must be equipped with a DQD device on the RCU.

#### 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 NUCLEI 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. 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 XWIN-NMR 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 `cpdng*` and `cpdngs*` statements are



available for Avance-AQX but not for the Avance-AQS.

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 **edpu1**. 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** → **Other topics** → **Writing pulse programs**

DDR - digital digitizer resolution

- takes an integer value
- temporary parameter calculated according to the equation:

$$DDR = \lceil \log_2(DECIM) \rceil + 1$$

- 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 2.2) 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. On Avance-AQX spectrometers, DE consists of 5 pre-scan subdelays DEPA, DEX, DE1, DE2 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  $\mu$ sec)

DEX: the receiver gate is opened (default 3  $\mu$ sec)

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  $\mu$ sec). The intermediate frequency is only used for `AQ_mod = DQD` or, if your spectrometers has an RX22 receiver, for any value of `AQ_mod`.

DE2: the phase of the receiver is set to zero (default 1  $\mu$ sec)

DEADC: the digitizer is enabled (default 4  $\mu$ sec)

Note that sub-delay DE2 is used on Avance-AQX but not on Avance-AQS.

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  $\mu$ sec. On Avance-AQS, the DE2 does not exist.

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:

$$\text{DECIM} = \text{DW}/\text{DWOV}$$

where DWOV is the oversampling dwell time. Note that DECIM can only take an integer value and DWOV must be greater than the minimum value for the current digitizer (see table 2.2). See also the acquisition status parameter DECIM.

#### DIGMOD - digitizer mode

- takes one of the values *analog*, *digital*, *homodecoupling-digital*
- 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 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 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.

#### DIGTYP - digitizer type

- takes one of the values listed in table 2.2.
- 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 digi-

tizer, you might want to change the default value of DIGTYP. Note that the SADC digitizer cannot be used for sequential acquisition (AQ\_mod = qseq).

<b>digtyp</b>	<b>digitizer resolution</b> (bit)	<b>DWOV range</b> (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**

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 **1 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 2.2) 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 than 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.

#### DSLIS - dataset list

- takes a character array value
- can be set from **eda** by entering a name in the *DSLIS* field or by clicking the down arrow and selecting a name from the appearing list. Note that this list also contains the entry *EDIT CURRENT* which allows you to edit the currently defined dataset list.
- can also be set by entering **dslist** on the command line
- interpreted by the pulse program statements `wr #n`, `wr ##`, `ifp`, `dfp`, `rfp`
- DSLIS defines the name of a variable dataset list. Such a list can be created with **edlist** → **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.

The list defined by DSLIS 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*, *medium*, *smooth*, *user\_defined*
- 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 *XWINNMRHOME/exp/stan/nmr/lists/DSPFIRM*.

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 2.2)
- 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  $\mu$ s rather than every DW  $\mu$ s 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 2.2). 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 ICON-NMR
- ICON-NMR 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 **edas** 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. Note that the value that you enter for FIDRES is 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 fulfil the same equation.

FnMODE - Acquisition mode of the indirect dimensions (2D and 3D)

- takes one of the values described below
- for 2D data, it can be set with **eda** or by entering **1 fnmode**

- for 3D data, it can be set with **eda** or by entering **2 fnmode** (F2 dimension) or **1 fnmode** (F1 dimension)
- interpreted by the pulse program statement `mc`

The parameter FnMODE is available in XWIN-NMR 3.0 and newer. It defines the acquisition mode of the indirect dimensions in a multi-dimensional experiment. The use of the `mc` statement and the FnMODE parameter, simplifies the switching of the acquisition mode and allows you to use the same pulse program for various experiments. For this reason, the number of Bruker pulse in XWIN-NMR 3.0 could be reduced compared to previous versions.

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 degrees. 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 degrees. 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 degrees. 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 degrees. 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* → *Other topics* → *Writing pulse programs*

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  $\mu$ s. 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/homo-decoupling-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)
- can be set from **eda** by clicking *GP031 edit*

Table 2.3

gradient parameters				
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

- the gradients are interpreted by pulse program statements `gron0 - gron31` and `:gp0 - :gp31`
- the filenames are interpreted by pulse program statements `:gp0 - :gp31`
- XWIN-NMR 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 dimension from the GP031 table. The `groff` command switches the static gradients off. According to table 2.3, 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 dimension. According to the table 2.3, 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 Shape Tool (command **stdisp**). The gradient files reside in the directory:

*XWINNMRHOME/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 dimensions. 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 XWIN-NMR. 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.

HPPMOD - 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)

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 ( $\mu\text{sec}$ )
- 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 dimension. 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*** (same as ***1 in0***)

***nd0*** (same as ***1 nd0***)

***1 sw***

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

In a 3D dataset, IN0 and ND0 play the same role in F1 indirect dimension as they do in the 2D dataset. F1, however, is the third dimension now. For the F2 indirect dimension 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*** (same as ***1 in0***)

***nd0*** (same as ***1 nd0***)

***in10*** (same as ***2 in10***)

***nd10*** (same as ***2 nd10***)

***1 sw***

***2 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 ***10, 11, 12*** 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 ***l0 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
```

```
..
```

```
l0 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
{
}
```

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**, **lopoi**, **sref**
- High resolution samples are usually locked on 2H or 19F. For these two nuclei, standard lock parameter files are delivered with XWIN-NMR. These can be edited with the command **edlock**.

#### MASR - MAS spin rate

- takes a float value
- can be set from **eda** or by entering **1 masr**
- can also be set with the commands **mas** or **masr**
- The MAS spin rate is usually set from the MAS control window which is opened with the command **mas**.

Note that entering **masr** on the command line does not merely set the parameter MASR but starts the XWIN-NMR command **masr**.

#### 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 dimension
- 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 dimension according to:

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

N0 is set to 2, if the evolution time contains a 180 degree 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 dimension
- Number of *d10* commands in the increment loops of a pulse program for 3D experiments. Used to calculate the spectral width in the F2 dimension according to

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

ND10 is typically set to 2, if the evolution time contains a 180 degree 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 OVERFLOW).

NUC1 - NUC8 - nucleus for frequency channel f1 - f8

- take a value from a predefined list of nuclei
- can be set from `eda` by clicking NUCLEI
- can also be set from `edasp` (same as clicking NUCLEI in `eda`)
- 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:

a 1D PROTON experiment without decoupling:

```
NUC1 = 1H
NUC2 = off
NUC3 = off
```

a 1D C13 experiment with 1H decoupling:

```
NUC1 = 13C
NUC2 = 1H
NUC3 = off
```

a 2D 1H experiment with 13C and 15N coupling:

```
NUC1 = 1H
NUC2 = 13C
NUC3 = 15N
```

In 2D datasets, NUC1 in the indirect dimension (F1) must be set explicitly by clicking F1-NUCLEI in `eda` (see the description of NUCLEI). In 3D datasets, this principle holds for both indirect dimensions, F2 and F1.

Because most Avance spectrometers are equipped with a maximum of four channels, NUC5 to NUC8 are normally unused. Therefore, they appear at the end of the

**eda** table.

NUCLEI - set up nuclei and spectrometer routing

- appears in **eda** but is not a parameter
- opens the routing table (**edasp**) in 1D, 2D-F2 and 3D-F3
- In 1D and in the 2D and 3D acquisition (= direct) dimension, NUCLEI is not a parameter but a button to open the routing table. From this table, you can assign nuclei to the logical frequency channels of the spectrometer and define the amplifier routing. Clicking NUCLEI in **eda** is the same as entering **edasp** on the command line. In the routing table, you can specify the nucleus or nuclei to be used for the experiment. For the f1 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. 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 NUCLEI table also shows the spectrometer routing. The hardware elements which have been detected when the spectrometer was configured (with **cf** or **config**) will appear in this list. When you select a nucleus for a certain channel, the default routing is automatically set. Normally, this is okay. 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 following rules:

- a) Between the logical frequency channels and FCU's any combination is allowed but only one to one.
- b) 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. FCU2 can only be connected to the first four amplifiers.

- c) The first four Amplifiers are automatically connected to Switchbox input in a fixed way. These connections cannot be changed.
- d) Between the Switchbox input and Switchbox output, any connection is allowed but only one to one.
- e) 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.
- f) 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  $^1\text{H}$  or 300 W for X nuclei.
- a high power output: typically 1000 W for  $^1\text{H}$  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.

Two extra switches are available to control the routing:

*Preferred preamplifier*

toggle between selective and unselective Preamplifier module

*Preferred output for  $^{19}\text{F}$*

toggle between the  $^{19}\text{F}$  and X Switchbox output for  $^{19}\text{F}$  nucleus

Note that changing these settings changes the default routing. When you change these settings, you must click the DEFAULT button to make them effective.

At the bottom of the NUCLEI table you will find the following buttons:

SAVE - save the parameters and routing 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, NUCLEI has a different functions in the two dimensions. 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 dimension. The same principle holds for 3D data; in F3, NUCLEI 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. 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.

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 *utilities* → *O1*, *O2*, *O3* (spectrum display)
- O1 can be set by clicking *utilities* → *sw-sfo1* (spectrum display)
- 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 *sw-sfo1* in the *utilities* menu can be used if you are setting up your experiment from an existing spectrum. It sets O1 to the center of the currently displayed spectral region. The buttons **O1**, **O2** and **O3** put the cursor on the spectrum. You can then set the corresponding frequency offset by moving the mouse and clicking the middle 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 **1 overflow** on the command line
- For OVERFLW = *check*, acquisition commands check for possible data overflow. 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 ( $\mu$ seconds)
- 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]`  $\mu$ seconds, `p1` executes a pulse of `P[1]`  $\mu$ seconds 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 or 3D
- 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, 2D or 3D. 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. 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 than 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 degrees.

PCPD[1-8] - array of CPD pulse lengths

- takes float values ( $\mu$ sec)
- 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]`.

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 informa-

tion, 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 **p10**, **p11** etc. on the command line
- can also be set from the **gs** dialog window
- interpreted by the pulse program statements p10, p11, p12 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:

```
p11:f2
p13: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 p11 and the command **p11** entered on the XWIN-NMR 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=p11
```

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 **edas** 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.

PROSOL - copy prosol parameters to corresponding acquisition parameters

- takes one of the values *true* or *false*
- is not a parameter but a push button in the **eda** dialog box
- Clicking the PROSOL button causes the prosol parameters to be read and copied to the corresponding acquisition parameters. As such, clicking PROSOL is equivalent to entering **getprosol** on the command line. The value of PROSOL indicates whether or not the prosol parameters have been copied, either by clicking PROSOL or running **getprosol**. Note that the **getprosol** command is automatically performed by ICON-NMR and **but-tonnmr**. Prosol parameters can be set up with the **edprosol** command. This is usually done once, after the installation of a new version of XWIN-NMR. The default relations between prosol and acquisition parameters are listed in table 7.1.

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. This list contains the entry **EDIT CURRENT** that allows you to edit the currently defined pulse program
- 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. 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 - QNP nucleus selection

- takes one of the values 1,2 or 3
- If the current probehead 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 \*\***
- only used on Avance-AQS spectrometers
- 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 and depends on the receiver system of your spectrometer. 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 **gs**.

#### RO - sample rotation frequency in Hz

- takes an integer value
- can be set from **eda** or by entering **1 ro** on the command line

- 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.

Note that RO can be set by entering **1 ro** on the command line but not by entering **ro**. The latter would start the command **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

SEOUT - SE 451 receiver unit output to be used

- takes one of the values **HR** or **BB**
- This parameter is only used on DMX spectrometers with an SE451 receiver. This parameter is normally set to **HR**.

SFO1 - SFO8 - irradiation (carrier) frequencies for channels f1 to f8

- take a double value (MHz)
- are automatically calculated from the equation:

$$\text{SFO1} = \text{BF1} + \text{O1}$$

$$\text{SFO2} = \text{BF2} + \text{O2}$$

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 **utilities** → **sw-sfo1** (spectrum display)
- The parameter SFO1 represents the irradiation (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`
- When you open **eda** and click on *SP07 edit*, a list of shaped pulse parameters will appear as displayed in table 2.4.

Table 2.4

parameters for shaped pulses				
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	..	..	..	..

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 Shape Tool (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 table 2.5)

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

## SOLVENT - the sample solvent

- takes a character string value

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

Table 2.5

- 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
- interpreted by **getprosol** and ICON-NMR 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 probehead and the solvent. The command **getprosol** interprets SOLVENT and PROBEHD and sets all dependent parameters accordingly. These parameters must be defined once, with **edprosol**, for all probeheads and solvents. After **getprosol** has been done, the **eda** parameter PROSOL is set to TRUE. In ICON-NMR 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 **utilities** → **sw-sfo1**
- 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  $\mu$ sec.

When you enter a certain value for SW, you may notice that it is slightly ad-

justed by XWIN-NMR. 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 table 2.6 and 2.7).

SW can also be set interactively by clicking the *sw-sfo1* button in the 1D *utilities* menu. This 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 dimension. In the indirect dimensions, 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	Sharp	Medium
DIGTYP		
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
IADC	25	100

**Table 2.6** Maximum SW (kHz) for AQ\_mod = DQD

SWH - spectral width in Hz

- takes a double value

DSPFIRM	Sharp	Medium	Smooth
DIFTYP			
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

- can be set from **eda** or by entering **swh** on the command line
- is related to SW according to the following equation:

$$\text{SWH} = \text{SW} * \text{SFO1}$$

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 2.6 and 2.7.

Note that SWH is stored in the parameter file `acq` 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 switchbox 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:

$$\text{FIDRES}=\text{SW}*\text{SFO1}/\text{TD}$$

In a 2D experiment, TD in the acquisition dimension (F2) has the same meaning as in 1D. In the indirect dimension (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 dimension (F3) has the same meaning as in 1D. In the indirect dimensions (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
```

In XWIN-NMR 3.0 and newer, the statement `lo to n times td1` can be replaced by the `mc` statement. Do not confuse this with the XWIN-NMR processing command `mc` that performs magnitude calculation.

TD0 - loop counter for multidimensional experiments

- takes an integer value
- interpreted by the pulse program statement `td0`
- 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. 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. This list contains the entry **EDIT CURRENT** that allows you to edit the currently defined amplitude list.
- can also be set by entering **valist** on the command line
- interpreted by user defined pulse program statements
- The parameter `VALIST` defines the 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}
```

Note that the second definition does not require a list file. For more information on using variable amplitude lists click:

***Help*** → ***Other topics*** → ***Writing pulse programs***

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. This list contains the entry **EDIT CURRENT** that allows you to edit the currently defined counter 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** → **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. This list contains the entry *EDIT CURRENT* that allows you to edit the currently defined delay 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 the name of variable delay list. Such a list can be created with **edlist**  $\rightarrow$  **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. This list contains the entry *EDIT CURRENT* that allows you to edit the currently defined pulse list.
- can also be set by entering **vpulist** on the command line
- interpreted by the pulse program statements `vp`, `ivp` and `vpidx`
- The parameter VDLIST defines the name of variable pulse list. Such a list can be created with **edlist**  $\rightarrow$  **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* → *Other topics* → *Writing pulse programs*

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. This list also contains the entry *EDIT CURRENT* which allows you to edit the currently defined temperature 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 the name of variable temperature list. Such a list can be created with **edlist** → **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

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)
- interpreted by **wobb**
- The parameter WBST determines the number of steps (frequencies) used for tuning and matching a probehead (wobbling). A probehead 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. 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 <sup>1</sup>.

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 **Acquire** → **Observe fid window** → **wobb-SW**
- interpreted by **wobb**
- The parameter WBSW sets the frequency range for tuning and matching a probehead (wobbling). The center of the wobble region is determined by SFO1. The command **atma** automatically calculates the optimum sweep width and does not interpret WBSW <sup>1</sup>.

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

---

1. During tuning/matching **atma** temporarily sets the parameters WBST and WBSW to the calculated values and then resets them to their original values.

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:

```
<du>/data/<user>/nmr/<name>/<expno>/
```

```
  acquus - acquisition status parameters
```

and can be viewed with **dpa**. 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 **ls 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 set 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 2.4).

#### AQSEQ - 3D acquisition order

- only used in 3D datasets
- takes one of the values `321`, `312`
- can be viewed with `dpa` or by entering `!s aqseq` on the command line
- interpreted by the processing command `tf3`
- AQSEQ describes the order in which the 3 dimensions have been acquired; F3-F2-F1 or F3-F1-F2. AQSEQ is automatically set according to the loop structure in the pulse program. 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 a command which processes the raw data, usually `tf3`. The output data of this command are always in the order F3-F2-F1. If the acquisition status parameter AQSEQ is not set, `tf3` evaluates the processing parameter AQORDER to determine the acquisition order.

#### 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 which work on the raw data like `em` or `ft` interpret the exact value of DECIM to account for the group delay. 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 2.4).

#### EXP - experiment performed

- takes a character array value

- ICON-NMR sets EXP to the value of the parameter set that was used for the experiment.

**FnMODE** - Acquisition mode in the indirect dimensions (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 **1s 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.

The parameter FnMODE is available in XWIN-NMR 3.0 and newer. It is interpreted by processing commands to determine the Fourier transform mode in the indirect dimension(s). In XWIN-NMR 2.6 and older the processing parameter MC2 is used for that purpose. Note that in XWIN-NMR 3.0 and newer, MC2 is still available. It is interpreted when the acquisition status parameter FnMODE has the value *undefined*.

**DSPFVS** - DSP firmware version

- takes an integer value between 11 and 15
- 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. DSPFVS, together with DECIM, is interpreted to account for the group delay of digitally filtered data. DSPFVS must also be interpreted by third party software which processes digitally filtered Avance data.

**LGAIN** - loop gain; lock regulator gain

- is set to a value between -80 and 0 dB
- can be viewed with **dpa** or by entering **1s 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. Note in this respect that the current value of loop gain can also be changed from the BSMS key-



board (by pressing the two keys indicated with *MENU*) or by the command ***lgain***.

LOCSHFT - lock shift

- takes one of the values *true* or *false*
- can be viewed with ***dpa*** or by entering ***ls locshft***
- The value of LOCSHFT indicates whether or not the sample was locked during the acquisition.

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 ***ls 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 ***ls 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 ***ls 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 ***ls masr***
- set by ***masr get***
- The acquisition status parameters MASR is continuously updated when the MAS monitor (started with ***masrmon***) is active. This, however, requires ***masrmon*** to be started from the current dataset. See also the description of MASR in chapter 2.4.

NC - normalization constant

- takes an integer value
- can be viewed with **dpa** or by entering **1s nc** (1D), or **2s nc** (2D)
- set by acquisition commands and by the processing commands **genfid**, **genser** and **addfid**
- interpreted by all processing commands that access raw data
- On Avance spectrometers, acquisition commands set NC to minus the value of DDR. 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 - probehead

- takes a character string value
- can be viewed with **dpa** or by entering **1s probhd** on the command line
- PROBHD is set at the end of an acquisition to the current probehead as it was defined with **edhead** before the acquisition.
- LOCNUC

#### HOLDER - sample changer holder position

- takes an integer value
- can be viewed by entering **1s holder**
- set by ICON-NMR

#### 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 **1s autopos**
- set by ICON-NMR

#### LOCKED - lock status during acquisition

- takes one of the values TRUE or FALSE
- can be viewed by entering **1s locked** on the command line
- The status parameter LOCKED indicates whether or not the magnetic field was locked during the entire acquisition

---

## 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

ect.

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

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

ect.

Note that the command *edsp* reads the values for OF\* that were stored by the previous *edsp* or *edasp* command. If 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

### Two switches to change the default routing

These parameters appear in the **edasp** dialog box under their descriptive names. Their two possible values are offered as radio buttons. They are not stored under the current dataset but in the file `specpar`.

DEFRSEL - Preferred preamplifier

takes one of the values selective or unselective

DEF19F - Preferred output for 19F

takes one of the values 19F or X

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

```
<du>/data/<user>/nmr/<name>/<expno>/
```

`acqus` - acquisition parameters

and the dataset independent file:

```
<xwhome>/conf/instr/<instrum>
```

`specpar` - **edasp** parameters

Note that the routing table can also be opened from the **eda** dialog box, by clicking the NUCLEI 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 (H<sub>0</sub>)

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 probehead and solvent dependent. They are stored for each probehead and for each solvent separately in:

```
<xwhome>/conf/instr/<instrum>/prosol/<probeID>/<solvent>/
```

bsmspar - probehead dependent lock parameters

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

```
<xwhome>/conf/instr/
```

2Hlock - probehead independent lock parameters



---

# Chapter 3

## Spectrometer configuration commands

---

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

## cf

### NAME

cf - configure the spectrometer

### DESCRIPTION

The command **cf** allows you to configure the spectrometer. It can be executed as a single command or as a part of the XWIN-NMR configuration suite (command **config**).

**cf** must be executed after installing a new version of the NMR Suite on a spectrometer. Furthermore, it is useful to run **cf** in case of spectrometer communication problems.

It prompts you for the NMR Superuser password and then for various information about your spectrometer. The type of questions depend on whether you perform a first time or a repeated configuration.

#### First time configuration:

**cf** will prompt you for the following information:

*Enter new instrument name:*

Enter a name for your spectrometer. The default name is *spect*. Note that the name that you enter is the hostname of the spectrometer internal CPU. As such, it must be different from the hostname of the PC or Workstation that controls the spectrometer. If you choose a hostname different from *spect*, you must specify this name in the `hosts` file (see below).

*Avance AMX ARX ASX Datastation:*

Enter your type of spectrometer or enter *Datastation*.

*What type of Avance?*

This question only appears if you entered *Avance* in the previous question and **cf** detected that you have an Avance-AQX. Enter your type of Avance: *DMX*, *DRX*, *DPX* or *DSX*.

*Basic 1H frequency (with offset OI=0) in MHz:*

Enter the 1H 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.

After you have answered these questions, a list of all possible external devices will appear. For each existing device, you have to specify the serial port to which it is connected. Non-existing devices must be specified with the value *no*. Furthermore, you can set the parameters POWCHK and LOCSWID here. The former sets the automatic peak power limitation to true or false. The latter specifies the router output channel of the amplifier to which the lock switch is connected (0 if the lock switch does not exist).

If your spectrometer is equipped with a sample changer and/or an MAS unit, **cf** will prompt you for some information about these units. Internally the commands **cfbacs** and/or **cfmas** are executed, respectively (see these commands for more information).

Next, the routing table will appear. Here, you can only set the connections between Switchbox output and Preamplifier modules, or, if there is no Switchbox, between the Amplifiers and Preamplifier modules. If the routing table shows invalid routing lines, remove them by clicking **CLEAR ALL**. Then define new lines according to you hardware cable connections and click **SAVE**.

Next, nuclei table will appear. Normally you can store this table as it is by clicking **SAVE**. If, however, you want to change the nuclei table, you can do that here. Note that you can also do that at a later time with the command **ed-nuc**.

Finally, an overview over the spectrometer configuration is shown. 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.

A first time configuration is a configuration that is, for example, done on a new spectrometer, a new computer, a new disk or a new XWIN-NMR home directory. If, however, a new version of XWIN-NMR is installed in the same directory as a previous version, the configuration is still available. **cf** then runs a repeated configuration.

### **Repeated configuration:**

**cf** will prompt you for the following information:

*Configuration for <name>, change that (y/n) ?*

where *name* is the spectrometer name which was specified the last time **cf** was executed. Hit **Enter** to keep this name or enter a new name.

All other questions are the same as the first time you ran **cf**. If the spectrometer hardware has not changed, you can answer each question with **Enter**.

After you have answered these questions, the list of all external devices will appear. The upper part of the window shows the devices that were assigned a tty number the last time **cf** was executed, i.e. the device that exist. The lower part (other settings) shows the devices that were specified with *no* during the last **cf**, i.e. the devices that do not exist. Normally, you can click **Save** to keep these settings unless your hardware has changed or you want to change POWCHK (for example to run **cortab**).

Next the nuclei table will appear which can be closed with **SAVE**. If, however, you have changed the basic frequency, you must first click **RESTORE**.

Next, the routing table will appear which can be closed by clicking **CANCEL**. If, however, any hardware cable connections between amplifiers and the HPPR preamplifier have changed, you must make the corresponding changes in the routing table and close it with **SAVE**. Note that this step in **cf** corresponds to the command **edasp setpreamp**.

### Hosts file

If, in the first step of **cf**, you specify an instrument name different from *spect*, this name must be specified in the `hosts` file. This file can be edited from the Windows Explorer or from a UNIX shell. It concerns the file:

`/etc/hosts` (under UNIX)

`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 drx500
```

### Hardware list

Avance-AQS spectrometers do not need a hardware list. Most Avance-AQX spectrometers, however, do need a hardware list which is interpreted by **cf** during a first time configuration. The hardware list is normally created during the installation of the spectrometer by the service engineer.

We strongly recommend to save the spectrometer configuration by copying the files `uxnmr.par` and, if it exists, `hardware_list` to floppy disk or CD and printing them out.

## INPUT FILES

`<xwhome>/exp/stan/nmr/form/`

`acqu.e.D` - acquisition parameter format file (**eda**) for Avance

## OUTPUT FILES

(also INPUT files for repeated configuration)

`<xwhome>/conf/instr/`

`curinst`

`<xwhome>/conf/instr/<instrum>`

`uxnmr.info` - spectrometer configuration overview

`uxnmr.par` - spectrometer configuration

`nuclei` - nuclei table

`<xwhome>/conf/instr/<curinst>/rs232_device`

`acb` - Amplifier control board rs232 channel

`bacs` - Sample changer rs232 channel

`bsms` - BSMS rs232 channel

`lock` - lock display rs232 channel

`preemp` - preemphasis rs232 channel

`preamp1` - HPPR preamplifier rs232 channel

`preamp2` - second HPPR preamplifier rs232 channel <sup>1</sup>

`preamp3` - third HPPR preamplifier rs232 channel <sup>1</sup>

`rx22` - rx22 or rxc receiver unit rs232 channel

`temp` - temperature unit rs232 channel

## SEE ALSO

`config`, `expinstall`, `cfe`, `cfbacs`, `cfmas`, `cfbpsu`, `cfmas`

---

1. For multiple RCU spectrometers only.

## cfbacs

### NAME

cfbacs - configure sample changer

### DESCRIPTION

The command **cfbacs** allows you to configure the sample changer. It first prompts you for the rs232 channel. Then it will ask you the following questions:

*Should the sample changer control the lift?*

with *no* as default answer.

*Delay between SX and the next command [sec]?*

where SX refers to the sample change. This delay is required for the sample to settle in the magnet. The first command after sample change is typically **ro** to switch the sample rotation on. The minimum value depends on the magnet type and lies, typically, between 10 and 30 seconds.

The command **cfbacs** can also be used if your spectrometer has no sample changer. This is, for example, useful for the automation setup with ICON-NMR on a separate workstation.

**cfbacs** can also be executed as a part of the XWIN-NMR configuration suite (command **config**). If your spectrometer uses a hardware list, and the sample changer is specified in that list, **cfbacs** can be skipped. The configuration of the sample changer is then executed by the command **cf** (see this command for more information on the hardware list).

### INPUT AND OUTPUT FILES

<xwhome>/conf/instr/<instrum>/rs232\_device/

bacs - serial port for sample changer unit

<xwhome>/conf/instr/<instrum>/

bacs\_params - number of holders in the sample changer

### SEE ALSO

config, cf, cfte, cfmas, cfbpsu

## **cfbpsu**

### **NAME**

cfbpsu - configure BPSU unit

### **DESCRIPTION**

The command **cfbpsu** allows set the rs232 channel for the BPSU unit. This unit is used for LC-NMR experiments.

**cfbpsu** can also be executed as a part of the XWIN-NMR configuration suite (command **config**).

### **INPUT AND OUTPUT FILES**

<xwhome>/conf/instr/<instrum>/rs232\_device/

bpsu - serial port for BPSU unit

### **SEE ALSO**

config, cf, cfte, cfmas, cfback

## **cftc**

### **NAME**

cftc - configure the temperature unit

### **DESCRIPTION**

The command **cftc** allows you to set the rs232 channel port for the temperature unit.

**cftc** can also be executed as a part of the XWIN-NMR configuration suite (command **config**).

### **INPUT AND OUTPUT FILES**

<xwhome>/conf/instr/<instrum>/rs232\_device/

temp - rs232 channel for the temperature unit

### **SEE ALSO**

config, cf, cfback, cfmas, cfbpsu

# compileall

## NAME

compileall - compile Bruker and User AU programs

## DESCRIPTION

The command ***compileall*** compiles all Bruker and User defined AU programs. In order to compile Bruker AU programs, these must be installed first. This can be done with the command ***expinstall***, with the option "Install Bruker library AU programs/modules" enabled.

For more information on AU programs please refer to the AU reference manual.

## INPUT FILES

<xwhome>/exp/stan/nmr/au/src/\*

AU programs (source files)

## OUTPUT FILES

<xwhome>/prog/au/bin/\*

AU programs (executable binaries)

## SEE ALSO

expinstall, cplbruk, cpluser, edau, xau, xaua, xaup, delau, renau

# config

## NAME

config - start the XWIN-NMR configuration suite

## DESCRIPTION

The command **config** starts the XWIN-NMR configuration suite. It opens a dialog box which contains a list configuration steps. If this dialog box does not appear, it is probably iconified. Just click its icon to open it. The **config** dialog box shows the following entries:

Spectrometer configuration	[cf]
Temperature unit configuration	[cfte]
Sample changer configuration	[cfbacs]
Solvent table setup	[edsolv]
Probehead table setup	[edhead]
Solvent dependent parameter setup	[edprosol]
Lock parameter setup	[locnuc,edlock]
Configure MAS pneumatic unit	[cfmas]
Configure BPSU unit for LC-NMR	[cfbpsu]
Avance spectrometer constants	[edscon]
Avance frequency routing	[edsp]
Installation of standard experiments	[expinstall]

For each entry the corresponding XWIN-NMR command is shown between square brackets. In order to start the configuration, do the following:

1. Check the entries which correspond to your spectrometer and uncheck the others. Note that [cf] should always be checked, even on a datastation. In case of doubt, you can just leave the default selection.
2. Click the **Start** button. The first checked command is executed. Most commands prompt the user for input.

Note that the dialog box is sometimes HIDDEN behind the config win-



dow!

Wait until the command has finished.

3. Click the *Continue* button to execute the next command.
4. Click the *Continue* button to execute the next command.
5. ect.
6. When all checked commands have been executed, click *Quit* to leave the configuration suite.

The button *Cancel* allows you to start the entire configuration suite from the beginning, i.e. to rerun all checked commands.

## INPUT FILES

<xwhome>/prog/tcl/xwish3-scripts/

config - config Tcl/Tk source file

see also the OUTPUT FILES of the individual commands

## OUTPUT FILES

see the OUTPUT FILES of the individual commands

## SEE ALSO

cf, cfte, cfbacs, edsolv, edhead, edprosol, locnuc, edlock, cfmas, cfbpsu, edscon, edsp, expinstall

## cortab

### NAME

cortab - correction table creation

### DESCRIPTION

The command **cortab** opens a window from which amplifier correction tables for acquisition can be created. These tables are used to correct the non-linearity of the pulse power level versus the pulse length. The amplifier linearization is performed in a range from -6 to above 75 dB and involves both amplitude linearization and phase correction. It requires 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. An example of a correction table is shown in table 3.1.

Once a correction table exists, it is automatically used by acquisition commands like **zg**. 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.

Please note that correction tables generated with Xwin-nmr 3.5 or newer cannot be used in older versions of Xwin-nmr, whereas older correction tables can be used with Xwin-nmr 3.5 or newer.

On Avance-AQS spectrometers, amplifiers abbreviated 'BLA' and the multiplication unit of the Signal Generation Unit ('SGU MULT') require a Cor-tab correction.

On Avance-AQX spectrometers, amplifiers and ASUs (Amplitude Setting Unit) amplifiers require Cortab correction. In contrast to the SGU mentioned above, both the multiplication unit ('ASU MULT') and the modulation unit ('ASU MOD') are affected.

Because the ASU (or SGU) appears earlier in the RF path, its correction table must be created first. The result of the ASU/SGU correction is then automatically used for the amplifier correction.

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

**Table 3.1**

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, the preamplifier must not be used when a high power amplifier (> 500 W) is linearized.

The **cortab** window allows you to setup the linearization experiments for various nuclei, perform these experiments and view the results. A regular Cortab session involves the steps listed below. Each step is specified by the corresponding button in the main Cortab window.

### ***Experiment Setup***

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.

On Avance-AQS spectrometers, the following functions are available:

- BLA Linearization + Verification
- SGU MULT Correction + Verification
- BLA Verification
- SGU MULT Verification

On Avance-AQX spectrometers, Amplifier (BLA) and ASU (MULT and MOD) functions are available.

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

Please note that the Cortab correction is mainly determined by the BLA linearization (see table 3.1). However, it is strongly recommended to perform the ASU/SGU linearization first in order to achieve optimum results.

If you have a Router board and you want to perform a MOD or MULT correction experiment, it is recommended that you connect the respective router output (e.g. RO1) to the external attenuator rather than the ASU/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 Results*** be-

low). 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.

- **Add Experiment**

Opens the routing table (like *edasp*) where you can determine the appropriate hardware routing for the chosen experiment. Please note that even if you want to perform an ASU/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 because otherwise you may seriously damage 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 ASU/SGU correction and a BLA linearization in one series because they require different hardware connections.

After having scheduled the experiment you are prompted for the real amplifier output power at -6dB. This value corresponds to the power (in Watt) measured at the output of the respective HPPR preamplifier module when you are pulsing with -6dB 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. The amplifier output value is crucial for any kind of output power limitation (e.g. to reduce the possibility of probehead damage) of the acquisition software. However, if the respective power value is not known, you can either use zero instead (ignoring the power limitation possibilities of the acquisition software) or measure it with a wattmeter.

- **Return**

Closes the dialog box and returns to the main Cortab window

### ***Start Experiment***

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 if the automatic peak power limitation is active, the Cortab experiment might be aborted with the error message: "power level (..) to large for selected probe channel (..) ". If this happens, you must, temporarily, switch off the peak power limitation in order to run Cortab. To do that, run **cf** setting POWCHK to FALSE. After Cortab has finished, run **cf** again setting POWCHK to TRUE.

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 ASU/SGU unit is scaled to the correction table of the 'weakest' ASU/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' ASUs/SGUs. The scaled correction table files have the extension '.scale' and, as such, they are inactive. They can be activated by removing the file extension.

Each linearization experiment involves the following steps:

- A receiver gain test 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 Results** button as described below.

**View Results**

This button is only active when an experiment is selected in the *Selection Desktop* of the Cortab window. You can select an experiment by clicking an element in the fields *Nucleus*, *Channel* and *Amplifier*. Clicking **View Results** 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. For example:

Amp\_Power\_res - theoretical and experim. amplifier output voltage  
Amp\_Phase\_res - theoretical and experim. amplifier phase values

When you click an entry, the contents of the corresponding file are shown. Note that files like Amp\_Power\_res do not contain the correction values themselves 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 more easy 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.

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:

- **Graphic window**  
opens a window with graphical representation of the Cortab result.
- **Back to main window**  
Closes the 'View Results' window and returns to the main Cortab window

### **Scheduled series**

Clicking this button opens a dialog box where the currently scheduled exper-

iments 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
- **Exit:** exits the dialog box and returns to the main Cortab window

### *Nucleus List refresh*

Clicking this button updates the nucleus list adding experiments that have been finished since Cortab has been started.

Cortab experiments use two spectrometer channels simultaneously:

- the pulse channel: selected by the user in the Cortab *Experiment Setup*, i.e. the channel that is routed to the amplifier (or ASU/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.

The selection of the observe channel is transparent to the user. An exception to this is the Avance-DPX spectrometer with ASU2/LOT. Here, only the f1 channel can be selected for observation. Cortab experiments on f1 (using f2 as the observe channel) require the following hardware modification:

- Remove the SMA connector of the cable which connects the ASU2/LOT FTUNE output to the TUNING IN of the preamplifier cover module.
- Connect the RX 22 LO J2 input SMA connector to the ASU2/LOT FTUNE output instead of the ASU2/LOT LO1 output.

The AVANCE DPX spectrometer is now in a hardware configuration which allows Channel f1 pulsing with Channel f2 observation. The configuration used here is only suitable for Channel f1 pulsing with observation of channel 2. For all other experiments (e.g. Hardware tests or NMR observation) you have to switch back to the Channel f1 observation configuration described below:

Channel f1 observation = default connections :

- The SMA connector of the cable which connects the ASU2/LOT FTUNE output to the TUNING IN of the preamplifier cover module has to be connected on the ASU2/LOT FTUNE output ⇒ Only f2 pulsing and wobbling operations are possible on Channel f2



- Connect the RX 22 LO J2 input SMA connector to the ASU2/LOT LO1 output instead of the ASU2/LOT FTUNE output.

## INPUT FILES

<xwhome>/prog/app-defaults/

Cortab - cortab window resources

<xwhome>/exp/stan/nmr/lists/pp.hwt

zg\_ctb - cortab pulse program

<xwhome>/exp/stan/nmr/par.avance

XCTB - cortab parameter set

<xwhome>/QTP/

Date.tmp - time and date of last *Hardware configuration*

functions.AV - functions for Avance-AQS (input of *Exper. Setup*)

functions.DRX - functions for Avance-AQX (input of *Exper. Setup*)

## OUTPUT FILES

### On Avance-AQX and Avance-AQS spectrometers

<xwhome>/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

amp<fcu no.>\_<nucl.>\_<ampl. RO>.raw - raw data files; for internal use only

amp<fcu no.>\_<nucl.>\_<ampl. RO>.scale - correction table standardized to 'weakest' ASU/SGU

amp\_table - table where amplifier output power is stored; for internal use only

mma<fcu no.>\_<nucl.>\_<ampl. RO> - ASU/SGU correction tables (MULT + MOD)

<xwhome>/QTP/<nucleus>/FCU<no>/<ampl. descr.>/<date>/

Amp\_Power\_res - theoretical and experimental amplifier output power  
Amp\_Phase\_res - experimental amplifier phase values  
Mult\_Power\_res - theoretical and experimental ASU/SGU output voltage (multiplier unit)  
Mult\_Phase\_res - experimental ASU/SGU phase values (multiplier unit)

**On Avance-AQX spectrometers only**

<xwhome>/qtp/<nucleus>/FCU<no>/<ampl. descr.>/<date>/

Mod\_Power\_res - theoretical and experimental ASU output voltage (modulator unit)

**SEE ALSO**

zg

## cplbruk, cpluser

### NAME

cplbruk - compile Bruker AU programs  
cpluser - compile user defined AU programs

### SYNTAX

cplbruk [<name> | all ]  
cpluser [<name> | all ]

### DESCRIPTION

The command **cplbruk** allows you to compile one or more Bruker AU programs. Before you can use it, the command **expinstall** must have been executed once, with the option "Install Bruker library AU programs/modules" enabled. Then you can use **cplbruk** in three different ways:

**cplbruk <name>** - compile the Bruker AU program <name>  
**cplbruk all** - compile all Bruker AU programs  
**cplbruk** - a list of Bruker AU programs appears; click one to compile it

If you specify an argument, then it may contain wildcards; for example **cplbruk a\*** compiles all Bruker AU programs which start with *a*.

The command **cpluser** works like **cplbruk**, except that it compiles user defined AU programs.

For more information on AU programs please refer to the AU reference manual.

### INPUT FILES

<xwhome>/exp/stan/nmr/au/src/\*

AU programs (source files)

### OUTPUT FILES

<xwhome>/prog/au/bin/\*

AU programs (executable files)

**SEE ALSO**

expinstall, compileall, edau, xau, xaua, xaup, delau, renau

## ednuc

### NAME

ednuc - edit the nuclei table

### DESCRIPTION

The command **ednuc** opens the nuclei table. This table consists of two columns; the left column shows a list a nuclei, the right column the corresponding basic frequency.

Within the nuclei table, you can perform the following actions:

- Change the frequency of a nucleus by entering a new number in the right column.
- Delete a nucleus by clicking its entry in the left column
- Click ADD to add a nucleus. A list of all nuclei will appear and you can click the one you want to add.
- Click RESTORE to 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**.
- Click SAVE to close the nuclei table, saving all changes.
- Click QUIT to close the nuclei table, discarding all changes

### INPUT FILES

<xwhome>/conf/instr/<instrum>/

nuclei - nuclei table

<xwhome>/exp/stan/nmr/lists/

nuclei.all - complete nuclei table (input of ADD and RESTORE)

### OUTPUT FILES

<xwhome>/conf/instr/<instrum>/

nuclei - nuclei table

**SEE ALSO**

edsolv, edhead, edlock, edprosol

# edprosol

## NAME

edprosol - edit probehead and solvent dependent parameters

## DESCRIPTION

The command **edprosol** opens a dialog box in which you can set the probehead and solvent dependent (prosol) parameters. This is typically done during the installation of your spectrometer for all probeheads and solvents you want to use. However, you can always run **edprosol** again at a later time to set the prosol parameters for additional probeheads and/or solvents.

Setting the prosol parameters involves the following steps:

1. Select the probehead, solvent(s) and nucleus:
  - **Probe name**  
By default, the current probehead (as defined with **edhead**) is selected. Prosol parameters must be defined for each probehead 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.
2. 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.).
3. 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
  - **F3** : prosol parameters for the f3 frequency channel
  - **Global** : a few prosol parameters that count for all frequency channelsBy default, **F1+F2** is checked. Most experiments only use channel f1 or f1

and f2. If you are using f3, you have to set the prosol parameters for this channel as well. Note that f1, f2 and f3 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 dimensions of a 2D or 3D dataset.

4. 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.

5. 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 determine 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. Note that the relation used by ***calc*** is determined by the 90° hard pulse.
6. 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***  
opens a list of all probeheads. You can select the probehead(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***.
- ***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 UNIX shell.

The **edprosol** window allows you to set up a probehead dependent tune file. You can do that by clicking **File** → **Edit tune file**. On first time execution, this command displays the example tune file that is delivered with XWIN-NMR. When you save the file, it is stored for the probehead that is currently defined in **edprosol**. Probehead 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 7.1. Note that **getprosol** is automatically performed as part of the ICON-NMR automation.

**edprosol** can also be executed as a part of the XWIN-NMR configuration suite (command **config**).

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

The command **edprosol** replaces the commands **prosol**, **solvloop** and **liprosol** which were used in XWIN-NMR versions 2.6 and older.

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/group/

example\_bsms - example tune file (input for first **File** → **Edit tune file**)

<xwhome>/prog/tcl/xwish3\_scripts

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

<xwhome>/prog/tcl/libtix/prosol/lib/lists

routing - default spectrometer routing

## INPUT AND OUTPUT FILES

If you select Solvent(s) *All*:

<xwhome>/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:

<xwhome>/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, edlock

## edscon

### NAME

edscon - edit spectrometer constants

### DESCRIPTION

The command **edscon** opens a dialog window, where you can set certain spectrometer parameters (constants). 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. It can be entered from the command line or it can be executed as part of the **config** suite.

For Avance-AQX spectrometers, the upper part of the **edscon** window offers the following parameters:

- BLKTR[1-8] - ASU and transmitter blanking preset time (default 3 $\mu$ )
- BLKPA[1-5] - preamplifier blanking preset time (default 3 $\mu$ )
- PHASPR[1-8] - phase switching preset time (default 3 $\mu$ )
- SHAPPR[1-8] - shaped pulse calculation preset time (default 1.6 $\mu$ )
- PHASP4[1-8] - 4PH modulator phase switching preset time (default 0.5 $\mu$ )

They specify the time between a blanking or switching step and the beginning of the RF pulse. As such they are called preset parameters. For Avance-AQS, only BLKTR appears.

BLKTR and BLKPA are blanking preset times. The principle of blanking is used in the ASU (Amplitude Setting Unit), transmitter and preamplifier. This means they only allow RF signal 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 ASU and transmitter are blanked BLKTR  $\mu$ sec before the pulse, the preamplifier is blanked BLKPA  $\mu$ sec before the pulse. All three units are unblanked (allow no further RF passing) at the end of the pulse.

PHASPR and PHASP4 are phase switch preset times. In many pulse programs, the phase of the RF pulse is switched after every scan, using a phase program (list). Because of the finite switching time, the phase switch is triggered before the start of the RF pulse. When a regular phase list like

```
ph1 = 0 90 180 270
```

is used, the phase is switched PHASPR  $\mu$ sec before the pulse. If, however, your

spectrometer is equipped with a 4-phase modulator, and a phase list like

```
ph1 = +x +y -x -y
```

is used, the phase is switched PHASP4  $\mu\text{sec}$  before the pulse. Note that the default value for PHASP4 (0.5 $\mu$ ) is much shorter than that for PHASPR (3 $\mu$ ). The reason is that the switching of phases created by the 4-phase modulator is much faster than that of regular phases.

SHAPPR is a shaped pulse preset time. A shaped pulse requires a short preparation time, for example to reset a pointer to the beginning of the list of values that defines the shape. Note that after each shaped pulse, the pulse program must contain a minimum delay of 2  $\mu\text{sec}$  on the respective channel.

Each preset parameter is an array of 16 elements but only elements 1 - 8 are used, corresponding to the maximum of 8 spectrometer channels. For BLKPA, only elements 1 - 5 are used, corresponding to the 5 preamplifier modules. For most high resolution experiments, you do not have to change the preset parameters and you can work with the default values that have been set during the installation of your spectrometer. However, for some solids state experiments, it is often useful to use shorter values. We recommend the following settings:

- CRAMPS experiments (with f1 as the 1H observe channel):  
 BLKTR[1] = 0.7  $\mu\text{s}$  for DMX  
 BLKTR[2] = 0.7  $\mu\text{s}$  for DRX  
 PHASPR[1] = 0.6  $\mu\text{s}$  for DMX with TFH board (see [uxnmr.info](#))
- CP experiments:  
 PHASPR[2] = 1.7  $\mu\text{s}$   
 PHASPR[2] = 0.6  $\mu\text{s}$  for DMX with TFH board (see [uxnmr.info](#))
- Back to back pulses:  
 PHASPR[n] = 0.6  $\mu\text{s}$  Avance-AQX with TFH board (see [uxnmr.info](#))  
 PHASPR[n] = 1.7  $\mu\text{s}$  for DMX with TFX or DRX

For more information on this topic, click **Help** → **Other topics** → **Solids Users Manual**

The use of the **edscon** preset parameters 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 **edscon** preset parameters to zero. In this case, the blanking and switching 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 ASU, 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) but is automatically introduced by the `go` statement.

On Avance-AQX spectrometers, DE consists of 5 pre-scan subdelays DE1, DE2, DERX, DEADC and DEPA that all start simultaneously at the beginning of DE. 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`.

DE2: the phase of the receiver is set to zero (default 1 μsec)

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. On Avance-AQS, the DE2 does not exist.

In most pulse programs, data sampling is performed by the `go` statement that 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* → *Other topics* → *Writing pulse programs*

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**

<xwhome>/conf/instr/<instrum>/

scon - spectrometer constants

## **SEE ALSO**

cf, config

## edsolv

### NAME

edsolv - edit solvent table

### DESCRIPTION

The command **edsolv** opens the solvent table. This table contains one line for each solvent and shows the solvent name, a short description and a reference number. A typical line looks like:

*Acetic* - *Acetic-Acid-D4* [02]

Furthermore, the solvent table contains the following buttons:

#### *Add/Change*

Appends an empty line in which you can add a new solvent. Please enter the solvent name, a short description (optionally) and a unique reference number. All existing lines are put in *edit* mode and can be changed. Make sure that each reference number is only used once.

#### *Delete*

Puts all lines in *delete* mode. If you click a entry now, it will be deleted.

#### *Save*

Closes the table, saving all changes.

#### *Abort*

Closes the table, discarding all changes.

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.

**edsolv** can also be executed as a part of the XWIN-NMR configuration suite (command **config**).

### INPUT FILES

<xwhome>/exp/stan/nmr/lists/

`solvents.all` - complete Bruker solvent list (input for first **edsolv**)

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

### **OUTPUT FILES**

`<xwhome>/exp/stan/nmr/lists/`

`solvents` - user solvent list

### **SEE ALSO**

`ednuc`, `edhead`, `edlock`, `edprosol`



# 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** and **edasp** differ in only respect:

- **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	<b>5000.00</b> 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	<b>5000.00</b> 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 XWIN-NMR. In this case, you only need to set the following switches:

*Preferred preamplifier*

toggle between selective and unselective Preamplifier module

*Preferred output for 19F*

toggle between the 19F and X Switchbox output for 19F nucleus

and click **SAVE** to store them.

The *Preferred preamplifier* determines the default preamplifier module (selective or unselective) for X-nuclei. The *Preferred output for 19F* determines the default output of the switchbox (X or 19F) for 19F.

**edsp** should be executed before **expinstall** which installs the standard parameter sets. **expinstall** reads the values of Preferred preamplifier and Preferred output for 19F, as they were set by **edsp**, and stores the default routing parameters accordingly.

## INPUT AND OUTPUT PARAMETERS

see **edasp**

## INPUT AND OUTPUT FILES

see **edasp**

## SEE ALSO

edasp, config, expinstall

# 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. It must be performed once after the installation of XWIN-NMR and after **cf** has been done. **cf** and **expinstall** are typically performed as a part of the **config** configuration suite.

**expinstall** first prompts you for the NMR Superuser password. After it has been entered correctly, you see a list of spectrometer types. The spectrometer type you have defined with **cf** is highlighted. You can simply click **Proceed**, unless you have a reason to choose a different type of spectrometer. **expinstall** will then offer you a list of tasks which can be selected or deselected. For routine spectroscopy, you can accept the default selection and click **Proceed** to continue. If the task *Convert Standard Parameter Sets* was selected, you will be prompted for some information about the spectrometer. Once this has been entered, **expinstall** will start to execute the selected tasks. The full list of tasks is:

- Install pulse programs  
These are used for all experiments.
- Install Bruker library AU programs  
Performing this task makes Bruker AU programs available for editing (**edau**) and compilation (**xau** or **cplbruk**).
- Recompile User AU programs  
AU programs must be (re)compiled after the installation of a new XWIN-NMR version because the installation has removed the AU binaries. **expinstall** only compiles User AU programs, not Bruker AU programs. The latter can be compiled with **compileall** or **cplbruk all**.
- Install CPD programs  
These are used for composite pulse decoupling experiments.
- Install gradient files  
These are used for gradient experiments.

- Install Library shape files  
These are used for selective excitation experiments.
- Convert standard parameter sets  
Bruker standard parameter sets, as they are delivered with the NMR Suite, were prepared at various field strengths. **expinstall** converts them to your spectrometer frequency. This includes the parameters BFx, Ox, SFOx and SW as well as the offsets of the shaped pulses (parameter SPOFFS). Note that for 2D (3D), the SW in F1 (F2 and F1) is kept and the increment IN0 (IN0 and IN10) is adjusted.
- Install standard scaling region files  
These contain the regions in which the reference peak for vertical scaling is searched by commands like **plot**, **li**, **lipp**, **pp\***.
- Enable Define Statements in Pulse programs  
Define statements are pulse program statements like:

```
;; d11=30m
```

at the beginning of a pulse program. In this form they are not active because lines starting with a ';' character are treated as comment. **expinstall** removes all occurrences of ';', thereby enabling the defined statements. Normal comment lines are not affected because they contain a single ';' only.

The NMR Suite is delivered with a set of pulse programs, CPD programs, gradient files, shape files and scaling region files for each spectrometer type (Avance, AMX, ARX etc.). **expinstall** installs the set which is needed on your spectrometer type.

If the task *Convert Standard Parameter Sets* is selected, **expinstall** will prompt you for the following information:

- Select type of digitizer :  
Click the digitizer that you want to store in the acquisition parameter DIGTYP in all parameter sets.
- Select acquisition mode :  
Click the acquisition mode that you want to store in the acquisition parameter AQ\_mod in all parameter sets
- Enter default pre-scan-delay DE :  
Enter the value that you want to store in the acquisition parameter DE in all parameter sets. Normally, you can accept the default value.

- Select printer :  
Click the printer that you want to store in the output (**edo**) parameter CURPRIN in all parameter sets. This printer will be used by commands like **lpa** and **lpp**.
- Select plotter :  
Click the printer that you want to store in the output (**edo**) parameter CURPLOT in all parameter sets. This printer will be used by commands like **view** and **plot**.
- Enter paper format:  
Enter A4, A3, A or B. The entered value will effect various plot (**edg**) parameters like CX, CY and DHEI in all parameter sets. If you enter any value other than A4, A3, A or B, the file:

`<xwhome>/exp/stan/nmr/lists/plotconvpar`

is interpreted for the paper format <sup>1</sup>. This file is delivered with XWIN-NMR and you can change it for your purpose from the Windows Explorer or from a UNIX shell.

## INPUT PARAMETERS

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

set by the user with **edsp** :

DEFRSEL - preferred preamplifier (default routing)

DEF19F - preferred output for 19F (default routing)

from the parameter sets as delivered with XWIN-NMR:

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

---

1. Note that the file plotconvpar.A3 is used for A3 and B and plotconvpar.A4 for A4 and A.

## 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

<xhome>/conf/instr/<instrum>/specpar - routing parameters  
 <xhome>/prog/au/src.exam/\* - Bruker AU programs (source files)  
 <xhome>/exp/stan/nmr/au/src/\* - AU programs (source files)  
 <xhome>/exp/stan/nmr/par.avance/\* - Bruker parameter sets for Avance  
 <xhome>/exp/stan/nmr/par.300/\* - Bruker parameter sets for A\*X  
 <xhome>/exp/stan/nmr/pp.dexam/\* - pulse programs for Avance  
 <xhome>/exp/stan/nmr/pp.exam/\* - pulse programs for AMX  
 <xhome>/exp/stan/nmr/cpd.dexam/\* - CPD programs for Avance  
 <xhome>/exp/stan/nmr/cpd.exam/\* - CPD programs for AMX  
 <xhome>/exp/stan/nmr/gp.dexam/\* - gradient programs for Avance

<xwhome>/exp/stan/nmr/gp.exam/\* - gradient programs for AMX

<xwhome>/exp/stan/nmr/wave.dexam/\* - shape files for Avance

<xwhome>/exp/stan/nmr/wave.exam/\* - shape files for AMX

<xwhome>/exp/stan/nmr/scl.exam/\* - scaling region files for Avance/AMX

Depending on the spectrometer type and/or application, expinstall uses various other input folders/files using the extensions:

- .rexam - high resolution on ARX
- .solids - solid state on AMX/ASX
- .imag - micro imaging on AMX
- .tomo - tomography
- .dsolids - solid state on Avance
- .dimag - micro imaging on Avance

## OUTPUT FILES

<xwhome>/exp/stan/nmr/au/src/\* - Bruker AU programs (source files)

<xwhome>/prog/au/bin/\* - AU programs (binary executables)

<xwhome>/exp/stan/nmr/par/\* - parameter sets for your spectrometer

<xwhome>/exp/stan/nmr/pp/\* - pulse programs for your spectrometer

<xwhome>/exp/stan/nmr/cpd/\* - CPD programs for your spectrometer

<xwhome>/exp/stan/nmr/gp/\* - CPD programs for your spectrometer

<xwhome>/exp/stan/nmr/wave/\* - shape files for your spectrometer

<xwhome>/exp/stan/nmr/scl/\* - scaling region files for your spectrometer

## SEE ALSO

cf, config, cplbruk, cpluser, compileall, rpar, wpar





---

# Chapter 4

## Lock commands

---

This chapter describes the commands which are involved in locking the magnetic field. Avance spectrometers use a digital lock that 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  $^2\text{H}$  and  $^{19}\text{F}$ .

# 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. It must be executed once for each probehead and each lock nucleus. The lock parameters can be determined from the BSMS keyboard or from the BSMS display (command **bsmsdisp**).

Before you run **edlock**, you must define the current probehead 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 probehead, a default lock table, which is delivered with XWIN-NMR, is opened.

The first line of the lock table shows the lock nucleus and the current probehead. The second line shows the lock frequency, the field value (H0) and the basic spectrometer frequency. The latter must be specified when the spectrometer is configured with **cf** or **config**. The lock frequency is automatically calculated from 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:

LPower - Lock power; 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 - 1.0 seconds)

LFilt - Loop filter, lock regulator cut-off frequency of the lowpass filter (1 - 200 Hz)

LPhase - Lock phase; the phase of the lock signal

Nucleus - Observe nucleus

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 (*sref*):**

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

At the bottom of the lock table, the following buttons/fields are available:

- SAVE - saves the current values and closes the dialog box
- BSMS-FIELD - reads the field value (H0) from the BSMS unit
- NUCLEUS - selects an observe nucleus for each solvent. This is equivalent to clicking the **Nucl.** button on one solvent and selecting **new**.
- NEW SOLVENT - adds an extra solvent entry to the list. The currently selected (highlighted) entry is copied and the solvent field can be edited. If no entry is highlighted at the time you click NEW SOLVENT, the last entry will be copied.
- DELETE - switches to delete mode. If you click on a solvent now, its entry will be deleted. If you click on the nucleus button of a particular solvent, the currently selected nucleus is deleted for that solvent.
- ABORT - closes the lock table discarding all changes
- LOADSTAN - loads the standard (default) values for all parameters. These are the same values which are loaded when **edlock** is executed for the first time.
- +/- POWER - in this field you can enter the number that should be added to the lock power (*LPower*) of all solvents. A positive number increases the lock power, a negative number decreases it.
- LIST - prints the lock parameters on the device specified by the **edo** parameter CURPRIN
- COPY\_VALUE automatically copies the value of the selected parameter to all solvents

The loop gain, loop time and loop filter can also be set with the XWIN-NMR 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 ICON-NMR automation, the lock parameters are read from the lock table and used as they are.

**edlock** can also be executed as a part of the XWIN-NMR configuration suite (command **config**).

For information on how to determine the lock parameters click **Help** → **Other topics** → **Avance Users Guide**. Furthermore, you can refer to the spectrometer hardware documentation. It can be viewed by inserting the BASH CDROM and clicking **Help** → **Other topics** → **BASH spectrometer documentation**.

## INPUT AND OUTPUT PARAMETERS

see DESCRIPTION above

## INPUT FILES

<xwhome>/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**

<xwhome>/conf/instr/<instrum>/

2Hlock - 2H lock parameters

19Flock - 19F lock parameters

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

bsmspar - solvent and probehead dependent lock parameters

**SEE ALSO**

lock, lopo, lopi, edhead, lockdisp, lgain, ltime, lfilter

## 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 XWIN-NMR 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.

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** → **Other topics** → **Avance Users Guide**. Furthermore, you can refer to the spectrometer hardware documentation. This can be viewed by inserting the BASH CDROM and clicking **Help** → **Other topics** → **BASH spectrometer documentation**.

### USAGE IN AU PROGAMS

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

**SEE ALSO**

lock, lopo, lopo, edlock

# lock

## NAME

lock - lock the magnetic field

## SYNTAX

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

## DESCRIPTION

The command **lock** performs the automatic 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. When you select a solvent, it reads the lock parameters for that solvent and performs an auto lock-in accordingly.

### **lock <solvent>**

reads the lock parameters for the specified solvent and performs an auto lock-in accordingly.

### **lock -acqu**

reads the lock parameters for the solvent defined by the acquisition parameter SOLVENT and perform an auto lock-in accordingly.

### **lock -noauto**

reads the lock parameters for the solvent defined by the acquisition parameter SOLVENT and performs a lock-in (not auto lock-in, see below) procedure.

**lock** reads the lock parameters from the lock table which has been set up with the command **edlock**. (see table 4.1).

The auto lock-in procedure involves the following steps:

1. Irradiation of the lock nucleus with frequency *Lock Freq + Distance*
2. Acquisition of the reflected lock signal
3. Fourier transform and magnitude calculation of the acquired FID
4. Determination of the position of the lock signal in the spectrum.
5. Adjusting the Field such that the lock signal is exactly on resonance.
6. Optimization the lock power, lock gain and lock phase.



Lock parameters	Description	BSMS keyboard
Field	field (H0)	FIELD
LPower	lock power	LOCK POWER
LPhase	lock phase	LOCK PHASE
-	lock gain	LOCK GAIN
LGain	loop gain	Menu <sup>a</sup>
Ltime	loop time	Menu <sup>a</sup>
LFilt	loop filter	Menu <sup>a</sup>
Distance	lock frequency shift	LOCK SHIFT

**Table 4.1**

a. Can be modified by simultaneously pressing the buttons indicated with *MENU*

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. 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. This is the difference between the BSMS lock unit and the SCM lock unit. The latter use the same lock irradiation frequency for each solvent and shift the field to put the lock signal on resonance. Note that all Avance and some AMX/ARX spectrometers are equipped with a BSMS unit.

The command **lock -noauto** simply reads the lock parameters and performs a lock-in accordingly. It does not adjust the field, nor does it optimize any lock parameters. **lock -noauto** is typically used for samples with multiple lock nuclei or mixed solvents. The advantage is that you are sure to lock-in on the lock nucleus that corresponds to the lock parameters whereas the auto lock-in procedure would use the nucleus with the strongest signal.

**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*** → ***Other topics*** → ***Avance Users Guide***. Furthermore, you can refer to the spectrometer hardware documentation. This can be viewed by inserting the BASH CDROM and clicking ***Help*** → ***Other topics*** → ***BASH spectrometer documentation***.

## 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 table 4.1

## INPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acqu - acquisition parameters

<xwhome>/conf/instr/

probehead - current probehead as defined with ***edhead***

<xwhome>/conf/instr/<instrum>/

2Hlock - lock table for nucleus 2H

19Flock - lock table for nucleus 19F

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

bsmspar - solvent and probehead dependent lock parameters

**USAGE IN AU PROGRAMS****LOCK**

executes the command *lock -acqu*.

**SEE ALSO**

edlock, lopo, lopoi, lockdisp, lgain, ltime, lfilter

# lockdisp

## NAME

lockdisp - open the lock display window

## DESCRIPTION

The command **lockdisp** opens the lock display window in which the lock signal is shown. The lock signal is the NMR signal of the lock nucleus which is usually  $^2\text{H}$ .

The lock display can be used in two different ways:

1. 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.

2. 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.

The lock display window offers the following buttons:

**mode**

toggles between two color modes. In one mode, forward and backward sweep have the same color. In the other mode, they have different colors.

**store**

prompts you for a filename to save the current size and position of the lock display window and the current **grid** and **mode** settings. The **read** button allows you to read previously saved settings. **lockdisp** starts up with standard settings which are stored in the file `default`. If this file does not exist, it start up with program internal default settings.

***read***

prompts you for the name of a lock display file which has previously been stored with *store*. After reading in this file, the lock display window is adjusted accordingly.

***quit***

closes the lock display window

**INPUT AND OUTPUT FILES**

<xwhome>/exp/stan/nmr/loc\_win/

default - current lock display settings

**SEE ALSO**

lock, edlock

## lopo, lopoi

### NAME

lopo - set the lock parameters  
lopoi - set the lock parameters interactively

### DESCRIPTION

The command **lopo** reads the lock parameters from **edlock** table for the solvent defined by the acquisition parameter SOLVENT. As such, the lock power, loop gain, loop time, loop filter, lock phase and frequency shift are set to the lock table values of Lpower, LGain, LTime, LFilter, LPhase and Distance, respectively. These values are set on the BSMS unit without performing lock-in.

**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. Note that

**lopo** → **Autolock** is equivalent to **lock -acqu**

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

The command **lopoi** works like **lopo**, except it does not interpret the parameter SOLVENT. **lopoi** shows a list of available solvents from which you can select one. It then sets the lock parameters according to the **edlock** table.

### 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 table 4.1

### INPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acqu - acquisition parameters

<xwhome>/conf/instr/

probehead - current probehead as defined by *edhead*

<xwhome>/conf/instr/<instrum>/

2Hlock - lock table for nucleus 2H

19Flock - lock table for nucleus 19F

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

bsmspar - solvent and probehead dependent lock parameters

## **USAGE IN AU PROGRAMS**

LOPO

## **SEE ALSO**

lock, edlock, lockdisp, lgain, ltime, lfilter





---

# Chapter 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 probehead, you can perform gradient shimming.

# 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, edtone, rsh, wsh, setsh, vish, delsh

# ed tune

## NAME

ed tune - edit tune file

## SYNTAX

ed tune [<name>]

## DESCRIPTION

The command **ed tune** allows you to edit an existing tune file or create a new one. Tune files are used by the **tune** command that performs automatic shimming.

**ed tune** takes one argument and can be used as follows:

### **ed tune**

Shows a list of existing tune files. You can click an entry to edit the corresponding file or you can enter a name in the field *Type New Name*. In the latter case, an empty file will be opened and can be set up as a new tune file.

### **ed tune <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. Two example files, `example` and `example_bsms` are delivered with XWIN-NMR.

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/group/

`example` - standard tune file for spectrometers with a SCM unit  
`example_bsms` - standard tune file for spectrometers with a BSMS unit  
`my_tunefile` - user defined tune file

## SEE ALSO

tune, rsh, wsh, setsh, vish, delsh, gradshim

# rsh

## NAME

rsh - read a set of shim values

## SYNTAX

rsh [<name>]

## DESCRIPTION

The command **rsh** reads a set of shim values to the shim unit. **rsh** without arguments opens a dialog box with a list of available shim sets. When you click on a shim set, its values are loaded to the shim unit. **rsh** can also be entered with a shim file as an argument.

After reading a shim file, it is usually necessary to optimize the shims, especially the Z and Z<sup>2</sup> 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 **autoshimon**, the shims will be continuously optimized during the experiment.

If you specify an argument, then it may contain wildcards; for example:

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

**rsh [m-z]\*** lists all shim files beginning with *m,n,...,z*

Shim files can be stored with the command **wsh** and viewed with **vish**. Single shims can be set with **setsh**.

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/bsms/

shim files

## USAGE IN AU PROGRAMS

RSH(name)

**SEE ALSO**

wsh, setsh, vish, delsh, edtune, tune, gradshim

# setsh

## NAME

setsh - set a single shim

## SYNTAX

setsh [<name>] [<value>]

## DESCRIPTION

The command **setsh** allows you to set single shims. It takes two arguments and can be used in one of the following ways:

**setsh**

prompts you for a shim and a value and then sets the specified shim accordingly

**setsh <shim>**

prompts you for a value and then sets the specified shim accordingly

**setsh <shim> <value>**

sets the specified shim to the specified value

Entire shim sets can be read to the shim unit with **rsh**, stored with **wsh** and viewed with **vish**.

## USAGE IN AU PROGRAMS

SETSH(shim, value)

## SEE ALSO

rsh, wsh, vish, delsh, edtune, tune, gradshim

# 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 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. When you click an entry, the corresponding tune file is interpreted and auto shimming is performed accordingly.

### **tune <tunefile>**

performs auto shimming according to the specified tune file.

### **tune .sx**

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

A example tune file is delivered with XWIN-NMR. 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. 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 XWIN-NMR 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. 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

```
<xwhome>/exp/stan/nmr/lists/group/
```

example - tune file for BSMS (XWIN-NMR 3.1 and newer) <sup>1</sup>

example - tune file for SCM/BSN18 (XWIN-NMR 3.0 and older)

example\_bsms - tune file for BSMS (XWIN-NMR 3.0 and older)

```
<xwhome>/conf/instr/<instrum>/prosol/<probeID>/
```

tunefile - tune file for the current probehead (input for **tune .sx**)

## USAGE IN AU PROGRAMS

```
TUNE(tunefile)
```

---

1. Note that all Avance spectrometers have BSMS. Some AMX/ARX spectrometers have BSMS, other have SCM/BSN18.

**TUNESX**

executes the command ***tune .sx***

**SEE ALSO**

edtone, gradshim, rsh, wsh, setsh, vish, delsh

# vish

## NAME

vish - view a shim file

## SYNTAX

vish [<name>]

## DESCRIPTION

The command **vish** allows you to view a shim file. It takes one argument and can be used in one of the following ways:

### **vish**

opens a dialog box with a list of available shim files. When you click a shim file, it is opened for viewing.

### **vish. <name>**

opens the specified shim file for viewing

If you specify an argument, then it may contain wildcards; for example:

**vish tol\*** lists all shim files beginning with *tol*

**vish [m-z]\*** lists all shim files beginning with *m,n,...,z*

Shim files can be stored with **wsh** and read to the shim unit with **rsh**. Single shims can be set with **setsh**.

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/bsms/

shim files

## SEE ALSO

rsh, wsh, setsh, delsh, edtune, tune, gradshim

# wsh

## NAME

wsh - write a set of shim values

## SYNTAX

wsh [<name>]

## DESCRIPTION

The command **wsh** writes the shim values which are currently installed on the shim unit to a shim file. It takes one argument and can be used in one of the following ways:

### **wsh**

opens a dialog box with a list of currently available shim files. Existing shim files can be overwritten or a new name can be entered.

### **wsh <name>**

writes the current shim values to a file with the specified name. The name of a shim file can be freely chosen.

If you specify an argument, then it may contain wildcards; for example:

**wsh tol\*** lists all shim files beginning with *tol*

**wsh [m-z]\*** lists all shim files beginning with *m,n,...,z*

Shim files can be read with the command **rsh** and viewed with **vish**. Single shims can be set with **setsh**.

## OUTPUT FILES

<xwhome>/exp/stan/nmr/lists/bsms/

shim files

## USAGE IN AU PROGRAMS

WSH(name)

## SEE ALSO

rsh, setsh, vish, delsh, edtone, tune, gradshim



---

# Chapter 6

## Probehead commands

---

This chapter describes the commands which are involved in probehead handling. This involves selecting the current probehead, the tuning and matching procedure and sample insert and eject.

# atma

## NAME

atma - automatic tuning and matching of ATM probeheads

## USAGE

atma [exact | course] [high]

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

## DESCRIPTION

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

1. Stop the sample rotation if it is on.
2. Read the nucleus with the lowest frequency as it was set up with **edasp**.
3. Determine the optimum sweep width and number of steps.
4. Tune and match the probehead.
5. Repeat step 3 to 4 for all other nuclei which were set up with **edasp** in the order of increasing frequency.
6. 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 than **atma** without an argument and will therefore be slower.

**atma course** will determine the optimum tuning and matching less precisely than **atma** without an argument will therefore be faster.

**atma high** will start with the nucleus with the highest frequency, and tune and match the probehead for each nucleus in the order of decreasing frequency.

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

On ATM probeheads, **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 probehead during automaton. In ICON-NMR, you can choose to do that before each experiment, after each sample change or after each solvent change.

**atma** is supported with XWIN-NMR 2.6 and later.

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

For more information on the command **atma** and ATM probeheads, see **Help** → **Other topics** → **Automatic Tuning and Matching**.

## INPUT PARAMETERS

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

## INPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

<xwhome>/conf/instr/<instrum>/

nuclei - nuclei table

<xwhome>/prog/wobble/

acq\_go4 - wobble parameters

Pulsprog\_X - wobble pulse program

## SEE ALSO

atmm, wobb

# atmm

## NAME

atmm - manual tuning and matching of ATM probeheads

## USAGE

atmm [manWbsw]

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

## DESCRIPTION

The command **atmm** is a manual tuning and matching procedure for ATM probeheads. It is not needed very often because ATM probes are designed for automatic tuning and matching with **atma**. Sometimes, however, the probeheads 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 probeheads. In that case, **atmm** allows you to manually tune and match the probehead for that nucleus.

**atmm** performs the following steps:

1. It reads the nucleus with the lowest frequency as it was set up with **edasp**.
2. It determines the optimum sweep width and number of steps.
3. It shows the reflected power (tuning/matching curve) in the XWIN-NMR acquisition data field.
4. It opens the **atmm** control window from where you can:
  - Select the nucleus for which you want to tune and match the probehead. By default, the nucleus with the lowest frequency is selected.
  - Perform course tuning/matching on broadband probeheads. This is the equivalent of setting the sliders on a non ATM probehead to predefined numbers.
  - Perform fine tuning/matching while observing the curve in the XWIN-NMR acquisition data field. This is the equivalent of turning the knobs or moving the sliders on a non ATM probehead. See **wobb** for a description on how to reach the optimum tuning and matching.

- Click **File** → **Quit** to finish the tuning/matching process.

**atmm** can be used with various options. For example, **atmm manWbsw** 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 options.

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

- **atmm** can only be used on ATM probeheads.
- **atmm** 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 XWIN-NMR whereas **wobb** requires you to turn the knobs (or move the sliders) on the probehead.

**atmm** is supported with XWIN-NMR 2.6 and later.

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

For more information on the command **atmm** and ATM probeheads, see **Help** → **Other topics** → **Automatic Tuning and 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

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

<xwhome>/prog/wobble/

acq\_go4 - wobble parameters

Pulsprog\_X - wobble pulse program

## SEE ALSO

atma, wobb

# edhead

## NAME

edhead - edit probehead table

## DESCRIPTION

The command **edhead** opens the probehead table. Each entry in this table contains a probehead name and reference number. A typical entry is:

*5 mm Dual 13C/1H* [03]

The entry which is defined as the current probehead, is highlighted.

The probehead table has a menu **Probe** in the upper left corner that offers the following commands:

### ***New Probe***

The probehead parameter editor will be opened where you can define a new probehead. You can do that by setting the probehead parameters (see below). The name of the new probe will be the combination:

*Sample diameter Probe Type Coil Nuclei Part number Serial number*

Note that the *Part number* and *Serial Number* are only part of newly defined probes, not of the probes in the standard list delivered with XWIN-NMR. The reference number of a new probe will be the first unused number available. This usually the number of the last entry plus one <sup>1</sup>.

### ***Copy Probe***

The current probehead will be copied to a new probehead with the same name. The new probe is added at the end of the list. The reference number will be the first unused number available. This usually the number of the last entry plus one <sup>1</sup>.

### ***Delete Probe***

The current probehead will be deleted. Its reference number will be unused until a new probehead is defined.

---

1. If a probe was deleted and left an unused reference number, this is assigned to the new probe.

***Define as Current Probe***

The highlighted entry will be defined as current probehead. If no entry is highlighted, an error message will appear. If the highlighted probehead has not been defined before, you will be prompted you with the question:

*Do you want to create the probe parameterset?*

If you click *No*, no probehead parameters will be stored but the highlighted entry will still be defined as current probehead. If you click *Yes*, the probehead parameter editor will be opened (see below).

***Edit Probe parameters***

When you click this button, a dialog box with various probehead parameters will appear (see below).

***Exit***

Close the probehead table, saving all changes.

The menu entries ***Define as current probe***, ***Edit probe parameters*** and ***Exit*** are also available as buttons at the bottom of the ***edhead*** dialog box.

The probehead parameter editor displays four groups of parameters:

*Production parameters* like:

Probe type  
Part number  
Serial number  
..

*Sample parameters* like:

Sample diameter  
Sample depth

*Temperature parameters* like:

Type of sensor  
Lowest allowed temperature  
Highest allowed temperature

*Coils parameters* like:

Number of coils

Gas compensation  
 Inside coil nuclei  
 Outside coil nuclei

*History:*

A text field where any information about the probe can be entered

Some probeheads have an additional parameter group, e.g.:

Broadband probes: *Tuning and Matching parameters*

Flow probes: *Probe flow parameters*

Note the probe parameters are just information for the user. They are not required for the acquisition and they are not interpreted by any XWIN-NMR command.

The current probehead as defined with **edhead**, is interpreted by the commands **edprosol**, **edlock**, **lock**, **lopo** and **setpre** as well as during ICON-NMR automation. At the end of an acquisition the current probehead is stored as the acquisition status parameter PROBHD (see **dpa**).

**edhead** can also be executed as a part of the XWIN-NMR configuration suite (command **config**).

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/

probeheads.all - default probehead list (input of first time **edhead**)

<xwhome>/prog/tcl/xwish3\_scripts

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

## INPUT AND OUTPUT FILES

<xwhome>/exp/stan/nmr/lists/

probeheads - probehead list (input of 2<sup>nd</sup> and later **edhead** and output)

<xwhome>/conf/instr/

probehead - current probeheads name and reference number

<xwhome>/conf/instr/probeheads

<part #>\_<serial #>\_<reference #>.par - probe parameters

**SEE ALSO**

cf, atma, atmm, wobb

## **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 probehead, 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



## **ro**

### **NAME**

ro - switch the sample rotation on or off

### **SYNTAX**

ro [ acqu | off | off wait ]

### **DESCRIPTION**

The command **ro** switches the sample rotation on or off. It takes two arguments and can be used as follows:

#### ***ro***

allows you to set the spin rate and switch the spinning on. **ro** first prompts you for the following information:

*Spinning on:* enter *yes* to switch the spinning on or *no* if you only want to set the spin rate.

*Spinning rate:* enter the demand spin rate

#### ***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.

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**

<du>/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**

ij, ej

# wobb

## NAME

wobb - manual tuning and matching of the probehead

## USAGE

wobb [high][ext50][raw][f1|f2]

## DESCRIPTION

The command **wobb** allows you to manually tune and match the probehead. An NMR probehead 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 probehead. However, if either or both are different, part of the transmitter power is reflected by the probehead. This results in a longer 90° pulse or, for a given pulse length, a smaller flip angle. Note that a multi nuclear probehead has a resonance circuit for each nucleus and each nucleus must be tuned and matched separately.

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 probeheads 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 **XWIN-NMR** data field. This is the so called wobble curve. At the probeheads resonance frequency, the curve shows a dip, the minimum reflected power. Tuning the probehead means adjusting its resonance frequency until it reaches **SFO1**. Matching the probehead means adjusting its impedance until the reflected power reaches zero.

The entire wobble procedure involves the following steps:

1. 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.
2. Setup the nucleus or nuclei for the current experiment with **edasp**. This will automatically set the parameters **SFO1**, **SFO2** etc.
3. Click **Acquire** → **Observe fid window** or enter **acqu** to switch to the **XWIN-NMR** acquisition menu. If, however, tuning and matching is observed on the preamplifier, this step can be skipped. (see below).
4. Enter **wobb**. The wobble curve will appear in the **XWIN-NMR** 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 **wobb-SW** or enter **wbchan** to increase the sweep width. You can do this while **wobb** is running.
5. 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.
6. 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 **wobb-SW** or enter **wbchan** in XWIN-NMR. Answer the question "Do you want to change the nucleus" with *yes*.

Repeat step 5 for the current nucleus.

7. If your experiment involves more than two nuclei, repeat step 6 for each further nucleus.
8. Click **stop** or enter **stop** on the command line to finish the wobble procedure.

A probehead has a tuning knob (labelled **T**) and matching knob (labelled **M**) for each resonance circuit. Most probeheads have two, one for <sup>1</sup>H and one for X-nuclei. When the tuning knob reaches the end of its range before the probehead 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 probehead. The horizontal row of LED's indicates tuning, the vertical row indicates matching. When you turn the tuning or matching knob at the probehead, you will see how the number of lit LED's changes. The probehead 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 XWIN-NMR acquisition display. In that case, you can simply click **return** to switch to the processing menu (main menu).

Broadband probeheads 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 probehead. When wobbling is started with these default values, only some fine tuning and matching is required.

The probehead resonance frequency and impedance is dependent on the sample.

This can be a problem in automation, where several samples are measured but the probehead is only matched and tuned on one of them. Bruker ATM probeheads support automatic tuning and matching which can be performed 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* → NUCLEI:

NUC1 - NUC4 - the nuclei for which the probehead is tuned and matched

SFO1 - SFO4 - irradiation frequency

## INPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

<xwhome>/prog/wobble/

acq\_go4 - wobble parameters

Pulsprog\_X - wobble pulse program

## SEE ALSO

atma, atmm, edasp

---

# Chapter 7

## Parameter handling commands

---

This chapter describes commands which are involved in parameter handling. This involves the setting of acquisition parameters, probehead/solvent dependent parameters and nuclei assignment. Furthermore, the usage of predefined parameters sets is discussed.

## ased, as

### NAME

ased - edit the acquisition parameters used in the current pulse program  
as - same as ased but prompts the user for each parameter

### DESCRIPTION

The command **ased** opens a dialog box in which you can set the acquisition parameters which are used for the current experiment. This means that **ased** shows much less parameters than **eda** which shows all acquisition parameters.

**ased** shows three types of parameters, namely parameters that are:

- used in all experiments like NS, TD etc.
- defined by **edasp** like NUC1, SFO1, NUC2, SFO2 etc.
- interpreted by the current pulse program, e.g. D[1], P[1], PL[1] etc.

**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.

**as** works like ased except that it prompts the user for each parameter.

### 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

<xwhome>/exp/stan/nmr/lists/pp/

the pulse program defined by PULPROG

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

## OUTPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

## SEE ALSO

eda, edcpul

# dpa

## NAME

dpa - display the acquisition status parameters

## DESCRIPTION

The command **dpa** displays the acquisition status parameters. These are set by acquisition commands and represent the status of the raw data.

The **dpa** dialog box offers the following buttons/fields:

- **Done**  
closes the **dpa** window or, on 2D or 3D data, go to the next dimension
- **I-Col**  
changes to one column display mode. In this mode a short description is shown for each parameter.
- **Parameter**  
allows you to search for a parameter. Just enter the parameters name (or a part of it) and hit the **Enter** key. The **dpa** window will scroll to the parameters position. If nothing happens, the parameter does not exist or is on the currently displayed page.
- **Next**  
selects the next parameter which starts with the string in the **Parameter** field

Acquisition status parameters can also be viewed by entering their names on the command line. For example:

On a 1D dataset:

**1s ns**

displays the acquisition status parameter NS

On a 2D dataset:

**2s td**

displays the F2 acquisition status parameter TD

**1s td**

displays the F1 acquisition status parameter TD

On a 3D dataset, the preposition **3s** can be used for the F3 dimension.

The NMR Superuser can change the **dpa** dialog box, for example remove parameters which are not used. This can be done from the Windows Explorer or from a UNIX shell by editing the file `normdp` (see below).

## INPUT FILES

`<xwhome>/exp/stan/nmr/form/acqu.l/`

`normdp` - format file for **dpa**

On 2D and 3D data the directories `acqu2.l` and `acqu3.l` contain a `normdp` file for the indirect dimensions.

`<du>/data/<user>/<name>/nmr/<expno>/`

`acqus` - acquisition status parameters

On 2D and 3D data the files `acqu2s` and `acqu3s` are used for the indirect dimensions, respectively (see also chapter 2.3).

## SEE ALSO

`eda`, `lpa`, `dp`, `dpc`, `dpo`, `dpp`, `dpg`, `dpgx`

# eda

## NAME

eda - edit acquisition parameters

## DESCRIPTION

The command **eda** opens a dialog box in which you can set all acquisition parameters. This dialog box offers the following buttons/fields:

- **Save**  
saves all parameters
- **I-Col**  
changes to one column display mode. In this mode a short description is shown for each parameter.
- **Parameter**  
allows you to search for a parameter. Just enter the parameters name (or a part of it) and hit the **Enter** key. The **eda** window will scroll to parameters position and select it. Note that the parameter might be on the current page.
- **Next**  
selects the next parameter which starts with the string in the **Parameter** field.
- **Cancel**  
leaves **eda** without saving any changes

The disadvantage of **eda** is that it shows many parameters, most of which do not have to be changed for a standard experiment. Generally, it is more convenient to use **ased** to set the acquisition parameters. This command only shows the parameters which are actually used for the current experiment.

Apart from using **eda** or **ased**, you can also set acquisition parameters by entering their names on the command line, e.g.:

```
ns  
prompts you to enter the number of scans.  
ns 1000  
sets the number of scans to 1000
```

For 2D datasets, **eda** shows one column of parameters for the F2 dimension and

one for the F1 dimension The *I-Col* button is not available. Note that most parameters are not dimension specific and, as such, appear for the F2 dimension only. 2D parameters can also be set from the command line, e.g.:

***td 4k***

sets the F2 time domain to 4k

***2 td 4k***

sets the F2 time domain to 4k (same as ***td 4k***)

***1 td 4k***

sets the F1 time domain to 4k

For 3D dataset, ***eda*** shows an extra column for the third dimension.

The NMR Superuser can change the ***eda*** dialog box, for example remove parameter which are not used. This can be done from the Windows Explorer or from a UNIX shell by editing the file `acqu.e` (see below).

## INPUT AND OUTPUT PARAMETERS

All acquisition parameters.

### INPUT FILES

<xwhome>/exp/stan/nmr/form/

`acqu.e` - format file for ***eda***

### INPUT AND OUTPUT FILES

<du>/data/<user>/<name>/nmr/<expno>/

`acqu` - acquisition parameters for the acquisition (direct) dimension

`acqu2` - acquisition parameters for F1 indirect dimension (2D) or F2 (3D)

`acqu3` - acquisition parameters for the F1 dimension (3D)

### SEE ALSO

ased, dpa

## 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. 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** or **config**) 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 following rules:

- a) Between the logical frequency channels and FCU's any combination is allowed but only one to one.
- b) 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.

- c) The first four Amplifiers are automatically connected to Switchbox input in a fixed way. These connections cannot be changed.
- d) Between the Switchbox input and Switchbox output, any connection is allowed but only one to one.
- e) 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.
- f) 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.

Two extra switches are available to control the routing:

*Preferred preamplifier*

toggle between selective and unselective Preamplifier module

*Preferred output for 19F*

toggle between the 19F and X Switchbox output for 19F nucleus

When you change these settings, you must click the DEFAULT button to make them effective.

At the bottom of the NUCLEI 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, NUCLEI has a different functions in the two dimensions. 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 dimension. The same principle holds for 3D data; in F3, NUCLEI 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. 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.

However, you only need to set the following switches:

*Preferred preamplifier*

toggle between selective and unselective Preamplifier module

*Preferred output for 19F*

toggle between the 19F and X Switchbox output for 19F nucleus

and click *SAVE* to store them.

The *Preferred preamplifier* determines the default preamplifier module (selective or unselective) for X-nuclei. The *Preferred output for 19F* determines the default output of the switchbox (X or 19F) for 19F.

In XWIN-NMR 3.1 and newer, 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 ALL*. If you make any changes here, you



must make sure that the cables at the spectrometer are connected accordingly. Note that, in XWIN-NMR 3.1 and newer, the command **edasp setpreamp** is implicitly called as a part of spectrometer configuration (command **cf**).

## INPUT AND OUTPUT PARAMETERS

set from **edasp** by setting *Preferred preamplifier*:

DEFRSEL - preferred preamplifier (default routing)

set from **edasp** by setting *Preferred output for 19F*:

DEF19F - preferred output for 19F (default routing)

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

<du>/data/<user>/<name>/nmr/<expno>/

acq - acquisition parameters

<xwhome>/conf/instr/<INSTRUM>/

specpar - routing parameters

## SEE ALSO

edsp, eda, cf, config, expinstall

# getprosol

## NAME

getprosol - get probehead and solvent dependent parameters

## DESCRIPTION

The command **getprosol** reads the probehead and solvent dependent parameters (the prosol parameters) and copies them to the corresponding acquisition parameters. The relations between the prosol and acquisition parameters are shown in table 7.1. 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 than 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:

```
/* relations <filename> */
```

before the actual pulse sequence. Note that the `/* */` characters are obligatory! 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.

**getprosol** replaces the command **gpro** and the AU program **pulsesort** which were used in XWIN-NMR versions older than 3.0.

Description	<i>edprosol</i>	<i>eda</i>
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

## INPUT AND OUTPUT PARAMETERS

see table 7.1

### INPUT FILES

<xwhome>/conf/instr/<instrum>/prosol/reasons

default - reasons file for most experiments

triple - reasons file for Protein experiments

triple\_na - reasons file for DNA experiments

lcnmr - reasons file for LC-NMR

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

nucleus.channel.amplifier - prosol parameters

### OUTPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acqu - acquisition parameters

reasons - copy of the input reasons file

### SEE ALSO

edprosol, eda

## rpar

### NAME

rpar - read a parameter set

### SYNTAX

rpar [<name> [<type>]]

### DESCRIPTION

The command **rpar** reads a parameter set to the current dataset. It takes two arguments and can be used as follows:

- **rpar**  
shows a list of existing parameter sets. When you click on a parameter set, you will get a list of parameter types available in that set. You can select one or more parameter types and then click **Copy** to copy them to the current dataset. Alternatively, you can click **Copy All** to copy all parameter types.
- **rpar <name>**  
shows a list of parameter types available in the parameter set <name>. You can select one or more parameter types and then click **Copy** to copy them to the current dataset. Alternatively, you can click **Copy All** to copy all parameter types.
- **rpar <name> <type>**  
copies the parameter type <type> of the parameter set <name>

The **Cancel** button in there **rpar** dialog box allows you to quit without reading any parameters.

The following parameter types exist:

- *acqu* - acquisition parameters
- *proc* - processing parameters
- *plot* - graphics and plot parameters
- *outd* - output device parameters
- *title* - plot title (user defined parameter sets only)
- *level* - 2D or 3D contour levels (user defined parameter sets only)

- *all* - all of the above parameter types

**rpar** only reads those parameter types which are selected and which exist in the specified parameter set.

Bruker parameter sets are delivered with XWIN-NMR and installed with the command **expinstall**. They all contain the parameter types *acqu*, *proc*, *plot* and *outd*.

User defined parameter sets can be created with **wpar** and contain the parameter types that were selected during creation. These include the types *acqu*, *proc*, *plot*, *outd*, *title* and, for 2D or 3D data, *level*.

**rpar** allows you to read parameters sets of various dimensionalities: 1D, 2D, etc. If the dimensionality of the current dataset and the parameter set you want to read are the same, e.g. both 1D, the dataset parameter files are overwritten. If the current dataset contains data (raw and/or processed data), these are kept. Furthermore, the status parameters are kept so you still have a consistent dataset. However, as soon as you process the data, the new processing parameters are used, the processed data files are overwritten and the processing status parameters are updated. When you start an acquisition, the new acquisition parameters are used, the raw data are overwritten and the acquisition status parameters are updated. If the current dataset is 1D, contains data (raw and/or processed) and you read a 2D parameter set, **rpar** will warn you that the current data will be deleted and ask you whether or not you want to continue. However, this warning will not appear if you enter the command with two arguments, i.e.:

```
rpar <name> <type>
```

In that case, data files of a different dimensionality are simply deleted. The reason is that is that **rpar** with two arguments is used in automation.

After reading a parameter set with **rpar**, you can modify parameters of the various types with the commands:

- **eda** - acquisition parameters
- **edp** - processing parameters
- **edg** and **edgx** - plot parameters
- **edo** - output device parameters
- **edlev** - contour levels
- **setti** - plot title

The first argument of *rpar* may contain wildcards, for example:

*rpar C\** shows all parameter sets beginning with the letter C

*rpar [H-Z]\** shows all parameter sets beginning with a letter between H and Z.

## INPUT FILES

<xhome>/exp/stan/nmr/par/<1D parameter set>/

*acqu* - acquisition parameters (parameter type *acqu*)

*proc* - processing parameters (parameter type *proc*)

*meta* - plot parameters (parameter type *plot*)

*meta.ext* - extended plot parameters (parameter type *plot*)

*outd* - output device parameters (parameter type *outd*)

*title* - plot title (parameter type *title*)

<xhome>/exp/stan/nmr/par/<2D parameter set>/

*acqu* - F2 acquisition parameters (parameter type *acqu*)

*acqu2* - F1 acquisition parameters (parameter type *acqu*)

*proc* - F2 processing parameters (parameter type *proc*)

*proc2* - F1 processing parameters (parameter type *proc*)

*meta* - plot parameters (parameter type *plot*)

*meta.ext* - extended plot parameters (parameter type *plot*)

*outd* - output device parameters (parameter type *outd*)

*title* - plot title (parameter type *title*)

*level* - 2D contour levels (parameter type *level*)

3D parameter sets also contain the files *acqu3* and *proc3* for the third dimension but do not contain the file *meta.ext*. User defined parameter sets may also contain the files *title* (1D and 2D) and *level* (2D and 3D).

## OUTPUT FILES

<du>/data/<user>/nmr/<1D data name>/<expno>/

*acqu* - acquisition parameters (parameter type *acqu*)

<du>/data/<user>/nmr/<1D data name>/<expno>/pdata/<procno>/

*proc* - processing parameters (parameter type *proc*)

*meta* - plot parameters (parameter type *plot*)

*meta.ext* - extended plot parameters (parameter type *plot*)

`outd` - output device parameters (parameter type *outd*)

`title` - plot title (parameter type *title*)

`<du>/data/<user>/nmr/<2D data name>/<expno>/`

`acqu` - F2 acquisition parameters (parameter type *acqu*)

`acqu2` - F1 acquisition parameters (parameter type *acqu*)

`<du>/data/<user>/nmr/<2D data name>/<expno>/pdata/<procno>/`

`proc` - F2 processing parameters (parameter type *proc*)

`proc2` - F1 processing parameters (parameter type *proc*)

`meta` - plot parameters (parameter type *plot*)

`meta.ext` - extended plot parameters (parameter type *plot*)

`outd` - output device parameters (parameter type *outd*)

`title` - plot title (parameter type *title*)

`level` - 2D contour levels (parameter type *level*)

3D data also contain the files `acqu3` and `proc3` for the third dimension but do not contain the file `meta.ext`.

## USAGE IN AU PROGRAMS

`RPAR(name, type)`

## SEE ALSO

`wpar`, `dirpar`, `delpar`, `renpar`, `expinstall`



# wpar

## NAME

wpar - write a parameter set

## SYNTAX

wpar [<name> [< type>]]

## DESCRIPTION

The command **wpar** stores the parameters of the current dataset in a parameter set. This parameter set is then available for general usage and can be read to any dataset with **rpar**.

**wpar** allows you to overwrite an existing parameter set or to enter a new name. It takes two arguments and can be used in one of the following ways:

### **wpar**

opens a dialog box with a list of existing parameter sets. You can click one of them, or enter a name in the field **Type New Name**. A new dialog box will then open showing a list parameter types. If you chose an existing parameter set, the existing parameter types in it are highlighted. You can select/deselect parameter types by clicking their entries. Then you can click **Copy** to store the highlighted types. Alternatively, you can click **Copy All** to store all types.

### **wpar <name>**

shows a list of parameter types. If you specified an existing parameter set, the existing parameter types in it are highlighted. You can select/deselect parameter types by clicking their entries. You can then click **Copy** to store the highlighted types. Alternatively, you can click **Copy All** to store all types.

### **wpar <name> <type>**

stores parameters of type <type> to parameter set <name>. If <type> already exists in <name>, it is overwritten.

The **Cancel** button in the **wpar** dialog box allows you to quit without storing any parameters

The following parameter types exist:

- *acqu* - acquisition parameters

- *proc* - processing parameters
- *plot* - graphics and plot parameters
- *outd* - output device parameters
- *title* - plot title
- *level* - 2D or 3D contour levels
- *all* - all of the above parameter types

**wpar** is often used as part of the following sequence:

1. Define a new dataset with **edc** or **new**.
2. Enter **rpar** to read a Bruker parameter set which defines the experiment you want to do.
3. Modify the acquisition parameters (with **eda**) to your preference and run the acquisition.
4. Modify processing parameters (with **edp**) to your preference and process the data.
5. Modify the plot parameters (with **edg**) to your preference, set the output device parameters (with **edo**) and plot the dataset <sup>1</sup>.
6. Store the parameters with **wpar** for general usage.

The first argument of **wpar** may contain wildcards, for example:

**wpar C\*** lists all parameter sets which begin with the letter C

**wpar [H-Z]\*** lists all parameter sets which begin with a letter between H and Z.

## INPUT FILES

<du>/data/<user>/nmr/<1D data name>/<expno>/

*acqu* - acquisition parameters (parameter type *acqu*)

<du>/data/<user>/nmr/<1D data name>/<expno>/pdata/<procno>/

*proc* - processing parameters (parameter type *proc*)

*meta* - plot parameters (parameter type *plot*)

*meta.ext* - extended plot parameters (parameter type *plot*)

*outd* - output device parameters (parameter type *outd*)

---

1. Alternatively, you can use XWIN-PLOT whose layout is not part of the parameter set.

<du>/data/<user>/nmr/<2D data name>/<expno>/

acq - F2 acquisition parameters (parameter type *acq*)  
acq2 - F1 acquisition parameters (parameter type *acq*)

<du>/data/<user>/nmr/<2D data name>/<expno>/pdata/<procno>/

proc - F2 processing parameters (parameter type *proc*)  
proc2 - F1 processing parameters (parameter type *proc*)  
meta - plot parameters (parameter type *plot*)  
meta.ext - extended plot parameters (parameter type *plot*)  
outd - output device parameters (parameter type *outd*)

3D data also contain the files `acq3` and `proc3` for the third dimension but do not contain the file `meta.ext`. Note that only the files whose parameter type has been selected within **wpar** are input files.

## OUTPUT FILES

<xwhome>/exp/stan/nmr/par/<1D parameter set>/

acq - acquisition parameters (parameter type *acq*)  
proc - processing parameter (parameter type *proc*)  
meta - plot parameters (parameter type *plot*)  
meta.ext - extended plot parameters (parameter type *plot*)  
outd - output device parameters (parameter type *outd*)  
title - plot title (parameter type *title*)

<xwhome>/exp/stan/nmr/par/<2D parameter set>/

acq - F2 acquisition parameters (parameter type *acq*)  
acq2 - F1 acquisition parameters (parameter type *acq*)  
proc - F2 processing parameters (parameter type *proc*)  
proc2 - F1 processing parameters (parameter type *proc*)  
meta - plot parameters (parameter type *plot*)  
meta.ext - extended plot parameters (parameter type *plot*)  
outd - output device parameters (parameter type *outd*)  
title - plot title (parameter type *title*)  
level - 2D contour levels (parameter type *level*)

3D parameter sets also contain the files `acq3` and `proc3` for the third dimension but do not contain the file `meta.ext` and `title`. Note that only the files whose parameter type has been selected within **wpar** are output files.

**USAGE IN AU PROGRAMS**

WPAR(name, type)

**SEE ALSO**

rpar, dirpar, delpar, renpar, expinstall

---

# Chapter 8

## Pulse and AU program commands

---

This chapter describes commands for the setup of pulse programs, gradient programs, CPP programs and AU programs. Furthermore, it describes the setup of various lists which can be used in acquisition like variable pulse or variable delay lists.

# edau

## NAME

edau - create or edit AU programs

## SYNTAX

edau [<name>]

## DESCRIPTION

The command **edau** allows you to list, create or edit AU programs. When used without argument, a two column list appears with the Bruker AU programs on the right side and the user defined AU programs on the left side. When you click a Bruker AU program, it is shown in view mode which means it cannot be modified. When you click a user defined AU program, it is opened with an editor and can be modified. When you close the view window or editor, you are prompted for one of the following options:

- (c)ompile - compile the AU program
- (b)ack - go back to displaying the list of AU programs
- (q)uit - quit without compiling the AU program

When you choose *c* for compilation (the default option), the AU program will be compiled using a C compiler. After successful compilation, the AU program can be executed by typing its name.

When **edau** is entered with an argument, the specified AU program will be opened. This allows you to write a new AU program or modify an existing one. The argument may contain wildcards, e.g. **edau a\*** displays a list of all AU programs starting with *a*.

Bruker AU programs must be installed once with **expinstall** before they can be opened with **edau**. The installation must be repeated when a new version of XWIN-NMR is installed. By default, Bruker AU programs are opened in view mode, which means they cannot be modified. However, if you enter **edau** and click the **Edit** button, the NMR Superuser password is requested and the program switches to *Edit* mode. When you now click a Bruker AU program, it will be opened with an editor and can be modified. Nevertheless, we recommend to leave the Bruker AU programs unchanged. If you want a modified version, just create a new AU program, read in the Bruker AU program, modify it to your

needs and store it.

**edau** uses the editor which is defined in the XWIN-NMR User Interface. If you want to use a different editor, type **setres** and modify the entry **Editor**.

For details on writing, compiling, and executing AU programs please refer to the AU reference manual (click **Help** → **Other topics** → **Writing AU programs**).

## INPUT FILES

<xwhome>/exp/stan/nmr/au/src/

AU program source files

**If the AU program is compiled (c option of edau):**

/<xwhome>/exp/stan/nmr/au/

makeau - AU compilation script

vorspann - C language definition file

submacro - preprocessor replacing macros and includes

<xwhome>/prog/include/

aucmd.h - AU macro definition file

/<xwhome>/prog/include/inc

AU program includes

## OUTPUT FILES

<xwhome>/exp/stan/nmr/au/src/

AU program source files

**If the AU program is compiled (c option of edau):**

<xwhome>/prog/au/bin/

AU program executable binary files

## SEE ALSO

expinstall, comepileall, cpluser, cplbruk, delau, renau

# edcgp

## NAME

edcgp - edit current gradient program

## SYNTAX

edcgp [<name>]

## DESCRIPTION

The command **edcgp** allows you to create or edit the current gradient program. The current gradient program is defined as the gradient program of the foreground dataset as defined by the acquisition parameter GRDPROG.

**edcgp** takes one argument and can be used as follows:

- **edcgp**  
open the current gradient program
- **edcgp <name>**  
open the gradient program <name> and make it the current gradient program.

If you specify an argument, then it may contain wildcards; for example:

**edcgp SIN\*** lists all gradient programs beginning with SIN

**edcgp [m-z]\*** lists all gradient programs beginning with m,n,...,z

## INPUT PARAMETERS

to be set with **eda** or by typing **grdprog** :

GRDPROG - the current gradient program (input of **edcgp**)

## OUTPUT PARAMETERS

can be viewed with **eda** or by typing **grdprog** :

GRDPROG - the current gradient program (output of **edcgp <name>**)

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/gp/\*



gradient programs

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

## **OUTPUT FILES**

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

## **SEE ALSO**

edgp

# edcpd

## NAME

edcpd - edit composite pulse decoupling (CPD) programs

## SYNTAX

edcpd [<name>]

## DESCRIPTION

The command **edcpd** allows you to list, create or edit CPD programs. If you enter **edcpd** without arguments, a list of all CPD programs is displayed. The list includes both the Bruker and the user defined CPD programs. When you click on a CPD program, it is opened with an editor. Alternatively, or you can enter a name in the field "Type New Name" to create a new CPD program. The **Print** button allows you to print the list of CPD programs.

If you enter the command with an argument, e.g. **edcpd <name>**, the CPD program <name> is opened or, if it does not exist, it is created. The argument may contain wildcards; e.g. **edcpd a\*** displays a list of all CPD programs which start with *a*.

Bruker CPD programs must be installed with **expinstall** before they can be opened with **edcpd**.

**edcpd** uses the editor which is defined in the XWIN-NMR User Interface. You can change it by typing **setres** and specify it in the field **Editor**.

## INPUT AND OUTPUT FILES

<xwhome>/exp/stan/nmr/lists/cpd/\*

Bruker and user defined CPD programs

## SEE ALSO

expinstall

# edcpul

## NAME

edcpul - edit the current pulse program

## SYNTAX

edcpul [<name>]

## DESCRIPTION

The command **edcpul** allows you to create or edit the current pulse program. The current pulse program is defined as the pulse program of the foreground dataset as defined by the acquisition parameter PULPROG.

**edcpul** takes one argument and can be used as follows:

- **edcpul**  
opens the current pulse program
- **edcpul <name>**  
opens the pulse program <name> and makes it the current pulse program.

Bruker pulse programs are opened in view mode which means they cannot be modified. User defined pulse programs are opened with an editor and can be modified.

If you specify an argument, then it may contain wildcards; for example:

**edcpul cos\*** lists all pulse programs beginning with cos

**edcpul [m-z]\*** lists all pulse programs beginning with m,n,...,z

## INPUT PARAMETERS

to be set with **eda** or by typing **pulprog** :

PULPROG - the current pulse program (input of **edcpul**)

## OUTPUT PARAMETERS

can be viewed with **eda** or by typing **pulprog** :

PULPROG - the current pulse program (output of **edcpul <name>**)

**INPUT FILES**

<xwhome>/exp/stan/nmr/lists/pp

Bruker and user defined pulse programs

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

**OUTPUT FILES**

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters (output of **edcpul <name>**)

**SEE ALSO**

edpul, edcpd, edgp

# 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. This list includes both the Bruker and the user defined gradient programs. When you click a gradient program it will be opened with an editor. Alternatively, you can enter a name in the field "Type New Name" to create a new gradient program. The **Print** button allows you to print the list of gradient programs.

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 **expinstall** before they can be opened with **edgp**.

**edgp** uses the editor which is defined in the XWIN-NMR User Interface. You can change it by typing **setres** and specify it in the field **Editor**.

## INPUT AND OUTPUT FILES

<xwhome>/exp/stan/nmr/lists/gp/\*

gradient programs

## SEE ALSO

edcpul, edpul, edcgp, expinstall

# edlist

## NAME

edlist - edit various lists

## SYNTAX

edlist [<type> [<name>]]

## DESCRIPTION

The command **edlist** allows you to list, create or edit various lists like delay lists, pulse lists and frequency lists. Most of these lists are used during acquisition. **edlist** can be used in one of the following ways:

**edlist** - displays all lists types

**edlist <type>** - displays all entries of the list type <type>

**edlist <type> <name>** - creates/edits the list <name> of type <type>

The second argument may contain wildcards, e.g. **edlist vd a\*** displays all variable delay lists which start with *a*.

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. At the end of this description we will discuss an alternative way to use lists, circumventing the acquisition parameter.

All lists that can be used by during acquisition are displayed in table 8.1. They typically contains 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.

## Delay lists

Delay lists contain any number of delay values with the corresponding units, e.g.:

10m  
2s

list type	description	defined by parameter	read by pulse progr. statements
<i>vd</i>	delay list	VDLIST	vd, ivd, vdidx
<i>vp</i>	pulse list	VPLIST	vp, ivp, vpidx
<i>f1</i>	frequency list (Avance)	FQ1LIST, FQ2LIST etc.	fq1, fq2, fq3 etc.
<i>f1, f2, f3</i>	frequency list (A*X)	F1LIST, F2LIST etc.	o1, o2, o3
<i>vc</i>	counter list	VCLIST	lo to x times c, ivc
<i>va</i>	amplitude list	VALIST	define list<power>
<i>ds</i>	dataset list	DSLIST	wr #n, wr ##, ifp, dfp, rfp
<i>vt</i>	temperature list	VTLIST	RVTLIST, VT, IVTLIST, DVTLIST <sup>a</sup>
<i>masr</i>	MAS spin rate list		AU prog multimas
<i>wave</i>	Shaped pulse list	SP07	sp1, sp2 etc.

**Table 8.1** Lists used in acquisition

- a. Note that these are AU program macros rather than pulse program statements

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* - 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.:

10μ  
20m

where μ = 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` - 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:

```
30μ 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
```

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:



```
l o t o x t i m e s c
```

where *x* is a pulse program label and *c* is the value at the current position of the counter list

```
i v c
```

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 defini-

tions:

```
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*** → ***Other topics*** → ***Writing pulse programs***

### **Dataset lists**

A dataset list contains any number of dataset definition, 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.

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
increment 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

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

### Shaped pulse lists

A shaped pulse list is a JCAMP files that defines a shaped pulse. It contains a JCAMP header and a list of relative power values (X) and phase values (Y). An example of a shaped pulse list is:

```
##TITLE= /u/exp/stan/nmr/lists/wave/Crp60comp.4
##JCAMP-DX= 5.00 Bruker JCAMP library
##DATA TYPE= Shape Data
##ORIGIN= Bruker Analytik GmbH
##OWNER= <demo>
##DATE= 99/02/09
##TIME= 19:06:33
##MINX= 0.000000e+00
##MAXX= 1.000000e+02
##MINY= 1.110700e-01
##MAXY= 3.599617e+02
##$SHAPE_EXMODE= Adiabatic
##$SHAPE_TOTROT= 1.800000e+02
##$SHAPE_BWFAC= 0.000000e+00
##$SHAPE_INTEGFAC= 1.564030e-01
##$SHAPE_MODE= 0
```

```

##NPOINTS= 4000
##XYPOINTS= (XY..XY)
0.000000e+00, 1.833010e+02
7.893400e-01, 1.779120e+02
1.578620e+00, 1.725340e+02
2.367810e+00, 1.671660e+02
3.156850e+00, 1.618090e+02
3.945700e+00, 1.564630e+02
4.734300e+00, 1.511280e+02
5.522600e+00, 1.458040e+02
6.310560e+00, 1.404900e+02
7.098130e+00, 1.351870e+02
7.885250e+00, 1.298950e+02
8.671890e+00, 1.246140e+02
9.457980e+00, 1.193440e+02
..
..

```

Shaped pulse lists are interpreted by a pulse program statement like:

```
p13:sp2:f1
```

which executes a shaped pulse with length P13 on channel f1, using the pulse defined by second entry (sp2) in the table defined by the acquisition parameter SP07.

### 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* → *Other topics* → *Writing pulse programs*

The ***edlist*** command also shows various files which are programs rather than lists of values, for example pulse programs, CPD programs, gradient programs and macros. These, however, are normally opened with the dedicated commands ***edpul***, ***edcpd***, ***edgp*** and ***edmac*** rather than with ***edlist***.

## INPUT AND OUTPUT FILES

<xwhome>/exp/stan/nmr/lists

- pp - pulse programs
- cpd - CPD programs
- gp - gradient programs
- wave - shaped pulse lists
- preemp - preemphasis lists
- ds - variable dataset lists
- va - variable amplitude (power) lists
- vc - variable counter lists
- vd - variable delay lists
- vp - variable pulse lists
- vt - variable temperature lists
- f1 - variable frequency lists
- f2, f3 - variable frequency lists (A\*X spectrometers only)
- mac - XWIN-NMR macros
- roi - 2D integral ranges
- scl - scaling region files
- masr - MAS spin rate list

## SEE ALSO

edmisc, edpul, edcpd, edgp, edmac

# edpul

## NAME

edpul - edit a pulse program

## SYNTAX

edpul [<name>]

## DESCRIPTION

The command **edpul** allows you to list, edit or create pulse programs.

**edpul** without arguments opens a dialog box with two columns of pulse programs. The right column shows the Bruker pulse programs, the left column the user defined pulse programs. When you click a Bruker pulse program, it is opened in view mode which means it cannot be changed. When you click a user defined pulse program, it is opened with an editor and can be modified. Instead of opening an existing pulse program, you can enter a name in the field "Type New Name" to create a new one.

The **edpul** dialog box also contains a button **Edit** which allows you to switch to *Edit-mode* and modify Bruker pulse programs. Be careful: they are overwritten when you install a new version of XWIN-NMR. Therefore, if you want to modify a Bruker pulse program you should store it under a new name.

**edpul <name>** opens the pulse program <name>. If <name> does not exist, an empty file is created and opened with an editor.

If you specify an argument, then it may contain wildcards; for example:

**edpul cos\*** lists all pulse programs beginning with *cos*

**edpul [m-z]\*** lists all pulse programs beginning with *m,n,...,z*

Bruker pulse programs must be installed with **expinstall** before they can be opened with **edpul**.

**edpul** uses the editor which is defined in the XWIN-NMR User Interface. If you want to use a different editor, type **setres** and modify the entry **Editor**.

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/pp/\*

Bruker and user defined pulse programs

**OUTPUT FILES**

<xwhome>/exp/stan/nmr/lists/pp/\*

user defined pulse programs

**SEE ALSO**

edcpul, edcpd, edgp, edcgp, expinstall, edlist





---

# Chapter 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 ICON-NMR.

## acqu

### NAME

acqu - switch to the acquisition menu

### DESCRIPTION

The command **acqu** switches XWIN-NMR to the acquisition menu. This is the menu where the raw data (FID) is displayed. Typing **acqu** is equivalent to clicking *Acquire* → *Observe fid window*. This command can only be applied from the processing menu (main menu).

In principle, an acquisition can be started from the main menu simply by typing **zg** or running an acquisition AU program like **au\_zg**. Furthermore, all acquisition commands are available under the *Acquire* menu item. Nevertheless, there are several reasons to switch to the acquisition menu first:

- to tune and match the probe by optimizing the wobble curve in the acquisition data field (command **wobb**)
- to adjust acquisition parameters interactively by observing the effect on a single scan FID (command **gs**)
- to observe the accumulation of the FID during acquisition (command **zg**)

If a long term acquisition has been started and is no longer observed, it is recommended to switch back to the processing menu by clicking *return*.

### SEE ALSO

wobb, rga, gs, zg

# butselnmr

## NAME

butselnmr - easy acquisition interface for selective experiments

## DESCRIPTION

The command **butselnmr** opens an icon box from which you can run 1D selective NMR experiments. It is an easy to use interface that is especially meant for the novice or occasional users.

The icon box contains buttons for the following actions:

### 1. *Create New Dataset*

Allows you to define a new dataset and switches the display to that dataset (executes the XWIN-NMR command **edc**).

### 2. *Lock*

Allows you to select the solvent and then performs the auto lock-in (executes the XWIN-NMR command **lock**).

### 3. *Tune/Match Probe*

Switches to the acquisition menu and allows you to tune and match the probehead (executes the XWIN-NMR command **wobb**).

### 4. *Automatic Shimming*

Performs shimming according to the tune file defined for the current probehead<sup>1</sup> (executes the XWIN-NMR command **tune .sx**).

### 5. *routine PROTON/def. regions*

Runs a 1D proton experiment performing the following steps:

- prompts the user for the experiment number (EXPNO)
- executes the XWIN-NMR command **rpar PROTON all** to read the PROTON parameter set
- executes the XWIN-NMR command **getprosol** to read the probe and solvent dependent parameters
- shows the experiment time and allows the user to actually start the acquisition (by clicking **OK**) or stop (by clicking **Cancel**). In the latter case,

---

1. A probehead dependent tune file can be defined from the **edprosol** dialog box.

the dataset has been created and the acquisition can be started at a later time, for example with ***xaua***.

- selects the region for selective excitation and stores it as `reg` file.

***selective pulse calib.***

this button allows you to determine the 90° selective pulse

***selective excitation, sel.ext. with gradients*** etc.

these buttons allow you to perform selective experiments

***User defined #1, User defined #2***

these button allow you to perform user defined eperiments.It executes the AU program `User_au_1` that must be setup by the user.

**6. *Processing***

Processes multiple FIDs that are stored in a series of contiguous experiments numbers (EXPNO's). It prompts you for the first EXPNO and the number of EXPNO's and then performs the command ***efp (em, ft, pk)*** on each EXPNO. Note that ***butselnmr*** actually runs the AU program ***multiefp*** to perform this task.

**7. *Plot***

Plots the current data by executing the XWIN-NMR command ***plot***.

**8. *Continue***

If the acquisition hangs for some reason, clicking this button continues with the next experiment.

**9. *Exit***

Exits the ***butselnmr*** window

For each button a help text is available that appears when you move the cursor over the buttons.

## INPUT FOLDERS

<xwhome>/exp/stan/nmr/par/

PROTON/\* - routine proton experiment

<xwhome>/prog/au/bin/

`butselau` - `butselnmr` AU program binary executable

`butsel90` - AU program to determine the 90° selective pulse (executable)

`multiefp` - AU program to process a series of 1D data (executable)

<xwhome>/exp/stan/nmr/au/src/

butselau - butselnmr AU program source

butsel90 - AU program to determine the 90° selective pulse (source)

multiefp - AU program to process a series of 1D data (source)

<xwhome>/prog/tcl/xwish3\_scripts

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

## **SEE ALSO**

buttonnmr

# buttonnmr

## NAME

buttonnmr - easy acquisition interface for common experiments

## DESCRIPTION

The command **buttonnmr** opens an icon box from which you can run common 1D and 2D NMR experiments. It is an easy to use interface that is especially meant for the novice or occasional users.

At the top of the window, the field *Xnucleus* allows you to select the X-nucleus you want to measure. This feature exists for XWIN-NMR 3.1 and newer. For earlier versions, X-nuclei measurements are restricted to 13C.

The icon box offers you buttons for the following actions:

### 1. *Create New Dataset*

Allows you to define a new dataset and switches the display to that dataset (executes the XWIN-NMR command **edc**).

### 2. *Lock*

Allows you to select the solvent and then performs the auto lock-in (executes the XWIN-NMR command **lock**).

### 3. *Tune/Match Probe*

Switches to the acquisition menu and allows you to tune and match the probehead (executes the XWIN-NMR command **wobb**).

### 4. *Automatic Shimming*

Performs shimming according to the tune file defined for the current probe head <sup>1</sup> (executes the XWIN-NMR command **tune .sx**).

### 5. *routine PROTON*

Runs a 1D proton experiment performing the following steps:

- prompts the user for the experiment number (EXPNO)
- executes the XWIN-NMR command **rpar PROTON all** to read the PROTON parameter set

---

1. A probehead dependent tune file can be defined from the **edprosol** dialog box.

- executes the XWIN-NMR command **getprosol** to read the probe and solvent dependent parameters
  - shows the experiment time and allows the user to actually start the acquisition (by clicking **OK**) or stop (by clicking **Cancel**). In the latter case, the dataset has been created and the acquisition can be started at a later time, for example with **xaua**.
- 6. *X nucleus without 1Hdec., X nucleus 1Hdec. with NOE***, ect.  
these buttons allow you to perform the Bruker standard experiments that corresponds to the button name and the X-nucleus defined at the top of the window (XWIN-NMR 3.1 and newer).
- routine 13C, DEPT90 13C***, ect.  
these buttons allow you to perform the 13C experiment that corresponds to the button name (XWIN-NMR 3.0 and older).
- 7. *User defined #1***  
these button allow you to perform a user defined experiment. It executes the AU program *User\_au\_1*. You can create this AU program with the command **edau User\_au\_1**. In the same way, you can activate the button ***User defined #2***.
- 8. *Processing***  
Processes the raw data. For a 1D experiment, this involves the command sequence **ef**, **apk** and **abs**, for a 2D experiment the command **xfb** (for more information on these commands see the Processing Reference Manual).
- 9. *Plot***  
Plots the data by executing the XWIN-NMR command **plot**.
- 10. *Continue***  
If the acquisition hangs for some reason, clicking this button continues with the next experiment.
- 11. *Exit***  
Exits the **buttonnmr** window
- For each button a help text is available that appears when you move the cursor over the buttons.

## INPUT FOLDERS

<xwhome>/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  
INV4LPLRND SW/\* - HMBC experiment  
INVIETGPSW/\* - gradient HMQC experiment  
INV4GPLPLRND SW/\* - gradient HMBC experiment

<xwhome>/prog/au/bin/

buttonau - buttonnmr AU program (binary)

<xwhome>/exp/stan/nmr/au/src/

buttonau - buttonnmr AU program (source)

<xwhome>/prog/tcl/xwish3\_scripts

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

## SEE ALSO

butselnmr



## **expt**

### **NAME**

expt - display the experiment time

### **DESCRIPTION**

The command **expt** calculates and displays the experiment time for the current data set. 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.

### **INPUT PARAMETERS**

NS - number of scans

AQ - acquisition time in seconds

### **SEE ALSO**

zg, gs

## go

### NAME

go - perform an acquisition and add 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 results as **zg**.

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).

In XWIN-NMR 3.0 and older, the **go** command can only be used on 1D data. In XWIN-NMR 3.1 and newer, it can also be used in 2D and 3D experiments. 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 dimension increment (see above). However, this problem only occurs when you use the **wr** statement to write the data to disk. In XWIN-NMR 3.1 and newer, you can use the **mc** statement instead. In that case, 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 XWIN-NMR 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*, suspend, resume

## gs

### NAME

gs - go setup; interactive parameter optimization during acquisition

### DESCRIPTION

The command **gs** allows you adjust parameters interactively during an acquisition. The FID display is continuously updated showing the effect of each parameter change.

**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

In order to see the effect of the parameter changes on the FID, you must first go to the acquisition menu. You can do that by clicking *Acquire* → *Observe fid window* or by entering **acqu** on the command line. The command **gs** will open a dialog box with a set of acquisition parameters. If you do not see the dialog box, it is probably iconified and you must first click its icon.

The dialog box only shows the parameters which are typically set during **gs** like irradiation frequencies, pulse lengths and power levels. By default, the irradiation frequency offset is selected for adjustment. A slider is available for each nucleus which has been set with **edasp**. You can adjust the frequency offset O1 by putting the cursor on its slider, pressing the left mouse button and moving the mouse. Alternatively, you can increase/decrease the value by clicking or pressing to the left/right of the slider. Other parameters can be changed in the same way. 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] - power levels  
SP[0-31] - shape power  
AMP[0-31] - amplitude (XWIN-NMR 3.1 and newer)  
PHCOR[0-31] - reference phases  
RG - receiver gain

P[0-31] - pulse lengths

D[0-31] - delays

Note that moving the RG slider causes an exponential change in the receiver gain. Therefore, it is often necessary to decrease the sensitivity before changing RG. 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.

The GS dialog box also contains the following buttons:

- *sens\*10* - increase the sensitivity of the sliders by a factor of ten
- *sens/10* -- decrease the sensitivity of the sliders by a factor of ten
- *save* - save the parameter that was changed last
- *save all* - save all changes
- *reset* - reset the parameter that was changed last
- *reset all* - reset all changes
- *stop acquisition* - stop the acquisition and leave the *gs* dialog box

As an alternative to moving sliders, you can also change parameters from the XWIN-NMR command line while *gs* is running. If you change a parameter from the above list, the slider value will be automatically updated and the effect will be shown in the FID display. For all other parameters, the effect of a change will not be shown. The entered value is, however, stored and becomes effective when a new acquisition command is executed (for example with *gs*, *zg* or *rga*).

## INPUT AND OUTPUT PARAMETERS

See the parameter list above.

## INPUT AND OUTPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

## SEE ALSO

zg, go, rga

# halt

## NAME

halt - halt the acquisition after of 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. Note that **halt** does not complete the current phase cycle.

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

## OUTPUT PARAMETERS

all acquisition parameters

## OUTPUT FILES

In a 1D dataset:

`<du>/data/<user>/nmr/<name>/<expno>/`

`fid` - raw data

`acqus` - acquisition parameters

In a 2D dataset:

`<du>/data/<user>/nmr/<name>/<expno>/`

`ser` - raw data

`acqus` - F2 acquisition status parameters

`acqu2s` - F1 acquisition status parameters

In a 3D dataset:

`<du>/data/<user>/nmr/<name>/<expno>/`

`ser` - raw data

`acqus` - F3 acquisition status parameters

`acqu2s` - F2 acquisition status parameters

acqu3s - F1 acquisition status parameters

**SEE ALSO**

stop, zg, go, rga, wobb

## **iconnmr**

### **NAME**

iconnmr - user interface for routine spectroscopy and automation

### **DESCRIPTION**

The command ***iconnmr*** starts the an icon driven user interface ICON-NMR for routine spectroscopy and automation. ICON-NMR can also be started from the desktop. It is fully described the ICON-NMR manual. The online version of this manual can opened by clicking ***Help*** from the ICON-NMR window.



**ii****NAME**

ii - initialize interface

**DESCRIPTION**

The command ***ii*** initializes the spectrometer interface. It tries to access all hardware parts of the spectrometer which are needed for the current experiment and loads the acquisition parameters.

***ii*** must be executed once when the spectrometer has been switched off.

**INPUT PARAMETERS**

all acquisition parameters

**USAGE IN AU PROGRAMS**

II

**SEE ALSO**

zg

# popt

## NAME

popt - parameter optimization

## DESCRIPTION

The command **popt** allows you to optimize acquisition parameters like pulses and delays. Before you start an optimization, you must run one acquisition with 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. You can do that by clicking the left mouse button in the data field to put the cursor on the spectrum and then click the middle mouse button on both side of the desired region. The XWIN-NMR display will automatically be adjusted to show the selected region only. You must now store this region by entered **dp11** and answer each question by hitting the **Enter** key.

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:

*On/Off*

Only parameters which are switched *On* will be optimized

*Parameter*

the parameter to be optimized. If a parameter is not checked, it will be stored as comment in the optimization protocol.

*OPTIMUM*

the optimization criterion (see below)

*STARTVAL*

the first value of the parameter

*ENDVAL*

the last value of the parameter (empty for INC  $\neq$  0 and VARMOD = LIN)

*NEXP*

number of experiments (empty for INC  $\neq$  0 and VARMOD = LIN)

*VARMOD*

parameter variation mode: linear or logarithmic

*INC*

parameter increment value

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

***Halt optimize*** : stop the optimization

***Read protocol*** : open a text file with the optimization result

***Add parameter*** : add a parameter entry

***Read parameter file*** : read an external popt setup file(e.g. from another dataset)

***SAVE*** - save the current optimization setup

***RESTORE*** - restore the last saved optimization setup

***Update data set*** - update popt setup after changing to a different dataset

***EXIT*** - exit the popt dialog box

Clicking the button ***Start optimize*** will start the optimization process. Note that it runs on the current dataset. For each parameter, 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 processed data file under the current dataset name and experiment number but under a different processing number. For the first parameter that is optimized this is PROCNO 999, for the second parameter 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 PARAMETERS)

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.

***Run optimization in background***

If checked, the foreground dataset will remain the same during the optimization. If it is not checked, the XWIN-NMR display will change to PROCNO 999 where the optimization result is displayed.

***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**.

If you want to rerun an optimization, you must first return to the starting PROCNO.

The command **popt** replaces the AU programs **paropt**, **paroptlin**, **paroptlog** and **parray**, which were used in XWIN-NMR versions older than 3.0.

**INPUT FILES**

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

popt.array - parameter optimization setup (input of **RESTORE**)

popt.protocol - parameter optimization result (input of **Read protocol**)

<du>/data/<user>/nmr/<name>/<expno>/pdata/<procno>/

proc - processing parameters

<xhome>/prog/tcl/xwish3\_scripts

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

<xhome>/prog/au/bin/

poptau - AU program that runs the optimization (executable)

pophalt - AU program that halts the current optimization (executable)

<xhome>/exp/stan/nmr/au/src/

poptau - AU program that runs the optimization (source)

`popthalt` - AU program that halts the current optimization (source)

## OUTPUT FILES

`<du>/data/<user>/nmr/<name>/<expno>/pdata/999`

1r - processed data containing the optimization result of the first parameter

`<du>/data/<user>/nmr/<name>/<expno>/pdata/998`

1r - processed data containing the optimization result of the 2<sup>nd</sup> parameter

`<du>/data/<user>/nmr/<name>/<expno>/`

`popt.array` - parameter optimization setup (output of **SAVE**)

`popt.protocol` - parameter optimization result (output of **Start optimize**)

`<du>/data/<user>/nmr/<name>/899/`

ser - 2D raw data containing the optimization result

## SEE ALSO

gs

## 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* → *Other topics* → *Writing pulse programs*

A resumed acquisition does not start with dummy scans. This can be a problem if the recycle delay is shorter than 4 times the T1 value of the measured nucleus.

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, zg, go

## **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, zg, go

# stop

## NAME

stop - stop the acquisition immediately

## DESCRIPTION

The command **stop** stops the acquisition immediately without writing the data to disk. 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. Further, it can be used to stop the command **gs**. 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.

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, zg, go, rga, wobb



# 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 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**.

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, zg, go

## **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.

### **OUTPUT FILES**

<du>/data/<user>/nmr/<name>/<expno>/

fid - 1D raw data

acqus - acquisition status parameters

### **SEE ALSO**

zg, go

## **xau, xaua, xaup**

### **NAME**

xau - execute an AU program  
xaua - execute the AU program specified with AUNM  
xaup - execute the AU program specified with AUNMP

### **SYNTAX**

xau [<name>]  
xaua  
xaup

### **DESCRIPTION**

The command **xau** allows you to execute an AU program. It is not used very often because AU programs are normally executed simply by entering their names. The command **xau** is needed in the following cases:

- the AU program has not been compiled yet
- an XWIN-NMR command with the same name exists
- to call an Au program from an AU program (using the macro XAU)

Note that usually AU programs have already been compiled with **edau**, **compileall**, **cp1bruk** or **cp1user**. Furthermore, it is not recommended to give an AU program the same name as an XWIN-NMR command. Before you start writing a new AU program, just type in the name you want to give it to find out if this name is already in use.

AU programs can be executed in three different ways:

**xau**

a list of all AU programs appears, you can click one to execute it

**xau <name>**

compiles the AU program <name> (if it is not compiled yet) and executes it

**<name>**

executes the AU program <name> (if it is compiled)

Furthermore, you can use the commands **xaua** and **xaup** to execute AU programs. These commands take no argument but 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 (ICON-NMR), the AU programs specified by AUNM and AUNMP perform the acquisition and the processing, respectively.

AU programs run in background and several of them can run simultaneously. You can use the command *follow* to see which AU programs and which part within each AU program are currently running. The command *kill* can be used to stop a running (or hanging) AU program.

For details on writing, compiling, and executing AU programs please refer to the XWIN-NMR AU reference manual (click *Help* → *Other topics* → *Writing AU programs*).

## 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

<xwhome>/exp/stan/nmr/au/src/

AU program source files (only input if the AU program is not compiled yet)

<xwhome>/prog/au/bin/

AU program binary executables

<du>/data/<user>/<name>/nmr/<expno>/

acqu - acquisition parameters (input file for *xaua*)

<du>/data/<user>/<name>/nmr/<expno>/pdata/<procno>/

proc - processing parameters (input file for *xaup*)

## USAGE IN AU PROGRAMS

XAU(name)

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 in this respect.

**SEE ALSO**

edau, delau, renau, expinstall, compileall, cplbruk, cpluser

## **zg**

### **NAME**

*zg* - zero go: perform an acquisition

### **DESCRIPTION**

The command **zg** performs an acquisition on the current dataset, overwriting possibly existing data. This involves the following sequence of actions:

1. Read the acquisition parameters
2. Compile the pulse program defined by the acquisition parameter PUL-PROG
3. 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
4. Update the acquisition status parameters

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 probehead

**edprosol** - define the probehead and solvent dependent parameters

**edc** or **new** - define a new dataset and make it the current dataset

**wobb** or **atma** - tune and match the probehead (**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 experiment you want to do

**getprosol** - get the probehead 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

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.

After `zg` has been started, you can observe the running acquisition in the acquisition menu. You can go there by clicking *Acquire* → *Observe fid window* or by entering `acqu` on the command line. Here the FID is displayed and continuously updated. Furthermore, an information window will appear showing the current scan, the total number of scans and the residual experiment time. You can go to the acquisition menu before or after you enter the command `zg`.

The purpose of dummy scans is to reach steady state conditions concerning  $T_1$  relaxation. This is necessary whenever the recycle delay of the experiment is shorter than 4 times the  $T_1$  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. The `zd` statement disables the execution of dummy scans.

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 dimension(s). Note that in XWIN-NMR 3.0 or later, Bruker 2D and 3D pulse program do not contain a **wr** statement. It has been replaced by the **mc** statement which performs both the disk write and loop back function.

**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. At the time of this writing, XWIN-NMR supports 1D, 2D and 3D datasets. However, the pulse program is not limited to 3D and allows you to acquire data in 4 or more dimensions. The resulting raw data (the **ser** file) can be processed by third party software that supports 4 or more dimensions. The parameters for the fourth and higher dimensions must be created manually.

Normally, an acquisition will run until it has finished. If, however, you want to interrupt it, you can do that with the command **halt**. This will finish the current scan, write the data to disk and then stop the acquisition. Note that **halt** finishes the current scan but not the current phase cycle. The command **stop** immediately stops the acquisition. It does not finish the current scan or write the data to disk. Instead of entering **stop**, you can also click the **stop** button. 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 you find out that 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**).

**zg** overwrites possibly existing raw data. Each user can configure XWIN-NMR to



get a warning about existing data or not. Enter the command **setres** and set the flag *ZG*safety to *on* to get a warning. 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. This will automatically switch the display to the processing menu (if it wasn't already there) and show you the spectrum. 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\_get1\*** and **multizg**. Furthermore, **zg** is automatically executed when experiments are started from the automation interfaces ICON-NMR and **buttonnmr**.

Note that an acquisition always runs in background which means that XWIN-NMR can do other tasks simultaneously. You can change the display to a different dataset and process data there. This is actually what happens all the time during an automation sequence performed by ICON-NMR.

XWIN-NMR 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* → *Other topics* → *Writing pulse programs*

## INPUT PARAMETERS

all acquisition parameters as described in chapter 2

## OUTPUT PARAMETERS

all acquisition status parameters as described in chapter 2

## INPUT FILES

### 1D data

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

### 2D data

<du>/data/<user>/nmr/<name>/<expno>/

acq1 - acquisition parameters direct dimension (F2)

acq2 - acquisition parameters indirect dimension (F1)

### 3D data

<du>/data/<user>/nmr/<name>/<expno>/

acq1 - acquisition parameters direct dimension (F3)

acq2 - acquisition parameters indirect dimension (F2)

acq3 - acquisition parameters indirect dimension (F1)

### All data

<xwhome>/exp/stan/nmr/lists/pp

<PULPROG> - the pulse program defined by PULPROG

<xwhome>/conf/instr/<curinst>/

scon - spectrometer parameters (created with *edscon*)

## OUTPUT FILES

### 1D data

<du>/data/<user>/nmr/<name>/<expno>/

fid - raw data

acqus - acquisition status parameters

### 2D data

<du>/data/<user>/nmr/<name>/<expno>/

ser - raw data

acqus - acquisition status parameters direct dimension (F2)

acq2s - acquisition status parameters indirect dimension (F1)

### 3D data

<du>/data/<user>/nmr/<name>/<expno>/

ser - raw data

acqus - acquisition status parameters direct dimension (F3)

acq2s - acquisition status parameters indirect dimension (F2)

acq3s - acquisition status parameters indirect dimension (F1)

**All data**

<du>/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

gradprog - shaped gradient definition (output if a ngrad statement is used)

**USAGE IN AU PROGRAMS**

ZG

**SEE ALSO**

go, gs, rga, wobb, atma, suspend, resume



---

# Chapter 10

## Temperature commands

---

This chapter describes commands which configure, control or monitor the temperature unit. A temperature unit which are delivered with Bruker spectrometers is 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.

## **edte**

### **NAME**

edte - edit the temperature parameters

### **DESCRIPTION**

The command **edte** opens a dialog box from which the temperature unit can be controlled. The functionality of this window is described in a separate manual that can be opened from the XWIN-NMR window by clicking **Help** → **Other topics** → **Temperature regulation** or from the **edte** window by clicking **Help** → **Edte Users Manual**

### **SEE ALSO**

edtg, tepar, temon, teget, te2get, treset, te2set, teready, te2ready

# edtg

## NAME

edtg - edit the temperature parameters for gradient temperature units

## DESCRIPTION

The command **edtg** opens a dialog box from which the gradient temperature unit can be controlled. The functionality of this window is described in a separate manual that can be opened from the **edtg** window by clicking **Help** → **Edtg Users Manual**

## SEE ALSO

edte, tepar, temon, teget, te2get, terset, te2set, teready, te2ready

## 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.

**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

<du>/data/<user>/nmr/<name>/<expno>/

acqus - acquisition status parameters

### USAGE IN AU PROGRAMS

TEGET

TE2GET

### SEE ALSO

edte, tepar, temon, terset, te2set, teready, te2ready



# temon

## NAME

temon - temperature monitor

## DESCRIPTION

The command **temon** opens a temperature monitor window which shows:

- the last measured temperature
- the time the last temperature was measured
- the minimum and maximum temperature since **temon** was started

The **temon** window offers the following buttons:

**Mode** - switches the display mode and shows the maximum and minimum temperature and the time at which they were measured

**Celsius/Kelvin** - switches between Celsius and Kelvin

### **LogFile**

opens a new window with the following buttons:

#### **Write log file ON/OFF**

a radio button which allows you to switch on/off the writing to a log file

#### **Time Interval**

allows you to set the interval in which the logfile is updated. A list of possible time intervals appears and you must double click a value to select it.

#### **print log file**

prints the log file to the system default printer

#### **OK**

closes the LogFile window

**Close** - close the **temon** window

A graphical temperature monitor can be started by clicking:

**Data → Monitor temp. & Output Power**

from the **edte** dialog window.

**OUTPUT FILES**

<userhome>/xwinnmr-<hostname>/

temon - last measured temperature

<du>/data/<user>/nmr/<name>/<expno>/

temperatureLog - temperatures measured at regular intervals

**SEE ALSO**

edte, edtg, tepar, teget, te2get, teset, te2set, teready, te2ready

# tepar

## NAME

tepar - get a predefined set of temperature unit parameters

## SYNTAX

tepar [<name>]

## DESCRIPTION

The command **tepar** gets a predefined set of temperature unit parameters. These are parameters which have been previously stored from the **edte** window by clicking *File* → *Save configuration*.

The command **tepar** takes one argument and can be used in one of the following ways:

**tepar**

shows a list of available temperature parameter sets. When you click one, it is loaded to the temperature unit.

**tepar <name.tcf>**

loads the temperature parameter set <name> to the temperature unit.

## INPUT FILES

<xwhome>/exp/stan/nmr/lists/eurotherm/tcf/\*

temperature parameter sets

## USAGE IN AU PROGRAMS

TEPAR(filename)

## SEE ALSO

edte, edtg, temon, teget, te2get, terset, te2set, teready, te2ready

## teready, te2ready

### NAME

teready - wait until the temperature set with *teset* has been reached

te2ready - wait until the temperature set with *te2set* has been reached

### SYNTAX

teready <stabilization time> <precision>

te2ready <stabilization time> <precision>

### DESCRIPTION

The command **teready** waits until the demand temperature has been within the specified precision range for at least 10 seconds. If this does not happen within the specified stabilization time, **teready** terminates. **teready** is typically executed after setting the target temperature with the command **teset**. For example, the command sequence:

```
teset 320.0
```

```
teready 300 0.5
```

sets the temperature to 320.0 K and waits until this has been reached approximately. More precisely, it waits until the temperature has been between 319.5 and 320.5 for at least 10 seconds. After that, **teready** waits another 300 seconds to allow for temperature stabilization of the sample.

The stabilization time (in seconds) and precision (in Kelvin) must be specified as arguments, they are not prompted for if they are omitted.

**te2ready** works like **teready** except that it controls the so called second temperature on a temperature unit with two regulators. These 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.

### USAGE IN AU PROGRAMS

TEREADY(time, precision)

TE2READY(time, precision)

**SEE ALSO**

edte, edtg, tepar, temon, teget, te2get, treset, te2set

## 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**.

**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 temperature on the temperature unit (input of **te2set**)

### INPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acqu - acquisition parameters

## **USAGE IN AU PROGRAMS**

TESET

TE2SET

VT

executes the command *teset* <*list\_index*>

## **SEE ALSO**

edte, edtg, tepar, temon, teget, te2get, teready, te2ready





---

# Chapter 11

## MAS/Solids commands

---

This chapter describes commands which are used for solid state experiments. In most of them magic angle spinning is used. As such, most of the commands described here concern the control of the MAS pneumatic unit. Furthermore, commands for controlling the high power control unit (HPCU) are described. Note that this unit is used on Avance-AQX but not on Avance-AQS. On the latter, the HPCU commands are obsolete.

## cfmas

### NAME

cfmas - configure the MAS pneumatic unit

### DESCRIPTION

The command **cfmas** allows you to configure the MAS pneumatic unit. It will prompt you for the rs232 channel and for the following information:

- *Minimum main pressure* (default 4000 mBar)
- *Insert air on time* (default 10 sec)
- *Eject air on time* (default 10 sec)
- *Sample diameter* (Wide or Normal)  
This actually concerns the type of eject mechanism. For probes with a tilting stator you must enter N(ormal), for probes with a fixed stator W(ide).
- *SpinLock tolerance* (default 5 Hz)  
For solid state MAS, the default value of 5 Hz can be accepted. For HR MAS, a value between 5 and 90 must be entered. The Spinlock tolerance refers to the required precision of the demand spin rate. If, for example, you specify 10 Hz and turn on the spinning, the spin rate will be adjusted until it has reached the demand spin rate plus or minus 10 Hz and stayed there for at least 5 seconds.

**cfmas** can also be executed as a part of the XWIN-NMR configuration suite (command **config**). When a hardware list is used (see **cf**), the MAS unit can be specified there. In that case, the command **cfmas** is not needed because the MAS unit is configured by **cf**.

### INPUT AND OUTPUT FILES

<xwhome>/conf/instr/<instrum>/rs232\_device/

mas - serial port for MAS unit

<xwhome>/conf/instr/<instrum>/

mas\_params - MAS parameters

**SEE ALSO**

cf, config, mas, mascontrol, masr, masi, masg, mash, mase

## edhpcu

### NAME

edhpcu - edit the acquisition plus HPCU parameters

### DESCRIPTION

The command **edhpcu** opens a dialog window with all acquisition parameters and all HPCU parameters. The former corresponds to the **eda** parameters. The latter involve the gain parameters HGAIN1 to HGAIN4 and XGAIN1 to XGAIN4 and the tune parameters TUNHIN, TUNHOUT and TUNXOUT.

### INPUT AND OUTPUT PARAMETERS

all acquisition parameters and the following HPCU parameters:

HGAIN1 - HGAIN4 - gain level 1- 4 for 1H/19F tube transmitter  
XGAIN1 - XGAIN4 - gain level 1- 4 for X nucleus tube transmitter  
TUNHIN -1H/19F high power tube transmitter input tuning  
TUNHOUT -1H/19F high power tube transmitter output tuning  
TUNXOUT - X nucleus high power tube transmitter output tuning

### INPUT AND OUTPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

### SEE ALSO

gethpcu, sethpcu, rackpow

## ed4ph

### NAME

ed4ph - edit the 4-phase modulator parameters

### DESCRIPTION

The command **ed4ph** opens a dialog box where you can set the parameters for the 4-phase modulator. This unit is mainly used for solid state experiments. This allows for a much faster 90° phase switching (+x, +y, -x, -y).

The following parameters are offered:

FX\_AX - F1 channel amplitude of +x

FX\_AMX - F1 channel amplitude of -x

FX\_PY - F1 channel phase of +y

FX\_PMY - F1 channel phase of -y

FX\_AY - F1 channel amplitude of +y

FX\_AMY - F1 channel amplitude of -y

FX\_PMX - F1 channel phase of -x

For the F2 channel, the corresponding parameters are called FH\_\*. For the F3 channel, the corresponding parameters are called FY\_\*. Note that not all spectrometers are equipped with the F3 channel. Furthermore, some spectrometers do have an F3 channel but no 4-phase modulator because the frequency is directly created on the synthesizer.

RECPHASE - receiver phase

When you enter a value in any of the above parameters and hit the return key, the entered value is immediately set on the 4-phase modulator.

At the bottom of the dialog box, the following push buttons are available:

**QUIT**

closes the dialog box

**LIST**

unused

***SaveToFile***

allows you to store the current values to disk. A list of existing file appears or you can enter a new filename.

***LoadParFile***

allows you to read a previously stored values from disk. A list of existing file appears.

***ReadAllPar***

unused

The regular way of phase switching, as it occurs in most high resolution experiments, is the use of a phase program like:

```
ph1 = 0 90 180 270
```

The phases are provided by the spectrometer FCU and require a certain time to be set, typically 3 $\mu$ sec. If you replace this phase program by the following one:

```
ph1 = +x +y -x -y
```

the same phase cycle is used but now the phases are provided by the 4-phase modulator. The switching time is now as short as 50nsec.

The 4-phase modulator can be use with or without an HPCU. In the former case, it must be connected to the 4-PH connector of the HPCU. In the latter case, it must be connected to the second *rs485* channel of the CCU and must be specified in the hardware list (see also **CF**). Note that spectrometers which are equipped with transistorized amplifiers rather than tube amplifiers are not equipped with (and do not need) an HPCU.

The 4-phase modulator can only be used for hard pulses, not for shaped pulses. On Avance-AQS spectrometers, the 4-phase modulator is not needed and, as such, not supported. The reason is that the frequency generation on the SGU doesn't cause any phase switching delays.

**OUTPUT FILES**

```
/xwhome/exp/stan/nmr/lists/4ph/
```

**SEE ALSO**

```
edhpcu
```

## **mas**

### **NAME**

mas - open the MAS control window

### **DESCRIPTION**

The command **mas** opens a dialog box where you can set the MAS parameters and control the pneumatic unit. The following entries are available:

#### ***Probe Setup Filename***

shows a list of available MAS parameter files. These are files that have been stored previously from the **mas** window and contain the probehead type.

#### ***Bearing Gas Port***

This button is only used for the older heat exchanger types that use 3 input lines. It allows you to select demand temperature. Possible values are *Ambient (default)*,  $-70^{\circ}\text{C}$  or  $-120^{\circ}\text{C}$ . After selecting a temperature, this is set on the pneumatic unit.

#### ***Probehead Selection***

a set of toggle buttons that allow you to set the following parameters:

- *Diameter*: sample diameter
- *Probe*: probehead type
- *Type*: magnet type, used to define tilt or no tilt of the stator (see ***Insert sample*** below)
- *Material*: rotor material
- *User*: unused

Clicking these buttons will toggle between their possible values. Only values that correspond to existing hardware are available. Together, the selected values will determine the algorithms that are used to spin the sample. When you enter **mas**, the current values are read from the MAS pneumatic unit.

***Main Pressure, Bearing Pressure, Drive Pressure, Spin Rate***

Each of these parameters shows three values:

**Demand:** field showing the current demand value. If you enter a new value, this is set on the pneumatic unit.

**Actual:** push buttons showing the actual value on the pneumatic unit. Note that they are only updated when clicked. They are, however, continuously updated, if the button *ContinuousUpdate* is highlighted

**Maximum:** not implemented.

The parameters are read from the MAS pneumatic unit when you enter the **mas** command. See also *Setting Mode*.

### **Setting Mode**

The MAS unit can work in two different modes:

#### **Manual mode**

you must enter the **Bearing Pressure** and **Drive Pressure**. The **Spin Rate** will be automatically adjusted.

#### **Automatic mode**

you must enter the **Demand Spin Rate**. The **Drive pressure** and **Bearing Pressure** will be automatically adjusted.

### **Spinning**

Allows you to switch the spinning on or off. Caution: after entering the **Insert Sample** command (see below), you must wait at least the time specified by **Insert Airtime** before you switch on the spinning.

### **Spin Locked**

Status flag that indicates whether or not the spin rate is locked. Spin Locked is automatically set to *yes* when the spin rate has reached the demand spin rate within a range of plus or minus 5 Hz and stayed there for more than 5 seconds. Spin Locked only works in Automatic mode.

### **StartupProgram**

Unused. It is automatically defined by **Probehead Selection** (see above).

### **Insert Air Time**

Allows you to modify the Insert Air Time. By default, the value that was set with **cfmas** is used. Note that the value entered here is not stored on



disk. When you exit the **mas** window, it is set back to the value that was entered with **cfmas**.

### ***Eject Air Time***

Allows you to modify the Eject Air Time. By default, the value which was set with **cfmas** is used. Note that the value entered here is not stored on disk. When you exit the **mas** window, it is set back to the value that was entered with **cfmas**.

### ***Insert sample***

Before you click this button you must insert the spinner into the sample transfer tube. The effect depends on the probehead type. For standard bore probes, which tilt the stator for sample change, a period specified by Insert Airtime is waited and then the stator is tilted back to the magic angle. For probes where the stator is not tilted (all wide bore and some standard bore probes) an air flow through the insert line pushes the spinner into the stator. Standard bore probes that are not tilted are for example, all probes with 2.5 mm diameter and all probes for spectrometer  $\geq 700$  MHz.

### ***Eject sample***

ejects the sample from the stator. Before you click this button, you must stop the spinning.

### ***Continuous Update***

causes the parameters *Main Pressure*, *Bearing Pressure*, *Drive Pressure* and *Spin Rate* to be continuously updated, i.e. read from the MAS pneumatic unit.

### ***Save***

saves the current parameters to a Probe Setup Filename

### ***Edit***

allows you to edit Probe Setup Files

### ***List***

unused

### ***Cancel***

exits the **mas** dialog box.

**INPUT FILES**

<xwhome>/exp/stan/nmr/lists/mas/

Probe Setup Files (input of *mas* → *Probe Setup Filename*)

<xwhome>/conf/instr/<instrum>/

mas\_params - MAS parameters

**OUTPUT FILES**

<xwhome>/exp/stan/nmr/lists/mas/

Probe Setup Files (output of *mas* → *Save*)

**SEE ALSO**

mascontrol, cfmas, masrmon, masr, masi, masg, mash, mase

# mascontrol

## NAME

mascontrol - open the MAS monitor and control window

## DESCRIPTION

The command *mascontrol* opens a dialog box where you can set the MAS parameters and monitor the MAS pneumatic unit. It can be used as an alternative to the commands *mas* and *masrmon*.

*mascontrol* starts by opening a small window where you can select the probehead. After doing that, a new window appears where the current MAS spin rate is displayed and the following buttons are available:

### *Mode*

When you click this button, the MAS spin rate is shown and new additional buttons will appear:

*insert* (for wide bore probes and standard bore probes without stator tilt)  
Before you click this button, you must insert the spinner into the sample transfer tube. The action of *insert* depends on the probehead type. For standard bore probes, which tilt the stator for sample change, *insert* waits a period specified by Insert Airtime and then tilts the stator back to the magic angle. For probes where the stator is not tilted (all wide bore and some standard bore probes), *insert* causes an air flow through the insert line that pushes the spinner into the stator.

*set angle* (for standard bore probes with stator tilt)  
See *insert*.

*eject*  
ejects the sample from the probehead. It switches the air flow on for a time which has been defined with *cfmas*.

*start*  
brings the sample rotation up to the demand spin rate.

*stop*  
stops sample rotation.

*etc.*

When you click this button, a new window appears with the radio button:

***Write log file ON/Write log file OFF***

switches the writing of the log file on/off. The log file is stored under the EXPNO of the dataset where ***mascontrol*** was started.

and the following push buttons:

***Screen Update***

allows you to set the screen update interval (seconds). A list of possible time intervals appears and you must double click a value to select it.

***Logfile time interval***

allows you to set the interval at which the logfile is updated. A list of possible time intervals appears and you must double click a value to select it.

***Select new probe***

opens a new window with all available probeheads will appear. When you click one, the ***mas*** dialog window will be opened where you can read a new Probehead Setup Filename or select a new probehead.

***Set new spin rate***

opens a new dialog window where you can change the spin rate.

***Show log file***

shows the MAS log file on the screen.

***Print log file***

prints the log file on the system default printer.

## INPUT AND OUTPUT PARAMETERS

can be viewed with ***dpa*** or by typing ***ls masr*** :

MASR - MAS spin rate

## INPUT FILES

<xwhome>/prog/tcl/xwish3\_scripts/

mascontrol - mascontrol Tcl/Tk startup script

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters

**OUTPUT FILES**

<du>/data/<user>/nmr/<name>/<expno>/

acqus - acquisition status parameters

masRateLog - spin rate log file

**SEE ALSO**

mas, masrmon, cfmas, masr, masi, masg, mash, mase

## **masr, masi, masg, mash, mase**

### **NAME**

masr - set the spin rate  
masi - insert the spinner  
masg - start spinning  
mash - halt the spinner  
mase - eject the spinner

### **DESCRIPTION**

MAS control like sample insert and eject, sample rotation etc. is usually done from one of the MAS control windows (commands *mas* or *mascontrol*). It can, however, also be done with the commands described on this page.

#### ***masr***

Prompts you for the demand spin rate and then sets the specified value on the MAS pneumatic unit. It also sets the acquisition parameter MASR.

#### ***masr get***

reads the actual spin rate from the pneumatic unit and stores it into the acquisition status parameter MASR.

#### ***masr acqu***

sets the demand spin rate on the MAS pneumatic unit according to the acquisition parameter MASR.

#### ***masi***

Before you enter this command you must insert the spinner into the sample transfer tube. The action of *masi* depends on the probehead type. For standard bore probes, which tilt the stator for sample change, *masi* waits a period specified by Insert Airtime and then tilts the stator back to the magic angle. For probes where the stator is not tilted (all wide bore and some standard bore probes), *masi* causes an air flow through the insert line that pushes the spinner into the stator.

#### ***masg***

Starts the sample rotation according to the demand spin rate.

***masg wait*** shows the values of Lock Status (LoSt), Spin Rate (VA), Driver Pressure (PD) Bearing Pressure (PB), Bearing Sense Pressure (PS). It blocks

all other MAS commands until Lock Status is 1. It is typically used in automation (see the macro **MASG** below).

***mash***

Stops the sample rotation.

***mash wait*** does the same but blocks all other MAS commands until the spin rate has become zero. It is typically used in automation (see the macro **MASH** below).

***mase***

Ejects the sample from the probehead. It switches the air flow on for a time which has been defined with ***cfmas***.

Note that all these commands, except for ***masr get***, are ignored when an acquisition is running. No error message is displayed.

More information on MAS is available in some manuals which can be opened from XWIN-NMR by clicking ***Help*** → ***Other topics***:

→ ***Solids Users Manual***

→ ***Help*** → ***Other topics*** → ***SB/MAS Manual***

→ ***BASH spectrometer documentation***

Note that the last manual requires the BASH CDROM.

**INPUT PARAMETERS**

MASR - MAS spin rate (input of ***masr acqu***)

**OUTPUT PARAMETERS**

MASR - MAS spin rate (acquisition status parameter, output of ***masr get***)

MASR - MAS spin rate (acquisition parameter, output of ***masr***)

**INPUT FILES**

<du>/data/<user>/nmr/<name>/<expno>/

acqu - acquisition parameters

<xwhome>/conf/instr/<instrum>/

mas\_params - MAS parameters (input of ***masi*** and ***mase***)

**USAGE IN AU PROGRAMS**

MASI

MASE

MASR

executes the command *masr acqu*

MASRGET

executes the command *masr get*

MASH

executes the command *mash wait*

MASG(retries)

executes the command *masg wait n*, where *n* is the number of retries

**SEE ALSO**

cfmas, mas, mascontrol, masrmon



## **masrmon**

### **NAME**

masrmon - monitor the MAS spin rate

### **DESCRIPTION**

The command **masrmon** monitors the MAS spin rate. It opens a dialog box with the following buttons:

#### ***LogFile***

opens a new window with the following buttons:

#### ***Write log file ON/OFF***

a radio button which allows you to switch on/off the writing to a log file

#### ***Time Interval***

allows you to set the interval in which the logfile is updated. A list of possible time intervals appears and you must double click a value to select it.

#### ***print log file***

prints the log file to the system default printer

#### ***OK***

closes the LogFile window

#### ***Mode***

allows you to toggle between the following display modes:

- the mode showing the current spin rate
- the mode showing the current, maximum and minimum spin rate and the times at which they occurred.

#### ***Close***

closes the MAS monitor window

### **OUTPUT PARAMETERS**

can be viewed with **dpa** or by typing **ls masr :**

MASR - MAS spin rate

**OUTPUT FILES**

<du>/data/<user>/nmr/<name>/<expno>/

acqus - acquisition status parameters

masRateLog - spin rate log file

**SEE ALSO**

mas, mascontrol, cfmas, masr, masi, masg, mash, mase

## rackpow, gethpcu, sethpcu

### NAME

rackpow - switches the high power cabinet on/off

gethpcu - reads the gain and I/O tuning values of tube amplifiers from the HPCU

sethpcu - sets the gain and I/O tuning values of tube amplifiers on the HPCU

### DESCRIPTION

The command **rackpow** switches the high power cabinet on or off. This is done according to the acquisition parameter POWMOD. For POWMOD = low, it is switched off, for POWMOD = high, it is switched on.

The command **gethpcu** reads the gain from the HPCU and stores them into the acquisition parameters HGAIN1 to HGAIN4, XGAIN1 to XGAIN4. Furthermore, it reads the I/O tuning values of tube amplifiers from the HPCU and stores them into the acquisition parameters TUNHIN, TUNHOUT and TUNXOUT.

The command **sethpcu** sets the gain from the HPCU according to the acquisition parameters HGAIN1 to HGAIN4, XGAIN1 to XGAIN4. Furthermore, it sets the I/O tuning values of tube amplifiers on the HPCU according to the acquisition parameters TUNHIN, TUNHOUT and TUNXOUT.

### INPUT AND OUTPUT PARAMETERS

HGAIN1 - HGAIN4 - gain level 1-4 for the 1H/19F tube transmitter

XGAIN1 - XGAIN4 - gain level 1-4 for the X nucleus tube transmitter

TUNHIN - 1H/19F high power tube transmitter input tuning

TUNHOUT - 1H/19F high power tube transmitter output tuning

TUNXOUT - X nucleus high power tube transmitter output tuning

### INPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters (input for **rackpow** and **sethpcu**)

### OUTPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/

acq - acquisition parameters (output of **gethpcu**)

**USAGE IN AU PROGRAMS**

RACKPOW  
SETHPCU  
GETHPCU

**SEE ALSO**

edhpcu

---

# Chapter 12

## Rename and delete commands

---

This chapter describes XWIN-NMR commands which allow you to rename or delete acquisition related files. These include parameter sets, pulse programs, gradient programs shim sets, AU programs, XWIN-NMR macros and various lists.

## delpar, delpul, delgp, delsh, delau, delmac

### NAME

delpar - delete parameter sets  
delpul - delete pulse programs  
delgp - delete gradient programs  
delsh - delete shim files  
delau - delete AU programs  
delmac - delete macros

### SYNTAX

delpar [<name>]

delpul, delgp, delsh, delau and delmac have the same syntax as delpar

### DESCRIPTION

The command **delpar** displays a list of parameter sets, both Bruker and user defined. Each entry shows the parameter set name and the parameter types in that set. You can mark one or more parameter sets for deletion and then click **Execute** to actually delete them. Furthermore, you can print the list by clicking the **Print** button. If you have accidentally removed Bruker parameter sets, you can re-install them with the command **expinstall**.

The other **del\*** commands mentioned here all work like **delpar**, deleting the type of files as specified above. Note that **delau** deletes both the selected AU source files and the corresponding executable files. If you accidentally delete Bruker AU programs, pulse programs or gradient programs, you can re-install them with **expinstall**. In case of AU programs, you must also compile them with **cpbruk** or **xau**.

### INPUT FOLDERS

<xwhome>/exp/stan/nmr/par - Bruker and user defined parameter sets

<xwhome>/exp/stan/nmr/lists/pp - Bruker and user defined pulse programs

<xwhome>/exp/stan/nmr/lists/gp - Bruker and user defined gradient programs

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

<xwhome>/exp/stan/nmr/lists/mac - Bruker and user defined macros

<xwhome>/exp/stan/nmr/au/src - Bruker and user defined source AU programs

<xwhome>/prog/au/bin - Bruker and user defined binary AU programs

<xwhome>/exp/stan/nmr/lists/mac - Bruker and user defined macros

## **USAGE IN AU PROGRAMS**

DELPAR(name)

No AU macros are available for the other **del\*** commands.

## **SEE ALSO**

dellist

# dellist

## NAME

dellist - delete various lists

## SYNTAX

dellist [<name>]

## DESCRIPTION

The command **dellist** displays a list of various lists types. Most of them are used in acquisition, e.g. *vd* lists contain variable delays which are read by the *vd* statement in pulse programs. When you click a list type, the available files of that type appear. You can click individual entries to mark them for deletion or click the button **Select all**. Clicking the **Execute** button deletes all marked entries.

All lists which appear with **dellist** can be printed by clicking the **Print** button.

## INPUT FILES

<xwhome>/exp/stan/nmr/lists

pp - pulse programs

cpd - CPD programs

gp - gradient programs

ds - dataset lists

vc - variable counter lists

vd - variable delay lists

vp - variable pulse lists

vt - variable temperature lists

f1 - frequency lists

f2, f3 - frequency lists (A\*X spectrometers only)

mac - XWIN-NMR macros

roi - 2D integral ranges

scl - scaling region files

masr - MASR rotation values



**SEE ALSO**

delpar, delpul, delgp, delsh, delau, delmac

## renau, renpar

### NAME

renau - rename AU programs  
renpar - rename parameter sets

### DESCRIPTION

The command **renau** opens a dialog box with all AU programs, both Bruker and user defined. You can rename one or more AU programs, simply by replacing their names. When you enter a new name and press the **Enter** key, the AU program is immediately renamed. This counts for both the source and the binary file. The dialog box can be closed by clicking *Cancel*.

**renpar** works like **renau**, except that it allows you to rename parameter sets. Parameter sets are directories, so **renpar** changes directory names; the names of the parameter files in those directories (*acqu, proc, meta, outd*) are not affected.

### INPUT AND OUTPUT FILES

For **renau**:

<xwhome>/exp/stan/nmr/au/src/\*  
<xwhome>/prog/au/bin/\*

For **renpar**:

<xwhome>/exp/stan/nmr/par/\*

### SEE ALSO

rengp, renlist, renlut, renmac, renpul

## renpul, rengp, renlut, renmac

### NAME

renpul - rename pulse programs  
rengp - rename gradient programs  
renlut - rename 2D lookup tables  
renmac - rename macros

### DESCRIPTION

The command **renpul** opens a dialog box with all pulse programs, both Bruker and user defined. You can rename one or more pulse programs, simply by replacing their names. When you enter a new name and hit the **Enter** key, the pulse program is immediately renamed. The dialog box can be closed by clicking **Cancel**.

In the same way, **rengp**, **renlut** and **renmac** allow you to rename, gradient programs, 2D lookup tables and macros respectively.

### INPUT AND OUTPUT FILES

<xwhome>/exp/stan/nmr/lists/pp/\*

pulse programs

<xwhome>/exp/stan/nmr/lists/gp/\*

gradient programs

<xwhome>/exp/stan/nmr/lut/\*

2D lookup tables

<xwhome>/exp/stan/nmr/lists/mac/\*

XWIN-NMR macros

### SEE ALSO

renlist, renau, renpar

## renlist

### NAME

renlist - rename various lists

### DESCRIPTION

The command ***renlist*** displays a list with various lists types, most of which are used in acquisition. When you click a list type, a dialog box appears with the available entries of that type. You can rename one or more entries, simply by replacing their names. When you enter a new name and hit the **Enter** key, the list is immediately renamed. The dialog box can be closed by clicking **Cancel**.

### INPUT AND OUTPUT FILES

<xwhome>/exp/stan/nmr/lists/\*

### SEE ALSO

renau, renpar, rengp, renlut, renmac, renpul

---

# Chapter 13

## Miscellaneous

---

This chapter describes XWIN-NMR commands that are somehow acquisition related and have not been described in previous chapters.

# pulsdisp

## NAME

pulsdisp - open the pulse program display

## DESCRIPTION

The command **pulsdisp** simulates the pulse sequence and shows it in a graphical representation. It uses the pulse program of the current dataset as it is defined by the acquisition parameter PULPROG. The graphical representation includes the pulse program timing, frequencies, phases and power levels.

The pulse program display can be started by entering **pulsdisp** on the command line or click **Windows** → **Pulse program**. First a small window will appear where you can select the length of the simulation and the elements to be displayed. the following fields/buttons are available:

- **Observe Setup**

This button to select the elements to be displayed. A new window will appear from which you can select the elements to be displayed. The following check buttons are available:

**CH<sub>n</sub>**

select the pulse channels to be displayed

**FCU<sub>n</sub>**

select the frequency unit for which you want to display the phases and/or power levels

**GCUX, GCUY, GCUZ**

select the X, Y and/or Z gradient output to be displayed

**RCUGO**

select receiver control to be displayed

**Receiver**

display the receiver status (on/off) and receiver phase

**Prog**

show the pulse program statement for each time segment

**Apply**

store the current settings of the Observe Setup window

**OK**

store the current settings of the Observe Setup window and quit

- **Time/Scans**

allows you to choose between the display of a certain time interval or a certain number of scans.

- **Start and Stop**

allows you to enter the simulation starting point and end point, respectively. The entered values will be interpreted as time values, if you selected **Time** in the previous step and as scan numbers if you selected **Scans**. You can also specify a negative number to simulate dummy scans. This, however, only works if the acquisition parameter DS is set to a positive value. If you checked one or more FCUn channels in the Observe Setup window, then the value in the **Start** field will be ignored. In that case, the simulation will always start with the first scan (scan 0 or time 0.0).

- **Run simulation**

starts the simulation. This usually takes a few seconds, depending on the pulse sequence and selected time interval or number of scans. When the simulation is finished, a new window will appear showing the graphical representation of the pulse sequence: the actual pulse program display. This consists of one row for each element that was selected in the Observe Setup window. You will see the exact timing of the entire sequence including the implicit delays that, for example, occur in the statement `g0=n`. The following buttons are available:

**Setup display:** clicking this button will open a new dialog box that is equivalent to the one opened by **Observe Setup**. It allows you to change the simulation selection, for the current simulation window. It will not affect the selection for possible other simulation windows nor does it affect the selection in the original **Observe Setup** window.

**Zoom in**

allows you to investigate the fine details of the pulse sequence. Particularly text labels indicating the length of durations, power levels in db or percent units, phases in degrees, or pulse program text will only appear after you zoom. Note that the timing sequence also shows the so called implicit delays. These are delays which do not appear as such in the pulse program but are introduced by statements like `g0=n`.

**Zoom out**

allows you to undo the effect of **Zoom in** and get an overview of the entire simulation.

***New display***

allows you to open an extra pulse display window which is a copy of the first one. You can, for example, use one window to zoom in and the other for an overview.

***Kill display***

allows you to close the current pulse display window.

***Print***

allows you to print the current pulse display window. A dialog box will be opened where you can print, print to file, preview the print and setup the printer. Note that, for a long simulation, you might want to use multiple sheets of paper. You can specify this in the ***Split to X\*Y Pages*** field in the ***Setup*** window.

If you checked one or more ***FCUn*** simulations in the Observe Setup window, and the pulse program has many phase changes, the simulation time might take very long. Furthermore, it can happen that the available memory limit is exceeded or the graphics gets too large and will be truncated (an error message will appear if this happens).

- ***New Display***
- ***Show program***  
show the contents of the pulse program
- ***Undo***  
undo the last change
- ***Quit***  
quit the pulse program display

The pulse display startup window also offers the following menu:

- ***File***
  - ***Quit***  
quite the pulse program display (same as the ***Quit*** button)
- ***View***
  - ***New display***
  - ***Kill displays***



### *Align displays*

- **Options**

#### *Scale setup*

allows you to select the time axis scaling mode; *cut* or *logarithmic*. *cut* scaling will result in a true representation of all duration in the pulse program shorter than the time constant TC. Durations that exceed TC, it will be represented with thick solid lines. TC can be specified in the field *Time constant*. Logarithmic scaling will cause a compressed representation of all durations. Nevertheless, since pulse lengths are usually a in the range of microseconds and delays in the range of milliseconds or seconds, a logarithmic representation can give a better overview.

#### *Observe setup*

same as the *Observe setup* button described above

#### *FCU simulator setup*

allows you to select the pulse power units (db, a logarithmic unit, or %, a linear unit).

#### *Printer setup*

allows you to select the printer and setup the print parameters for printing the pulse program display window.

- **Help**

Pulse program display Quick-Start Guide

If you change acquisition parameters or the pulse program while the pulse program display is open, you must quit and restart it. Just clicking *New display* or even *Run simulation* again will not account for the change.

## **INPUT FILES**

<xwhome>/prog/app-defaults

PulseDisplay - pulse program display resources

## **SEE ALSO**

spdisp

# spdisp

## NAME

spdisp - symbolic (graphical) pulse program display

## DESCRIPTION

The command **spdisp** opens a new window with a symbolic pulse program display. 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 NMR-SIM. The functionality of the program is described in the NMR-SIM manual. This can be opened by clicking **Help** from the window that is opened by **spdisp**. I

**spdisp** actually starts the AU program **showpp**, which, in turn, calls the pulse program display of the NMR-SIM program. The following commands are equivalent:

**spdisp <name>**

**showpp <name>**

**nmr sim -showp <name>**

**spdisp** can only start if the following conditions are met:

- the Bruker AU programs have been installed (with **expinstall**)
- the NMR-SIM program is installed (from the NMR Suite CDROM)

The difference between **spdisp** and **pulsdisp** is the following. **spdisp** gives a graphical representation of the pulse sequence and is well suited for desktop publishing. **pulsdisp** shows the pulse sequence in much greater detail and is primarily useful for the investigation of new pulse sequences.

## INPUT FILES

<xwhome>/exp/stan/nmr/au/

showpp - AU program started by **spdisp** (source file)

<xwhome>/prog/au/bin

showpp - AU program started by *spdisp* (binary file)

**SEE ALSO**

pulsdisp, nmrsim

# edacb

## NAME

edacb - edit the Amplifier Control Board parameters

## SYNTAX

edacb [<filename>]

## DESCRIPTION

The command **edacb** allows you to edit the ACB parameters and load them to the spectrometer hardware. The ACB parameters define the upper limits above which the power is switched off. These limits are used to protect the probehead in case too high power levels or pulse lengths are used. Normally the default limits can be used in which case you do not need to execute **edacb**. Only for special probeheads or transmitters, it might be useful to set sharper limits.

At the top of the dialog box, the current probehead (as defined with **edhead**) is displayed. Below that, the ACB parameters are listed. These are:

- pulse power: pulse power in W(att) or %
- pulse width: pulse length in ms (millisecond) or %
- duty cycle: the (pulse width)/period ratio in %
- reflected power: pulse power not reaching the probe in W(att) or %

where period is defined as the time between the start of two successive pulses. The current values of the ACB parameters are shown for each amplifier (as detected by **cf**). Furthermore, the minimum and maximum allowed values are displayed for each parameter. The pulse power can be entered with the unit W, without unit (same as W) or with the unit %. The latter is interpreted as is the percentage of the nominal power of the amplifier. The pulse width can be entered with the unit ms, without unit (same as ms) or with the unit %. The latter is interpreted as the percentage of the maximum pulse width as specified above the entered value.

Note that the ACB parameters are interdependent. For example, if you decrease the pulse power, you can increase the pulse width or, if you increase the period between the pulses, you can also increase the pulse width.

At the bottom of the dialog box, the following buttons are offered:

***Set parameters***

Sets the current values on the spectrometer hardware. You are first prompted for the NMR Superuser password.

***Save as..***

Saves the ACB values to disk, including minimum and maximum values and probehead identification. You are first prompted for the NMR Superuser password. Then a list of existing filenames and you can select one of them. Alternatively, you can enter a new filename.

***Load from..***

Allows you to read previously saved values from disk. A list of existing files will appear and you can select one of them. If the minimum/maximum values do not match your amplifiers (as detected by **cf**), or the probehead identification does not match the current probehead (as defined by **edhead**) you will get a warning. In that case you are prompted to cancel or continue the loading.

***Save***

Only active after the current values have been saved with **Save as..** Saves the ACB parameters to the filename displayed in the title bar of the dialog box.

***About***

Displays information about the **edacb** version and software support.

***Quit***

Closes the **edacb** dialog box discarding all changes.

The command **edacb** can also be entered with one argument; the name of an ACB parameter file. The values in the file are automatically loaded to the hardware. As such, the command **edacb <filename>** can also be used in automation.

**INPUT AND OUTPUT FILES**

`<xwhome>/conf/instr/<instrum>/acbpar/*`

ACB parameters files

**SEE ALSO**

cf

## **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 cryo-probe. Cryo probeheads 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 probeheads.

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

# getlim1d, getlcosy, getlxhco, getljres, getlinv

## NAME

getlim1d - determine the 1D spectral width according to integral ranges  
getlcosy - determine the 2D cosy spectral width according to 1D integral ranges  
getlxhco - determine the 2D xhco spectral width according to 1D integral ranges  
getljres - determine the 2D jres spectral width according to 1D integral ranges  
getlinv - determine the 2D invers spectral width according to 1D integral ranges

## DESCRIPTION

The **get1\*** commands listed above determine and set the optimum spectral width for the specified experiment types. The optimum spectral width is determined from one or two associated 1D datasets that are defined as the so called second and third dataset (see **edc2**). 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 **get1\*** commands determine the spectral width such that it includes all integral ranges, in other words, all signals.

**get1\*** commands are typically used in automation. They are called from AU programs like **au\_get11d**, **au\_getlcosy**, **au\_getlinv** and **au\_getlxhco**. These, in turn are called by ICON-NMR where the preparation and the actual experiment are defined as a so-called 'composite' 2D experiment (see ICON-NMR 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.

Note that F2-SW refers to the acquisition parameter SW in the F2 dimension and F1-SW refers to the same parameter in the F1 dimension.

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. Both can be defined with the command **edc2**.

## OUTPUT PARAMETERS

SW - spectral width in ppm

## INPUT FILES

<du>/data/<user>/nmr/<name>/<expno>/pdata/<procno>

intrng - integral regions

curdat2 - definition of second and third dataset

## OUTPUT FILES

<du>/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



---

# Chapter 14

## NMR Suite files

---

This chapter describes the files which are involved in XWIN-NMR acquisition. For each file, the commands that typically create or modify the file are specified. If no command is specified, the file is delivered with XWIN-NMR and not modified by any command. Furthermore, the commands that typically interpret a file are specified. If no command is specified, the file can usually be read from a UNIX shell or Windows Explorer or is to be used by an external program. Note that the specified commands can be started, manually, from the XWIN-NMR command line or, automatically, from an AU program or ICON-NMR. Files that are created by ICON-NMR are not described here but in the ICON-NMR manual. Note that, for each file, only the most important commands that access the file are mentioned here.

For each file, the file type is specified which has one of the following letters:

- *a* - ascii file. It can be opened by a text editor.
- *j* - JCAMP-DX file. It can be opened with a text editor and interpreted by any software which supports JCAMP-DX.
- *b* - binary file. Data file with consecutive 32-bit integer values.
- *e* - binary executable
- *d* - directory with files and/or sub directories
- *t* - Tcl/Tk script

## 1D data files

File	Created or modified by <sup>a</sup>	Interpreted by	Description (file type)
<code>&lt;du&gt;/data/&lt;user&gt;/nmr/&lt;name&gt;/&lt;expno&gt;</code>			
<i>fid</i>	<b>zg, genfid</b>	<b>bc, em, ft</b>	1D raw data (b)
<i>acqu</i>	<b>rpar, eda</b>	<b>zg</b>	acquisition parameters (j)
<i>acqu</i> s	<b>rpar, zg</b>	<b>dpa, lpa</b>	acquisition status parameters (j)
<i>audita.txt</i>	<b>zg</b>		acquisition audit trail (j)
<i>pulse_</i> <i>program</i>	<b>zg</b>		precompiled pulse program (j)
<i>format.lpa</i>	<b>zg</b>	<b>lpa</b>	acquisition parameter format (a)
<i>format.temp</i>	<b>zg</b>	<b>plot, view</b>	acquisition parameter format (a)
<i>relations</i>	<b>getprosol</b>		prosol-acquisition relations (a)
<i>cpdprg1</i>	<b>zg</b> (cpd1) <b>zg</b> (cpds1)		CPD program defined by CPDPRG1 (a)
<i>cpdprg2</i>	<b>zg</b> (cpd2) <b>zg</b> (cpds2)		CPD program defined by CPDPRG2 (a)
<i>etc.</i>			
<i>scon</i>	<b>zg</b>		copy of the <b>edscon</b> parameters (j)
<i>spsnam1</i>	<b>zg</b> (sp1)		shaped pulse definition (j)
<i>spsnam2</i>	<b>zg</b> (sp2)		shaped pulse definition (j)
<i>etc.</i>			
<i>fq1list</i>	<b>zg</b> (fq1)		variable frequency list (j)
<i>fq2list</i>	<b>zg</b> (fq2)		variable frequency list (j)
<i>etc.</i>			
<i>vclist</i>	<b>zg</b> (vc)		variable counter list (a)
<i>vdlist</i>	<b>zg</b> (vd)		variable delay list (a)
<i>vplist</i>	<b>zg</b> (vp)		variable pulse list (a)
<i>vtlist</i>			variable temperature list (a)
<i>popt.array</i>	<b>popt</b>	<b>popt</b>	parameter optimization setup (a)

**1D data files**

<b>File</b>	<b>Created or modified by<sup>a</sup></b>	<b>Interpreted by</b>	<b>Description (file type)</b>
<i>popt.protocol</i>	<i>popt</i>	<i>popt</i>	parameter optimization result (a)
<i>masRateLog</i>	<i>masrmon,</i> <i>mascontrol</i>		MAS spin rate log file (j)
<i>cag_par</i>	<i>zg</i>		gradient rotation matrices

a. Some files are only created if the corresponding pulse program statement (specified between brackets) is used.

**2D data files**

<b>Filename</b>	<b>Created or updated by</b>	<b>Interpreted by</b>	<b>Description (file type)</b>
<i>&lt;du&gt;/data/&lt;user&gt;/nmr/&lt;name&gt;/&lt;expno&gt;</i>			
<i>ser</i>	<i>zg, genser</i>	<i>xfb, xf2</i>	raw data - series of FIDs (b)
<i>acqu</i>	<i>rpar, eda</i>	<i>zg</i>	F2 acquisition parameters (j)
<i>acqu2</i>	<i>rpar, eda</i>	<i>zg</i>	F1 acquisition parameters (j)
<i>acqu<sub>s</sub></i>	<i>rpar, zg</i>	<i>dpa</i>	F2 acquisition status parameters (j)
<i>acqu<sub>2s</sub></i>	<i>rpar, zg</i>	<i>dpa</i>	F1 acquisition status parameters (j)
Various lists as specified under 1D data files.			

## 3D data files

Filename	Created or modified by	Interpreted by	Description (file type)
<code>&lt;du&gt;/data/&lt;user&gt;/nmr/&lt;name&gt;/&lt;expno&gt;</code>			
<i>ser</i>	<i>zg</i>	<i>tf3</i>	raw data - series of FIDs (b)
<i>acqu</i>	<i>rpar, eda</i>	<i>zg</i>	F3 acquisition parameters (j)
<i>acqu2</i>	<i>rpar, eda</i>	<i>zg</i>	F2 acquisition parameters (j)
<i>acqu3</i>	<i>rpar, eda</i>	<i>zg</i>	F1 acquisition parameters (j)
<i>acqu3s</i>	<i>rpar, zg</i>	<i>dpa</i>	F3 acquisition status parameters (j)
<i>acqu2s</i>	<i>rpar, zg</i>	<i>dpa</i>	F2 acquisition status parameters (j)
<i>acqu3s</i>	<i>rpar, zg</i>	<i>dpa</i>	F1 acquisition status parameters (j)
Various lists as specified under 1D data files.			

## User defined settings

File	Created or modified by	Interpreted by	Description (file type)
<code>&lt;userhome&gt;/xwinnmr-&lt;hostname&gt;/</code>			
<i>resources</i>	<i>setres</i>	<i>zg, edcpul, edau, plot</i>	XWIN-NMR resource settings (j)
<i>default.por</i>	<i>search</i>	<i>search</i>	default portfolio (a)
<i>autoshim</i>	<i>gradshim</i>		gradient shimming directory (d)
<code>&lt;xwhome&gt;/prog/curdir/&lt;user&gt;</code>			
<i>history</i>	<i>hist on</i>		history of commands and error messages (a)
<i>curdat</i>	<i>edc, new</i>	<i>edc, new</i>	currently displayed dataset (j)
<i>individual_user_notebook.txt</i>	<i>Help → individual user...</i>	<i>Help → individual user...</i>	individual user notebook (a)

## AU Programs

Filename	Created or modified by	Interpreted by	Description (file type)
<xwhome>/prog/au/src.exam			
*		<b>expinstall</b>	Bruker AU program sources (a)
<xwhome>/prog/au/bin			
*	<b>compileall cpluser,edau</b>	<b>xau</b> <sup>a</sup>	User defined AU executables (e)
*	<b>compileall cplbruk,edau</b>	<b>xau</b>	Bruker AU executables (e)
<xwhome>/prog/include/			
<i>aucmd.h</i>		<b>edau</b>	AU macro definitions (a)
<xwhome>/prog/include/inc			
*		<b>edau</b>	AU macro and inclusion files (a)
<xwhome>/exp/stan/nmr/au			
<i>scr</i>	<b>expinstall</b>	<b>cplbruk, xau</b>	Bruker AU sources (d)
<i>scr</i>	<b>edau</b>	<b>cpluser, xau</b>	User defined AU sources (d)
<i>vorspann</i>		<b>edau</b>	C-languages definition file (a)
<i>makeau</i>		<b>edau</b>	AU compilation script (a)
<xwhome>/exp/stan/nmr/au/src			
*	<b>expinstall</b>	<b>compileall cplbruk,edau</b>	Bruker AU sources (a)
*	<b>edau</b>	<b>compileall cplbruk,edau</b>	User defined AU sources (a)

a. Note that AU programs can be started by entering **xau <name>** or simply **<name>**

### Documentation files

Filename	Interpreted by	Description (file type)
<i>&lt;xwhome&gt;/prog/docu/english</i>		
<i>xwinproc/pdf</i>	<b>Help</b>	manuals for acquisition, processing, installation, release letter etc. (d)
<i>avance/pdf</i>	<b>Help</b>	Avance spectrometer users guide (d)
<i>xwpman/pdf</i>	<b>Help</b>	XWIN-PLOT manuals (d)
<i>iconman/pdf</i>	<b>Help</b>	ICON-NMR manuals (d)

### NMR Suite executables

Filename	Interpreted by	Description (file type)
<i>&lt;xwhome&gt;/prog/cpr</i>		
<i>xcpu</i>	<b>XWIN-NMR</b>	XWIN-NMR graphics program (e)
<i>cpr</i>	<b>XWIN-NMR</b>	XWIN-NMR command interpreter (e)
<i>&lt;xwhome&gt;/prog/mod</i>		
<i>confed</i>	<b>cf</b>	configuration module (e)
<i>edparproc</i>	<b>eda, edp</b>	parameter editor module (e)
<i>expinstall</i>	<b>expinstall</b>	AU and parameter installation module (e)
<i>go</i>	<b>zg, go, wobb</b>	acquisition module (e)
<i>proc1d</i>	<b>em, ft, pk</b>	1D processing module (e)
<i>shimcntl</i>	<b>lock, rsh</b>	lock and shim module (e)
..		
see also Tcl/Tk scripts and AU programs		

### Tcl/Tk scripts

Filename	Interpreted by	Description (file type)
<i>&lt;xwhome&gt;/prog/tcl/xwish3_scripts</i>		
<i>butselnmr</i>	<b><i>butselnmr</i></b>	Easy interface for selective experiments (t)
<i>buttonnmr</i>	<b><i>buttonnmr</i></b>	Easy interface for standard experiments (t)
<i>config</i>	<b><i>config</i></b>	Spectrometer configuration (t)
<i>edhead</i>	<b><i>edhead</i></b>	Probehead selection (t)
<i>edprosol</i>	<b><i>edprosol</i></b>	Probehead/Solvent dependent parameter setup (t)
<i>glp</i>	<b><i>glp</i></b>	Program Good Laboratory Practice (t)
<i>gradshim</i>	<b><i>gradshim</i></b>	Gradient shimming (t)
<i>iconnmr</i>	<b><i>gradshim</i></b>	ICON-NMR interface (t)
<i>mascontrol</i>	<b><i>mascontrol</i></b>	MAS control and monitor (t)
<i>masrmon</i>	<b><i>masrmon</i></b>	MAS monitor (t)
<i>popt</i>	<b><i>popt</i></b>	Multiple parameter optimization (t)
<i>smaile</i>	<b><i>smaile</i></b>	Data transfer by Email (t)
<i>temon</i>	<b><i>temon</i></b>	Temperature monitor (t)

## Spectrometer configuration files

Filename	Created or modified by	Interpreted by	Description (file type)
<xwhome>/conf/instr/			
<i>curinst</i>	<i>cf</i>	<i>cf</i>	spectrometer name (a)
<i>patchlev</i>			XWIN-NMR patchlevel (a)
<i>hardware.exam</i>	<i>cf</i> <i>makelist</i>		example hardware list containing all possible hardware devices (a)
<i>nmr_user_notebook.txt</i>		<i>Help</i>	notebook for all NMR users (a)
<i>system_notebook.txt</i>		<i>Help</i>	notebook for system administrator (a)
<i>probehead</i>	<i>edhead</i>	<i>lock,edlock, getprosol edprosol</i>	current probe definition (a)
<i>probeheads</i>	<i>edhead</i>	<i>edhead</i>	probe parameter files (d)
<xwhome>/conf/instr/<instrum>			
<i>2Hlock</i>	<i>edlock</i>	<i>lock, lopo</i>	2H lock table (a)
<i>bacs_params</i>	<i>cfbacs, cf</i>		sample changer information (a)
<i>uxnmr.info</i>	<i>cf</i>		spectrometer configuration overview (a)
<i>cortab</i>	<i>cortab</i>	<i>zg</i>	amplifier power and phase correction tables (d)
<i>uxnmr.par</i>	<i>cf</i>		spectrometer configuration (j)
<i>nuclei</i>	<i>cf, ednuc</i>	<i>edasp</i>	nuclei table (a)
<i>scon</i>	<i>edscon</i>	<i>zg</i>	spectrometer fine tuning settings (j)
<i>specpar</i>	<i>edsp, edasp</i>	<i>zg</i>	routing information
<i>hardware_list</i>		<i>cf</i>	list of hardware components that are not automatically detected (a)
<i>mas_params</i>	<i>cfmas</i>	<i>masi, mase</i>	information about the MAS unit (a)



## Spectrometer configuration files

Filename	Created or modified by	Interpreted by	Description (file type)
<i>prosol</i>	<i>edprosol</i>	<i>getprosol</i>	probe/solvent dependent parameter files (d)
<i>autoshim</i>	<i>gradshim</i>	<i>gradshim</i>	gradient shimming files (d)
<i>bbis_bla1</i>	<i>cf</i>	<i>ii, zg</i>	linear amplifier information (a)
<i>bbis_bla2</i>	<i>cf</i>	<i>ii, zg</i>	linear amplifier information (a)
etc.			
<i>bbis_fcu1</i>	<i>cf</i>	<i>ii, zg</i>	fcu1 information (a)
<i>bbis_fcu2</i>	<i>cf</i>	<i>ii, zg</i>	fcu2 information (a)
etc.			
<i>bbis_rcu</i>	<i>cf</i>	<i>ii, zg</i>	rcu information (a)
<i>bbis_tcu</i>	<i>cf</i>	<i>ii, zg</i>	tcu information (a)
<i>bsmsdisp.calibr</i>	<i>bsmsdisp</i>		BSMS display calibration (a)
<xwhome>/conf/instr/<curinst>/rs232_device			
<i>acb</i>	<i>cf</i>	<i>acbdisp</i>	ACB rs232 channel + settings (a)
<i>bacs</i>	<i>cfbacs, cf</i>		sample changer rs232 channel (a)
<i>bsms</i>	<i>cf</i>	<i>lock, rsh</i>	BSMS rs232 channel + settings (a)
<i>lock</i>	<i>cf</i>	<i>lockdisp</i>	lock display rs232 channel (a)
<i>preamp1</i>	<i>cf</i>	<i>zg, ii, rga</i>	first HPPR rs232 channel (a)
<i>preamp2</i>	<i>cf</i>	<i>zg, ii, rga</i>	second HPPR rs232 channel (a)
..			
<i>preemp</i>	<i>cf</i>	<i>zg</i>	preemphasis channel + settings (a)
<i>rx22</i>	<i>cf</i>	<i>zg</i>	RX22 receiver channel + settings (a)
<i>temp</i>	<i>cfte, cf</i>	<i>edte, treset</i>	temperature unit rs232 channel (a)
<xwhome>/conf/instr/servtool <sup>a</sup>			
<i>acbttool</i>		NMR-CHECK	ACB service tool (d)

**Spectrometer configuration files**

<b>Filename</b>	<b>Created or modified by</b>	<b>Interpreted by</b>	<b>Description (file type)</b>
<i>bsmstool</i>		NMR-CHECK	BSMS service tool (d)
<i>gradtool</i>			Gradient service tool (d)
<i>hpcutool</i>			HPCU service tool (d)
<i>hpprool</i>		NMR-CHECK	HPPR preamplifier service tool (d)
<i>rx22tool</i>		NMR-CHECK	Rx22 receiver service tool (d)
<i>Unitool</i>			Generic service tool for AQS and HPPR II (d)

a. Service tools can be executed from the Windows Command prompt (or UNIX shell) or from NMR-CHECK.

## Format files

Filename	Created or modified by	Interpreted by	Description (file type)
<xwhome>/exp/stan/nmr/form/			
<i>acb.e</i>		<b><i>edacb</i></b>	acquisition control board parameters (a)
<i>acqu.e</i>	<b><i>cf</i></b>	<b><i>eda</i></b>	acquisition parameters (a)
<i>acqu.e.D</i>		<b><i>cf</i></b>	acquisition parameters for Avance (a)
<i>acqu.e.X</i>		<b><i>cf</i></b>	acquisition parameters for AMX with SCM (a)
<i>acqu.e.R</i>		<b><i>cf</i></b>	acquisition parameters for ARX with SCM (a)
<i>acqu.e.X.B</i>		<b><i>cf</i></b>	acquisition parameters for AMX with BSMS (a)
<i>acqu.e.R.B</i>		<b><i>cf</i></b>	acquisition parameters for ARX with BSMS (a)
<i>curd.e</i>		<b><i>edc</i></b>	current data path parameters (a)
<i>curd2.e</i>		<b><i>edc2</i></b>	second data path parameters (a)
<i>dosy.e</i>		<b><i>eddosy</i></b>	dosy parameter (a)
<i>edp3.e</i>		<b><i>edp3</i></b>	F3 processing parameters in 3D (a)
<i>edp2.e</i>		<b><i>edp2</i></b>	F2 processing parameters in 2D/3D (a)
<i>edp1.e</i>		<b><i>edp1</i></b>	F1 processing parameters in 2D/3D (a)
<i>hpcu.e</i>	<b><i>cf</i></b>	<b><i>edhpcu</i></b>	acquisition + HPCU parameters (a)
<i>hpcu.e.D</i>		<b><i>cf</i></b>	acqu. + HPCU parameters for Avance (a)
<i>hpcu.e.X</i>		<b><i>cf</i></b>	acqu. + HPCU parameters for AMX - SCM (a)
<i>hpcu.e.X.B</i>		<b><i>cf</i></b>	acqu. + HPCU parameters for AMX - BSMS (a)
<i>outd.e</i>		<b><i>edo</i></b>	output parameters (a)
<i>proc.e</i>		<b><i>edp</i></b>	processing parameters in 1D/2D/3D (a)
<i>preemp.e</i>		<b><i>edpre</i></b>	preemphasis parameters (a)

## Format files

Filename	Created or modified by	Interpreted by	Description (file type)
<i>spec.e</i>		<i>edsp, edasp</i>	routing format file on AMX/ARX
<i>scon.e</i>	<i>cf</i>	<i>edscon</i>	spectrometer constants for your spectrometer (a)
<i>scon.e.A</i>		<i>cf</i>	spectrometer constants for Avance-AQS (a)
<i>scon.e.D</i>		<i>cf</i>	spectrometer constants for Avance-AQS (a)
<i>used_from.e</i>		<i>edc2 used_from</i>	source 2D data path parameters (a)
<i>maxent1d.e</i>		<i>maxed</i>	1D MAXENT parameters (a)
<i>maxent2d.e</i>		<i>maxed</i>	2D MAXENT parameters (a)
<i>maxent3d.e</i>		<i>maxed</i>	3D MAXENT parameters (a)
<i>t1par</i>		<i>t1/t2</i>	T1/T2 relaxation format files (d)
<i>plot</i>		<i>edx</i>	1D plot parameter format files (d)
<i>plot2</i>		<i>edx</i>	2D plot parameter format files (d)
<i>plotw</i>		<i>edgw</i>	stack plot parameter format files (d)
<i>plotx</i>		<i>edgx</i>	extended plot parameter format files (d)
<i>acqu.1</i>		<i>dpa, lpa, plot</i>	acqu. status format files for the first dimension of 1D, 2D or 3D data (d)
<i>acqu2.1</i>		<i>dpa, lpa, plot</i>	acqu. status format files for the second dimension of 2D or 3D data (d)
<i>acqu3.1</i>		<i>dpa, lpa, plot</i>	acqu. status format files for the third dimension of 3D data (d)
<i>curd.1</i>		<i>dpc, lpc, plot</i>	current dataset status format files (d)
<i>outd.1</i>		<i>dpo, lpo, plot</i>	output status parameters (d)
<i>plot.1</i>		<i>dpo, lpo, plot</i>	1D plot status format files (d)

### Format files

Filename	Created or modified by	Interpreted by	Description (file type)
<i>plot2.l</i>		<i>dpo, lpo, plot</i>	2D plot status format files (d)
<i>proc.l</i>		<i>dpp, lpp, plot</i>	processing status format files for the first dimension of 1D, 2D or 3D data (d)
<i>proc2.l</i>		<i>dpp, lpp, plot</i>	processing status format files for the second dimension of 2D or 3D data (d)
<i>proc3.l</i>		<i>dpp, lpp, plot</i>	processing status format files for the third dimension 3D data (d)
<i>&lt;xwhome&gt;/exp/stan/nmr/form/acqu.l</i>			
<i>formased</i>		<i>ased</i>	acquisition parameters used in pulse program (a)
<i>normdp</i>		<i>dpa</i>	acquisition status parameters to be viewed (a)
<i>normlp</i>		<i>lpa</i>	acquisition status parameters to be printed (a)
<i>normpl</i>		<i>plot, view</i>	acquisition status parameters to be plotted (a)
<i>normdp.D</i>		<i>cf</i>	acquisition status parameters for Avance (a)
<i>normdp.R</i>		<i>cf</i>	acqu. status parameters for ARX with SCM (a)
<i>normdp.R.B</i>		<i>cf</i>	acqu. status parameters for ARX with BSMS (a)
<i>normdp.X</i>		<i>cf</i>	acqu. status parameters for AMX with SCM (a)
<i>normdp.X.B</i>		<i>cf</i>	acqu. status parameters for AMX with BSMS (a)
<i>hpcudp</i>		<i>dpa</i>	as normdp + HPCU parameters (a)
<i>hpculp</i>		<i>lpa</i>	as normlp + HPCU parameters (a)
<i>hpcupl</i>		<i>plot, view</i>	as normpl + HPCU parameters (a)



---

# Chapter 15

## Bruker addresses

---

### **Germany**

Bruker BioSpin GmbH  
Silberstreifen  
D-76287 Rheinstetten  
Tel: (++49) (721) 51 61 0  
Fax: (++49) (721) 51 71 01

Bruker BioSpin Software Department  
Silberstreifen  
D-76287 Rheinstetten  
Tel: (++49) (721) 5161 440  
Fax: (++49) (721) 5161 480

<http://www.bruker-biospin.de>

ftp server: <ftp.bruker.de>

Email: [ut@bruker.de](mailto:ut@bruker.de) (sales)  
[mbu@bruker.de](mailto:mbu@bruker.de) (service)  
[applik@bruker.de](mailto:applik@bruker.de) (application)  
[nmr-software-support@bruker.de](mailto:nmr-software-support@bruker.de) (software)  
[license@bruker.de](mailto:license@bruker.de) (licenses)

**USA**

Bruker BioSpin Corporation  
15 Fortune Drive  
Manning Road  
Billerica, MA. 01821-3991

Tel: (++1)(978) 667 9580 195 (center)  
      (++1)(978) 667 9580 444 (application)  
Fax: (++1)(978) 667 6168 (center)  
      (++1)(978) 667 2955 (application)

<http://www.bruker-biospin.com>

ftp server: <ftp.bruker.com>

Email: [applab@bruker.com](mailto:applab@bruker.com)  
      [center@bruker.com](mailto:center@bruker.com)  
      [software@bruker.com](mailto:software@bruker.com)

**Switzerland**

Bruker BioSpin AG  
Industriestraße 26  
CH-8117 Fällanden

Tel: (++41)(1) 8 25 91 11  
Fax: (++41)(1) 8 25 96 96

web server: [www.bruker.ch](http://www.bruker.ch)

E-Mail: [epweb@bruker.ch](mailto:epweb@bruker.ch)

[all\\_ap@bruker.ch](mailto:all_ap@bruker.ch)

**France**

Bruker BioSpin S.A.  
34, rue de l'industrie  
F-67166 Wissembourg/Cedex

Tel: (++33)(3) 88 73 68 00  
Fax: (++33)(3) 88 73 68 79

E-Mail: [support-rmn@bruker.fr](mailto:support-rmn@bruker.fr) (customer support)

**England**

Bruker BioSpin LTD.  
Banner lane



Coventry CV4 9GH  
Tel: (+44) (2476) 855200  
Fax: (+44) (2476) 465317  
Email: [service@bruker.co.uk](mailto:service@bruker.co.uk)  
[apps@bruker.co.uk](mailto:apps@bruker.co.uk)

Our webpage

<http://www.bruker-biospin.de/analytic/nmr-dep/about/offices/contact.htm>

provides contact addresses of our facilities and offices worldwide



---

# Command - Index

A-xx: Acquisition Reference Manual

P-xx: Processing Reference Manual

---

A B C D E F G H I J K L M N O P Q R S T U V W X Z

## A

abs P-48  
abs1 P-142  
abs2 P-144  
absd P-48  
absd1 P-142  
absd2 P-144  
absf P-48  
absot1 P-146  
absot2 P-148  
abst1 P-146  
abst2 P-148  
acqu A-192  
add P-51  
add2d P-150, P-152  
addc P-53  
addfid P-51  
and P-54  
apk P-56  
apk0 P-56  
apk1 P-56  
apkf P-56  
apks P-56  
as A-150  
ased A-150  
atma A-134  
atmm A-136  
auditcheck P-512  
autoplot P-306  
autoshim A-120

## B

bc P-59  
bcm P-62  
bcm1 P-154  
bcm2 P-154  
browse P-392  
butselnmr A-193  
buttonnmr A-196

## C

cf A-62  
cfbacs A-66  
cfbpsu A-67  
cfmas A-240  
cfte A-68  
compileall A-69, P-426  
config A-70  
conv P-490  
convdta P-496  
convsys P-495  
cortab A-72  
cplbruk A-81, P-427  
cpluser A-81, P-427  
crplock A-276  
crpobs A-276  
crpoff A-276  
crpon A-276  
ct1 P-359  
ct2 P-362

## D

dat1 P-365  
del P-393

## Command Index

---

del2d P-397

dela P-393

delau A-260, P-429

deldat P-393

delf P-397

delgp A-260, P-429

deli P-397

dellist A-262, P-431

delmac A-260, P-429

delmisc P-433

delp P-393

delpar A-260, P-429

delpul A-260, P-429

dels P-397

delser P-397

delsh A-260, P-429

dir P-400

dir2d P-403

dira P-400

dirdat P-400

dirf P-403

diro P-405

dirp P-400

dirpar P-434

dirs P-403

dirser P-403

div P-64

dosy2d P-156

dosy3d P-260

dp P-435

dpa A-152, P-436

dpc P-438

dpg P-439

dpgx P-440

dpo P-441

dpp P-442

dt P-66

duadd P-67

### **E**

ed4ph A-243

eda A-154

edacb A-274

edasp A-156

edau A-172, P-444

edc P-407

edc2 P-409

edcgp A-174, P-446

edcpd A-176, P-448

edcpul A-177, P-449

eddosy P-451

edg P-308

edgp A-179, P-453

edgw P-308

edgx P-308

edhead A-138

edhpcu A-242

edinfo P-310

edlev P-312

edlist A-180, P-454

edlock A-104

edmac P-456

edmisc P-458

ednuc A-83

edo P-410

edp P-460

edp1 P-463

edp2 P-463

edp3 P-463

edprosol A-85

edpul A-188, P-464

edscon A-89  
edsolv A-93  
edsp A-95  
edt1 P-367  
edte A-228  
edtg A-229  
edtune A-121  
ef P-69  
efp P-69  
ej A-142  
elim P-368  
em P-71  
expinstall A-97, P-466  
expt A-199  
**F**  
f1disco P-157  
f1projn P-159  
f1projp P-159  
f1sum P-162  
f2disco P-157  
f2projn P-159  
f2projp P-159  
f2sum P-162  
filt P-74  
flplot P-314  
fmc P-73  
follow P-514  
fp P-76  
fromjdx P-498  
ft P-77  
**G**  
gdcheck P-515  
gdcon P-82  
genfid P-84  
genser P-164

gethpcu A-257  
getlcosy A-277  
getlim1d A-277  
getlinv A-277  
getljres A-277  
getlxhco A-277  
getprosol A-160  
gf P-86  
gfp P-86  
gm P-88  
go A-200  
gs A-202  
**H**  
halt A-204  
hist P-517  
hoff P-518  
hon P-518  
ht P-91  
**I**  
iconnmr A-206  
ift P-93  
ii A-207  
ij A-142  
int2d P-315  
int2dref P-315  
**J**  
jconv P-504  
**K**  
kill P-519  
**L**  
ldcon P-94  
levcalc P-317  
lfilter A-108  
lgain A-108  
li P-319

limits2d P-321  
lipp P-322  
lippf P-322  
lock A-110  
lockdisp A-114  
lopo A-116  
lopoi A-116  
lp P-324  
lpa P-325  
lpc P-326  
lpg P-327  
lpgx P-328  
lpo P-329  
lpp P-330  
lppi P-331  
ls P-96  
lsta P-372  
lstp P-370  
ltime A-108  
**M**  
mas A-245  
mascontrol A-249  
mase A-252  
masg A-252  
mash A-252  
masi A-252  
masr A-252  
masrmon A-255  
mc P-98  
mdcon P-99  
mul P-101  
mulc P-103  
**N**  
nm P-104  
nxtp P-374

**O**  
or P-105  
**P**  
paste P-412  
pd P-376  
pd0 P-376  
pft2 P-379  
pk P-107  
plot P-332  
plotreg P-335  
plots P-332  
plotw P-332  
plotx P-332  
popt A-208  
pp P-336  
pp2d P-341  
pp2dmi P-341  
pph P-336  
ppj P-336  
ppp P-336  
pps P-336  
ppt1 P-381  
proj P-166  
ps P-109  
ptilt P-168  
ptilt1 P-170  
pulsdisp A-268  
**Q**  
qsin P-110  
qsinc P-112  
**R**  
r12 P-293  
r12d P-299  
r12p P-302  
r13 P-295

r13d P-300  
r13p P-303  
r23 P-297  
r23d P-301  
r23p P-304  
rackpow A-257  
re P-414  
ren P-416  
renau A-264, P-471  
rengp A-265, P-472  
renlist A-266, P-475  
renlut A-265, P-472  
renmac A-265, P-472  
reno P-416  
renpar A-264, P-471  
renpul A-265, P-472  
rep P-414  
resume A-212  
rev1 P-172  
rev2 P-172  
rga A-213  
rhnp P-173  
rhpp P-173  
rlut P-343  
rmisc P-473  
rmplot P-344  
ro A-143  
rpar A-163, P-476  
rs P-114  
rsc P-176  
rser P-181  
rser2d P-184  
rsh A-122  
rspc P-383  
rsr P-186

rstp P-384  
rv P-116  
rvnp P-173  
rvpp P-173  
**S**  
sab P-117  
search P-417, P-419  
setdef P-520  
sethpcu A-257  
setres P-521  
setsh A-124  
setti P-345  
show P-522  
simfit P-385  
sinc P-119  
sinm P-121  
sino P-346  
spdisp A-272  
sref P-349  
status P-523  
stop A-214  
sub1 P-189  
sub1d1 P-189  
sub1d2 P-191  
sub2 P-191  
suspend A-215  
sym P-193  
syma P-195  
symj P-197  
**T**  
tabs1 P-265  
tabs2 P-263  
tabs3 P-261  
te2get A-230  
te2ready A-234

## Command Index

---

te2set A-236  
teget A-230  
temon A-231  
tepar A-233  
teready A-234  
teset A-236  
tf1 P-279  
tf1p P-288  
tf2 P-274  
tf2p P-286  
tf3 P-267  
tf3p P-284  
tht1 P-292  
tht2 P-291  
tht3 P-290  
tilt P-199  
tm P-123  
tojdx P-501  
tojdx5 P-501  
tr A-216  
traf P-131  
trafs P-131  
trf P-126  
trfp P-126  
tune A-125  
**U**  
uwm P-132  
**V**  
vconv P-507  
view P-352  
viewmg P-352  
vieww P-352  
viewx P-352  
vish A-130

**W**  
wmisc P-480  
wobb A-145  
wpar A-167, P-482  
wra P-422  
wrđ P-422  
wrp P-422  
wrpa P-422  
wsc P-201  
wser P-204  
wserp P-207  
wsh A-131  
wsr P-210  
**X**  
xau A-217, P-484  
xaua A-217, P-484  
xaup A-217, P-484  
xf1 P-213  
xf1m P-216  
xf1p P-218  
xf1ps P-220  
xf2 P-222  
xf2m P-216  
xf2p P-226  
xf2ps P-220  
xfb P-228  
xfbm P-216  
xfbP P-242  
xfbps P-220  
xhelp P-524  
xht1 P-244  
xht2 P-245  
xif1 P-246  
xif2 P-246  
xmac P-487



xor P-134  
xtrf P-248  
xtrf2 P-248  
xtrfp P-252  
xtrfp1 P-252  
xtrfp2 P-252  
xwinplot P-353  
xwp\_lp P-354  
xwp\_pp P-356

**Z**

zert1 P-255  
zert2 P-257  
zf P-136  
zg A-220  
zp P-138



---

# Index

---

## Numerics

2D lookup tables A-265

4-phase modulator A-90, A-243

## A

abs command A-36, A-53, A-197

ACB parameters A-274

ACB service tool A-287

accumulation of the FID A-192

acqu command A-221

acquisition

dimension A-44, A-46

interface A-193, A-196

menu A-192, A-196, A-202

mode A-12, A-44, A-52

order A-53

parameters A-7, A-8, A-9, A-10, A-11, A-154,  
A-163, A-164, A-165, A-167, A-192,  
A-202, A-204, A-208, A-280, A-281,  
A-282, A-289

status parameters A-7, A-9, A-52, A-152, A-  
280, A-281, A-282, A-291

time A-11, A-45, A-150

air flow A-142, A-247, A-249, A-252, A-253

amplifier

correction table A-72

linearization A-72

routing A-32, A-34, A-156, A-158

amplitude list A-181, A-183

Amplitude Setting Unit A-72

AMX spectrometer A-17, A-25, A-98, A-100, A-111,  
A-181

analog filter A-17, A-23

apk command A-53, A-197, A-223

ARX spectrometer A-98, A-101, A-111

as command A-150

ased command A-150, A-154, A-220

ASU

blanking A-89

correction A-72

linearization A-74

ATM probehead A-134, A-136, A-148, A-220

atma command A-51, A-134, A-136, A-220

atmm command A-136

attenuator A-72

AU program

binaries A-69, A-81, A-97, A-173, A-283

Bruker defined A-69, A-81, A-97, A-172

compilation A-69, A-81, A-172, A-217

creation A-172

deletion A-260

edit A-172

execution A-217

files A-283

for acquisition A-12, A-192, A-218

for processing A-218

installation A-97

kill A-218

macro A-4, A-5

rename A-264

sources A-69, A-81, A-173, A-283

user defined A-81, A-97, A-172

au\_zg AU program A-13, A-192

auto lock-in A-110, A-111, A-193, A-196

autolock key A-54, A-108, A-112, A-116

automatic

receiver gain optimization A-213

shimming A-121, A-125, A-128, A-193, A-196

tuning and matching A-134

automation with Icon-nmr A-5, A-43, A-66, A-87, A-  
106, A-140, A-206, A-218, A-277

autoshim A-120

Avance spectrometer A-5, A-31, A-98, A-100, A-  
111, A-181, A-284

## B

basic frequency A-13, A-32, A-34, A-57, A-59, A-83,  
A-104, A-147, A-156

BPSU unit A-67, A-70

Bruker

AU programs A-69, A-81, A-97, A-172, A-200,  
A-264, A-283

CPD programs A-15, A-176

gradient files A-24

gradient programs A-179

parameter sets A-14, A-34, A-98, A-104, A-158,  
A-164, A-168, A-218, A-260

pulse programs A-15, A-23, A-30, A-36, A-40,  
A-177, A-188, A-215, A-265

shape files A-42

solvent list A-93

spectrometers A-17, A-227

Bruker addresses A-293

BSMS

display A-104, A-122, A-143, A-287

keyboard A-8, A-54, A-58, A-104, A-106, A-  
108, A-111, A-112, A-114, A-116, A-  
119, A-122, A-128, A-142, A-143, A-  
146

lock unit A-111

rs232 channel A-65

service tool A-288

unit A-54, A-103, A-105, A-111, A-112, A-126,  
A-143

bsmsdisp command A-104, A-122, A-143

butselau AU program A-195

butselnmr A-193

buttonau AU program A-198

buttonnmr command A-39, A-196, A-223

## C

calibration A-59, A-105

carrier frequency A-34, A-35, A-41, A-99, A-100

cf command A-13, A-32, A-62, A-97

cfbacs command A-66

cfbpsu command A-67

cfmas command A-240, A-249, A-253

cfte command A-67, A-68

chemical shift A-35, A-59, A-105, A-111

communication problems A-62

compilation of AU programs A-81, A-97, A-172, A-  
217

compileall command A-69, A-97, A-217

composite experiments A-277

composite pulse decoupling A-14, A-176

config command A-32, A-62, A-70

configuration

commands A-61

files A-286

of the BPSU unit A-67

of the MAS unit A-240

of the sample changer A-66

of the spectrometer A-62

of the temperature unit A-68

of XWIN-NMR A-222

suite A-62, A-70

contour levels A-163, A-164, A-165, A-166, A-168,  
A-169

conventions in this manual A-3

Cortab

experiment A-72, A-76

linearization A-72

program A-72

verification A-72

cortab command A-72

counter list A-181

CPD

command A-37

power level A-161

programs A-14, A-38, A-97, A-100, A-176, A-  
187

pulse length A-37, A-151, A-161

cplbruk command A-81, A-97, A-217, A-260

cpluser command A-81, A-217

crplock command A-276

crpobs command A-276

crpoff command A-276

crpon command A-276

cryo preamplifier A-276

## D

data path parameters A-289

dataset list A-19, A-181

decimation factor A-16, A-21, A-53

define statement A-50, A-98

delau command A-260

delay list A-181

delete

AU programs A-260  
gradient programs A-260  
macros A-260  
parameter sets A-260  
pulse programs A-260  
shim files A-260  
various lists A-262  
delgp command A-260  
dellist command A-262  
delmac command A-260  
delpar command A-260  
delpul command A-260  
delsh command A-260  
demand  
  spin rate A-143, A-240, A-246, A-249, A-252  
  temperature A-46, A-230, A-234, A-236, A-245  
deuterium A-104  
digital  
  digitizer resolution A-15, A-18  
  filtering A-5, A-15, A-16, A-17, A-20, A-21, A-53  
  lock A-103  
  quadrature detection A-12, A-18  
digitizer  
  mode A-17  
  resolution A-15, A-18  
  start A-16  
  type A-11, A-17, A-18, A-23, A-100  
dimension  
  2D direct A-4, A-9, A-32  
  2D indirect A-4, A-9, A-31, A-44, A-46  
  3D direct A-9  
  3D indirect A-5, A-9  
dimensionality of data A-36, A-164  
DNA/RNA experiments A-160  
dpa command A-7, A-8, A-52, A-152  
DQD  
  device A-12  
  mode A-18  
DSP firmware A-54  
dummy scans A-18, A-150, A-220, A-221, A-269  
duty cycle A-274  
dwell time A-17, A-20, A-43, A-150

**E**

ed4ph command A-243  
eda command A-7, A-8, A-10, A-11, A-161, A-164, A-168  
edacb command A-274  
edasp command A-13, A-21, A-25, A-31, A-32, A-34, A-38, A-57, A-220  
edau command A-5, A-38, A-97, A-172, A-217  
edc command A-168, A-193, A-196, A-220  
edcgp command A-174  
edcpd command A-15, A-176, A-187  
edcpul command A-39, A-177, A-179  
edg command A-164, A-168  
edgp command A-179, A-187  
edgx command A-164  
edhead command A-40, A-56, A-85, A-104, A-138, A-220  
edhpcu command A-242  
edlev command A-164  
edlist command A-19, A-23, A-47, A-48, A-49, A-50, A-180  
edlock command A-8, A-29, A-87, A-104, A-108, A-110, A-116  
edmac command A-5, A-187  
ednuc command A-13, A-63, A-83  
edo command A-105, A-164, A-168  
edp command A-154, A-164, A-168  
edprosol command A-8, A-39, A-43, A-85, A-125, A-161, A-220  
edpul command A-36, A-187, A-188  
edscon command A-16, A-89, A-224  
edsolv command A-93  
edsp command A-95, A-156  
edte command A-228, A-233  
edtq command A-229  
edtune command A-121, A-125  
ef command A-197  
efp command A-194  
ej command A-142  
eject  
  air time A-240, A-247  
  mechanism A-240  
  the sample A-247, A-249, A-253  
  the spinner A-252  
em command A-53, A-194, A-223

eurotherm temperature unit A-227  
experiment time A-31, A-197, A-199  
exinstall command A-17, A-40, A-69, A-81, A-96,  
A-97, A-98, A-164, A-172, A-188  
expt command A-31, A-192, A-199

## F

### FID

area A-127  
display A-12, A-34, A-41, A-202, A-203  
envelope A-125  
number A-36  
resolution A-21, A-46  
signal A-40

field value A-104, A-105, A-114

follow command A-218

font conventions A-3

format files A-289

frequency

list A-23, A-181, A-182, A-225, A-280  
unit A-268

ft command A-36, A-53, A-194, A-223

## G

gethpcu command A-257

getlcosy command A-277

getlim1d command A-277

getlinv command A-277

getljres command A-277

getprosol command A-8, A-39, A-43, A-87, A-160,  
A-193, A-197, A-220

getxhco command A-277

go command A-11, A-200, A-222

gpro command A-160

gradient

experiment A-198  
file A-24, A-25, A-97, A-98  
parameter table A-23  
parameters A-24, A-25  
program A-25, A-100, A-174, A-179, A-187, A-  
260, A-265  
pulse length A-161  
recovery delay A-161  
service tool A-288  
shaped A-24

shimming A-25, A-220, A-282, A-287

static A-24

strength A-24

gradshim command A-220

group delay A-5, A-53, A-54

gs command A-11, A-12, A-34, A-41, A-192, A-202,  
A-214

## H

halt command A-31, A-135, A-200, A-204, A-212,  
A-214, A-215, A-222

hard pulse A-86, A-87, A-161

hardware list A-64, A-66, A-240, A-286

hardware resolution A-15, A-18

high power

accessory A-38

amplifier A-25, A-33, A-157

cabinet A-257

output A-33, A-157

preamplifier A-25

routing A-58, A-100, A-159

tube transmitter A-242, A-257

holder position A-56

homodecoupling

duty cycle A-25

experiment A-17

power level A-161

homogeneity of the magnetic field A-114, A-125

hostname A-62

hosts file A-62, A-64

HPCU

parameters A-242

service tool A-288

unit A-257

HPPR preamplifier A-25, A-147, A-276

service tool A-288

## I

iconnmr command A-206

ii command A-11, A-207

ij command A-142

impedance

of a probehead A-145, A-146, A-147

input parameters A-8

insert

airtime A-240, A-246, A-247, A-249, A-252  
the sample A-142, A-246, A-247  
the spinner A-247, A-249, A-252  
integral ranges A-187, A-277  
interface  
  of the spectrometer A-207  
irradiation frequency A-23, A-32, A-34, A-35, A-41,  
  A-57, A-59, A-99, A-100, A-111, A-151,  
  A-156, A-182, A-202  
irradiation frequency offset A-35, A-57, A-59, A-105,  
  A-182, A-202

## K

kill command A-135, A-204, A-214, A-218, A-222

## L

LC-NMR A-67, A-70, A-160  
lfilter command A-106, A-108  
lgain command A-55, A-106, A-108  
liprosol command A-87  
lock  
  channel A-276  
  display A-106, A-114  
  display file A-115  
  frequency A-104, A-114  
  gain A-110  
  irradiation frequency A-59  
  level A-126, A-127  
  nucleus A-28, A-104, A-114  
  parameters A-7, A-29, A-58, A-70, A-104, A-  
    110, A-116  
  phase A-59, A-105, A-106, A-110, A-114, A-  
    116, A-128  
  power A-55, A-59, A-104, A-105, A-106, A-  
    110, A-116  
  regulator cut-off frequency A-55, A-59, A-104  
  regulator filter A-108  
  regulator gain A-54, A-59, A-104, A-108  
  regulator time A-55, A-59, A-104, A-108  
  shift A-55  
  signal A-105, A-106, A-110, A-112, A-114, A-  
    116  
  status A-56  
  sweep A-106  
  table A-104

lock command A-8, A-29, A-43, A-58, A-105, A-  
  108, A-110, A-116, A-193, A-196, A-220  
lockdisp command A-106, A-112, A-114, A-220  
lock-in procedure A-58, A-110, A-111, A-116  
logical frequency channel A-21, A-32, A-58, A-86,  
  A-100, A-156, A-159  
loop counter A-27  
loop filter A-55, A-59, A-104, A-106, A-108, A-111,  
  A-116  
loop gain A-54, A-59, A-106, A-108, A-111, A-116  
loop time A-55, A-59, A-104, A-106, A-108, A-111,  
  A-116  
loopadj AU program A-106, A-108  
lopo command A-8, A-29, A-43, A-58, A-105, A-  
  112, A-116  
lopoi command A-29, A-116  
ltime command A-106, A-108

## M

macros  
  in AU programs A-5, A-283  
  in XWIN-NMR A-5, A-187, A-260, A-262, A-  
    265  
magic angle A-239, A-249, A-252  
magnet  
  air flow A-142  
  strength A-62  
  type A-66, A-245  
MAS  
  commands A-253  
  control A-29, A-245, A-249, A-252  
  log file A-250  
  manual A-253  
  monitor A-55, A-255  
  parameters A-240, A-245  
  pneumatic unit A-70, A-240, A-245, A-246, A-  
    247, A-249, A-252, A-286  
  spin rate A-29, A-55, A-249, A-253  
  spin rate list A-181, A-187  
mas command A-29, A-245, A-249, A-252  
mascontrol command A-252  
mase command A-252  
masg command A-252  
mash command A-252  
masi command A-252

masr command A-29, A-55, A-252  
masrmon command A-55, A-249, A-255  
matching knob A-146, A-147  
matching the probehead A-51, A-134, A-135, A-136,  
A-145, A-192, A-193, A-196  
mc command A-46  
micro imaging A-101  
mixed solvents A-111  
mixing time A-86  
multiefp AU program A-194  
multimas AU program A-186  
multizg AU program A-223

## N

new dataset A-168, A-193, A-196, A-220  
NMR Superuser A-13, A-62, A-97, A-153, A-155, A-  
172  
nuclei  
list A-34, A-83  
selection A-156  
table A-13, A-65, A-83, A-286

## O

O1 spikes A-12  
observe nucleus A-59, A-105  
optimization  
criterion A-208  
of acquisition parameters A-202, A-208  
of the lock parameters A-110  
of the receiver gain A-213  
protocol of popt A-208  
output device parameters A-163, A-164, A-165, A-  
168  
output parameters A-7, A-8  
overflow A-18, A-35, A-213  
oversampling A-16, A-17, A-18, A-20, A-23, A-53  
dwell time A-17, A-20

## P

parameter  
editor A-8, A-10  
optimization A-202, A-208  
sets A-97, A-98, A-163, A-167, A-168, A-218,  
A-260, A-264  
types A-163, A-164, A-167, A-168, A-260

value types A-10  
paropt AU program A-210  
paroptlin AU program A-210  
paroptlog AU program A-210  
parray AU program A-210  
phase  
alignment A-42, A-43, A-86  
correction factor A-37  
correction of a pulse A-37  
correction of the receiver A-37  
correction of the spectrum A-16, A-92  
cycle A-19, A-37, A-200, A-204, A-222  
list A-202  
of a pulse A-268  
of the lock signal A-59, A-105, A-106, A-111,  
A-114, A-116, A-128  
of the receiver A-16, A-91, A-268  
program A-36  
values of a shaped pulse A-185  
phtran AU program A-37  
pk command A-194  
plot  
parameters A-7, A-163, A-164, A-165, A-168  
title A-163, A-164, A-165, A-168  
plot command A-194, A-197  
popt command A-208  
portfolio A-282  
power  
level A-8, A-38, A-42, A-43, A-86, A-87, A-  
151, A-161, A-202, A-268, A-269  
list A-183  
mode A-38  
preamplifier  
blanking A-89  
gain A-39  
modules A-25, A-33, A-38, A-58, A-96, A-100,  
A-157, A-158, A-159  
of a cryoprobe A-276  
of an MSL A-39  
routing A-32, A-33, A-58, A-96, A-99, A-156,  
A-157, A-158, A-159  
switching A-16, A-91  
tuning and matching A-146  
preferred output for 19F A-33, A-58, A-96, A-99, A-  
157, A-158, A-159



preferred preamplifier A-33, A-58, A-96, A-99, A-157, A-158, A-159

pre-scan

delay A-15, A-91, A-150

subdelays A-16, A-91

probehead A-40, A-104

acquisition parameter A-56

ATM A-134, A-136

broadband A-136, A-147

commands A-133

cryo A-276

current A-104, A-138, A-139, A-220

dependent lock parameters A-87

dependent parameters A-8, A-43, A-59, A-85, A-160, A-197, A-220

dependent tunefile A-87

impedance A-145

name A-138

parameter editor A-138

parameters A-138, A-139

resonance circuit A-145

resonance frequency A-136, A-145, A-147

table A-70, A-138

tunefile A-193

tuning and matching A-51, A-134, A-136, A-145, A-193, A-196, A-220

type A-245, A-249, A-252

processing

commands A-3, A-7, A-36, A-46, A-52, A-53, A-54, A-223

menu A-192

parameters A-7, A-163, A-164, A-165, A-168

prosol command A-87

prosol parameters A-7, A-8, A-39, A-85, A-160

protein experiments A-160

pulsdisp command A-11, A-268

pulse channels A-268

pulse length

CPD A-37, A-151, A-161

decrement A-26

display A-271

increment A-26

of a 180 degree pulse A-161

of a 90 degree pulse A-86, A-145, A-161

of a gradient pulse A-161

of a hard pulse A-86

of a soft pulse A-86

parameter A-35

reset A-26

pulse list A-181

pulse program

Bruker defined A-177, A-188

compilation A-220

constants A-13

creation A-177, A-188

current A-177

define statements A-98

deletion A-260

display A-268

edit A-177, A-188

for AMX A-100

for wobble A-135

installation A-97

label A-27, A-30, A-49, A-183

rename A-265

timing A-268

user defined A-177, A-188

pulsesort AU program A-160

## Q

QNP probe A-40

qnpset AU program A-40

quad images A-12

quadrature detection A-12

## R

rackpow command A-257

raw data A-5, A-7, A-9, A-30, A-36, A-45, A-152, A-164, A-192, A-197, A-200, A-204, A-222, A-224, A-280, A-281

receiver

channel A-40

control A-268

gain A-40, A-150, A-202, A-203

gain optimization A-213

gate A-16, A-91

output A-41

phase A-16, A-37, A-91, A-268

rs232 channel A-65, A-287

service tool A-288

- status A-268
  - recycle delay A-19, A-212, A-221
  - reference
    - 50 ohm resistor A-145
    - frequency A-40, A-100
    - number A-93, A-138
    - peak A-98
    - phase A-202
    - shift A-59, A-105
    - signal A-59, A-105
    - substance A-13, A-111
  - referencing the spectrum A-104
  - relations
    - file A-87, A-160, A-280
    - prosol-acquisition A-87, A-160, A-280
  - relaxation delay A-151
  - rename
    - 2D lookup tables A-265
    - AU programs A-264
    - gradient programs A-265
    - lists A-266
    - macros A-265
    - parameter sets A-264
    - pulse programs A-265
  - renau command A-264
  - rengp command A-265
  - renlist command A-266
  - renlut command A-265
  - renmac command A-265
  - renpar command A-264
  - renpul command A-265
  - resonance circuit A-145, A-147
  - resonance frequency
    - of a probehead A-136, A-145, A-146, A-147
  - resume command A-11, A-200, A-212
  - rga command A-11, A-213
  - ro command A-41, A-66, A-143
  - ROESY experiment A-86, A-161
  - rotation of the sample A-40, A-66, A-126, A-134, A-143, A-144, A-146, A-249
  - rotor material A-245
  - routing
    - parameters A-34, A-96, A-158
    - table A-13, A-32, A-57, A-156, A-157
  - rpar command A-17, A-34, A-36, A-40, A-163, A-167, A-168, A-193, A-196, A-220
  - rsh command A-122, A-128, A-220
  - RX22
    - rs232 channel A-65, A-287
    - service tool A-288
  - RX22 receiver A-16, A-91
- ## S
- sample
    - changer A-66, A-70
    - depth A-139
    - diameter A-138, A-139, A-240, A-245
    - eject A-142, A-247
    - insert A-142, A-246, A-247
    - position A-142
    - rotation A-66, A-126, A-134, A-143, A-146, A-249
    - rotation frequency A-40, A-144
    - solvent A-42, A-59
    - temperature A-19, A-221, A-230, A-234, A-236
    - transfer tube A-247, A-249
  - saturation of the lock signal A-106
  - scaling region file A-98, A-187
  - SCM unit A-111
  - SE451 receiver A-41
  - sensitivity of GS sliders A-203
  - sequential mode A-12, A-18
  - sethpcu command A-257
  - setres command A-173, A-223
  - setsh command A-124
  - setti command A-164
  - SGU verification A-74
  - shape files A-42, A-86, A-98, A-101
  - Shape Tool A-25, A-42
  - shaped gradient
    - file name A-225
  - shaped pulse
    - file name A-225, A-280
    - frequency offset A-99
    - in CPD programs A-38
    - list A-181, A-185
    - parameters A-42
    - power A-38
  - shim
    - commands A-119, A-127

- definition file A-125
  - file A-122, A-127, A-130, A-131, A-260
  - increment A-127
  - set A-119, A-122
  - unit A-122, A-124, A-127, A-128, A-130
  - value A-122, A-124, A-131, A-220
  - shimming
    - automatically A-125, A-193, A-196
    - manually A-122, A-124
    - on the FID A-125
    - on the lock level A-126
    - on the lock signal A-114
    - using simplex A-127
  - simplex algorithm A-127
  - simplex AU program A-128
  - simulation pulse sequence A-268
  - simultaneous mode A-12
  - single channel detection A-12
  - sliders
    - on a probehead A-136, A-147
  - smilies A-53
  - soft pulse A-86, A-87
  - solid state A-33, A-101, A-157, A-240, A-243
  - solvent
    - acquisition parameter A-42
    - dependent lock parameters A-59, A-87, A-111
    - dependent parameters A-8, A-85, A-160, A-193, A-197, A-220, A-287
    - list A-93, A-104, A-110
    - table A-70, A-93
  - solvloop command A-87
  - spdisp command A-272
  - spectral width
    - acquisition parameter A-43
    - adjustment A-20
    - in 2D data A-26, A-30
    - maximum A-44
    - optimization A-277
    - parameter A-43, A-44
  - spectrometer
    - communication problems A-62
    - configuration A-61, A-62, A-70, A-96, A-286
    - constants A-89
    - frequency A-98, A-104
    - hardware A-64
    - installation A-64
    - interface A-207
    - internal CPU A-62
    - name A-62, A-64, A-286
    - routing A-32, A-34, A-57, A-95, A-156, A-158
    - type A-97, A-98
  - spectrum
    - display A-34, A-41
    - edges A-53
    - noise A-200
    - phase correction A-16, A-37, A-92
    - referencing A-58
    - resolution A-45
  - spin rate A-29, A-55, A-126, A-143, A-240, A-245, A-249, A-250, A-252, A-255
  - spinlock tolerance A-240
  - sref command A-29, A-43, A-59, A-105
  - stator of an MAS probe A-240, A-247, A-249, A-252
  - status of raw the data A-152
  - status of the raw data A-7, A-9
  - stdisp command A-25, A-42
  - steady state conditions A-19, A-221
  - stop command A-135, A-147, A-200, A-204, A-212, A-214, A-215, A-222
  - suspend command A-200, A-212, A-215
  - Switchbox A-33, A-38, A-45, A-58, A-96, A-157, A-159
- ## T
- T1 relaxation A-19, A-221
  - te2get command A-230
  - te2ready command A-234
  - te2set command A-236
  - teget command A-230
  - temon command A-231
  - temperature
    - acquisition parameter A-46
    - commands A-227
    - list A-50, A-181, A-184, A-280
    - maximum A-231
    - minimum A-231
    - monitor A-231
    - of the inlet capillary A-234
    - of the inlet capillary A-230
    - of the probe preamplifier A-276

- of the probehead coil A-276
- of the second regulator A-230, A-234, A-236
- parameter set A-233
- parameters A-228, A-229
- stabilization A-19, A-221, A-234
- unit A-46, A-185, A-227, A-228, A-229, A-230, A-233, A-236
- unit configuration A-68, A-70
- unit rs232 channel A-65, A-287
- temporary parameters A-8
- tepar command A-233
- teready command A-234
- teset command A-46, A-234, A-236
- tf3 command A-53
- time domain A-45, A-150
- TMS A-111
- TOCSY experiment A-30, A-86, A-161
- tomography A-101
- tr command A-216, A-221
- transfer data to disk A-216
- transmitter
  - blanking A-89
  - frequency A-145
  - gate A-14
  - power A-145
  - tuning A-242, A-257
- tube amplifiers A-257
- tune command A-87, A-121, A-124, A-125, A-127, A-193, A-196
- tune file A-87, A-121, A-125, A-126, A-193, A-196
- tuning knob A-146, A-147
- tuning the probehead A-51, A-134, A-135, A-136, A-145, A-192, A-193, A-196

## U

- user defined
  - AU programs A-69, A-81, A-172, A-264
  - CPD programs A-15, A-176
  - experiments A-194
  - gradient files A-24
  - gradient programs A-179, A-260
  - hard pulses A-87
  - parameter sets A-163, A-164, A-165, A-260
  - pulse programs A-40, A-177, A-188, A-215, A-260, A-265

- relations file A-87
- shape files A-42
- soft pulses A-87
- XWIN-NMR settings A-282
- User Interface A-173, A-176, A-179, A-188

## V

- variable
  - amplitude list A-47, A-183, A-187
  - counter list A-48, A-187, A-280
  - dataset list A-19, A-187
  - delay list A-49, A-180, A-187, A-280
  - frequency list A-187, A-280
  - pulse list A-49, A-187, A-280
  - temperature list A-50, A-187, A-280
- vish command A-130

## W

- well-plate A-56
- wide bore probe A-247, A-249, A-252
- wobb command A-51, A-134, A-136, A-145, A-192, A-193, A-196, A-220
- wobble
  - curve A-51, A-146, A-192
  - parameters A-135, A-137, A-148
  - procedure A-146
  - pulse program A-135, A-137, A-148
  - steps A-51, A-148
  - sweep width A-51, A-148
- wpar command A-164, A-167
- wsh command A-130, A-131

## X

- xau command A-97, A-217, A-260
- xaua command A-13, A-194, A-197, A-217
- xaup command A-217
- XCMD macro A-5
- xfb command A-197
- Xwin-nmr
  - command sequence A-5
  - display A-193, A-196, A-210, A-222
  - file handling A-3
  - graphics A-284
  - home directory A-4, A-63
  - installation A-39, A-63, A-96, A-97, A-172, A-

188  
 macros A-5, A-265  
 menu A-6  
 parameters A-7  
 patchlevel A-286  
 popup menus A-6  
 resource settings A-282  
 status line A-126  
 version A-87, A-135, A-137, A-160, A-200, A-202, A-210

**Z**

zg command A-7, A-8, A-11, A-192, A-200, A-220, A-221

**Numerics**

2D lookup tables A-265  
 4-phase modulator A-90, A-243

**A**

abs command A-36, A-53, A-197  
 ACB parameters A-274  
 ACB service tool A-287  
 accumulation of the FID A-192  
 acqu command A-221  
 acquisition  
   dimension A-44, A-46  
   interface A-193, A-196  
   menu A-192, A-196, A-202  
   mode A-12, A-44, A-52  
   order A-53  
   parameters A-7, A-8, A-9, A-10, A-11, A-154, A-163, A-164, A-165, A-167, A-192, A-202, A-204, A-208, A-280, A-281, A-282, A-289  
   status parameters A-7, A-9, A-52, A-152, A-280, A-281, A-282, A-291  
   time A-11, A-45, A-150  
 air flow A-142, A-247, A-249, A-252, A-253  
 amplifier  
   correction table A-72  
   linearization A-72  
   routing A-32, A-34, A-156, A-158  
 amplitude list A-181, A-183  
 Amplitude Setting Unit A-72  
 AMX spectrometer A-17, A-25, A-98, A-100, A-111, A-181  
 analog filter A-17, A-23  
 apk command A-53, A-197, A-223

ARX spectrometer A-98, A-101, A-111  
 as command A-150  
 ased command A-150, A-154, A-220  
 ASU  
   blanking A-89  
   correction A-72  
   linearization A-74  
 ATM probehead A-134, A-136, A-148, A-220  
 atma command A-51, A-134, A-136, A-220  
 atmm command A-136  
 attenuator A-72  
 AU program  
   binaries A-69, A-81, A-97, A-173, A-283  
   Bruker defined A-69, A-81, A-97, A-172  
   compilation A-69, A-81, A-172, A-217  
   creation A-172  
   deletion A-260  
   edit A-172  
   execution A-217  
   files A-283  
   for acquisition A-12, A-192, A-218  
   for processing A-218  
   installation A-97  
   kill A-218  
   macro A-4, A-5  
   rename A-264  
   sources A-69, A-81, A-173, A-283  
   user defined A-81, A-97, A-172  
 au\_zg AU program A-13, A-192  
 auto lock-in A-110, A-111, A-193, A-196  
 autolock key A-54, A-108, A-112, A-116  
 automatic  
   receiver gain optimization A-213  
   shimming A-121, A-125, A-128, A-193, A-196  
   tuning and matching A-134  
 automation with Icon-nmr A-5, A-43, A-66, A-87, A-106, A-140, A-206, A-218, A-277  
 autoshim A-120  
 Avance spectrometer A-5, A-31, A-98, A-100, A-111, A-181, A-284

**B**

basic frequency A-13, A-32, A-34, A-57, A-59, A-83, A-104, A-147, A-156  
 BPSU unit A-67, A-70  
 Bruker  
   AU programs A-69, A-81, A-97, A-172, A-200, A-264, A-283  
   CPD programs A-15, A-176

gradient files A-24  
 gradient programs A-179  
 parameter sets A-14, A-34, A-98, A-104, A-158,  
 A-164, A-168, A-218, A-260  
 pulse programs A-15, A-23, A-30, A-36, A-40,  
 A-177, A-188, A-215, A-265  
 shape files A-42  
 solvent list A-93  
 spectrometers A-17, A-227  
 Bruker addresses A-293  
**BSMS**  
 display A-104, A-122, A-143, A-287  
 keyboard A-8, A-54, A-58, A-104, A-106, A-108,  
 A-111, A-112, A-114, A-116, A-119, A-122,  
 A-128, A-142, A-143, A-146  
 lock unit A-111  
 rs232 channel A-65  
 service tool A-288  
 unit A-54, A-103, A-105, A-111, A-112, A-126,  
 A-143  
 bsmsdisp command A-104, A-122, A-143  
 butselau AU program A-195  
 butselnmr A-193  
 buttonau AU program A-198  
 buttonnmr command A-39, A-196, A-223

## C

calibration A-59, A-105  
 carrier frequency A-34, A-35, A-41, A-99, A-100  
 cf command A-13, A-32, A-62, A-97  
 cfbacs command A-66  
 cfbpsu command A-67  
 cfmas command A-240, A-249, A-253  
 cfte command A-67, A-68  
 chemical shift A-35, A-59, A-105, A-111  
 communication problems A-62  
 compilation of AU programs A-81, A-97, A-172, A-  
 217  
 compileall command A-69, A-97, A-217  
 composite experiments A-277  
 composite pulse decoupling A-14, A-176  
 config command A-32, A-62, A-70  
 configuration  
 commands A-61  
 files A-286  
 of the BPSU unit A-67  
 of the MAS unit A-240  
 of the sample changer A-66  
 of the spectrometer A-62

of the temperature unit A-68  
 of XWIN-NMR A-222  
 suite A-62, A-70  
 contour levels A-163, A-164, A-165, A-166, A-168,  
 A-169  
 conventions in this manual A-3  
**Cortab**  
 experiment A-72, A-76  
 linearization A-72  
 program A-72  
 verification A-72  
 cortab command A-72  
 counter list A-181  
**CPD**  
 command A-37  
 power level A-161  
 programs A-14, A-38, A-97, A-100, A-176, A-  
 187  
 pulse length A-37, A-151, A-161  
 cplbruk command A-81, A-97, A-217, A-260  
 cpluser command A-81, A-217  
 crplock command A-276  
 crpobs command A-276  
 crpoff command A-276  
 crpon command A-276  
 cryo preamplifier A-276

## D

data path parameters A-289  
 dataset list A-19, A-181  
 decimation factor A-16, A-21, A-53  
 define statement A-50, A-98  
 delau command A-260  
 delay list A-181  
 delete  
 AU programs A-260  
 gradient programs A-260  
 macros A-260  
 parameter sets A-260  
 pulse programs A-260  
 shim files A-260  
 various lists A-262  
 delgp command A-260  
 dellist command A-262  
 delmac command A-260  
 delpar command A-260  
 delpul command A-260  
 delsh command A-260  
 demand

spin rate A-143, A-240, A-246, A-249, A-252  
 temperature A-46, A-230, A-234, A-236, A-245  
 deuterium A-104  
 digital  
   digitizer resolution A-15, A-18  
   filtering A-5, A-15, A-16, A-17, A-20, A-21, A-53  
   lock A-103  
   quadrature detection A-12, A-18  
 digitizer  
   mode A-17  
   resolution A-15, A-18  
   start A-16  
   type A-11, A-17, A-18, A-23, A-100  
 dimension  
   2D direct A-4, A-9, A-32  
   2D indirect A-4, A-9, A-31, A-44, A-46  
   3D direct A-9  
   3D indirect A-5, A-9  
 dimensionality of data A-36, A-164  
 DNA/RNA experiments A-160  
 dpa command A-7, A-8, A-52, A-152  
 DQD  
   device A-12  
   mode A-18  
 DSP firmware A-54  
 dummy scans A-18, A-150, A-220, A-221, A-269  
 duty cycle A-274  
 dwell time A-17, A-20, A-43, A-150

## E

ed4ph command A-243  
 eda command A-7, A-8, A-10, A-11, A-161, A-164, A-168  
 edacb command A-274  
 edasp command A-13, A-21, A-25, A-31, A-32, A-34, A-38, A-57, A-220  
 edau command A-5, A-38, A-97, A-172, A-217  
 edc command A-168, A-193, A-196, A-220  
 edcgp command A-174  
 edcpd command A-15, A-176, A-187  
 edcpul command A-39, A-177, A-179  
 edg command A-164, A-168  
 edgp command A-179, A-187  
 edgx command A-164  
 edhead command A-40, A-56, A-85, A-104, A-138, A-220  
 edhpcu command A-242  
 edlev command A-164

edlist command A-19, A-23, A-47, A-48, A-49, A-50, A-180  
 edlock command A-8, A-29, A-87, A-104, A-108, A-110, A-116  
 edmac command A-5, A-187  
 ednuc command A-13, A-63, A-83  
 edo command A-105, A-164, A-168  
 edp command A-154, A-164, A-168  
 edprosol command A-8, A-39, A-43, A-85, A-125, A-161, A-220  
 edpul command A-36, A-187, A-188  
 edscon command A-16, A-89, A-224  
 edsolv command A-93  
 edsp command A-95, A-156  
 edte command A-228, A-233  
 edtg command A-229  
 edtune command A-121, A-125  
 ef command A-197  
 efp command A-194  
 ej command A-142  
 eject  
   air time A-240, A-247  
   mechanism A-240  
   the sample A-247, A-249, A-253  
   the spinner A-252  
 em command A-53, A-194, A-223  
 eurotherm temperature unit A-227  
 experiment time A-31, A-197, A-199  
 expinstall command A-17, A-40, A-69, A-81, A-96, A-97, A-98, A-164, A-172, A-188  
 expt command A-31, A-192, A-199

## F

FID  
   area A-127  
   display A-12, A-34, A-41, A-202, A-203  
   envelope A-125  
   number A-36  
   resolution A-21, A-46  
   signal A-40  
 field value A-104, A-105, A-114  
 follow command A-218  
 font conventions A-3  
 format files A-289  
 frequency  
   list A-23, A-181, A-182, A-225, A-280  
   unit A-268  
 ft command A-36, A-53, A-194, A-223

**G**

gethpcu command A-257  
 getlcosy command A-277  
 getlim1d command A-277  
 getlinv command A-277  
 getljres command A-277  
 getprosol command A-8, A-39, A-43, A-87, A-160,  
 A-193, A-197, A-220  
 getxhco command A-277  
 go command A-11, A-200, A-222  
 gpro command A-160  
 gradient  
   experiment A-198  
   file A-24, A-25, A-97, A-98  
   parameter table A-23  
   parameters A-24, A-25  
   program A-25, A-100, A-174, A-179, A-187, A-  
   260, A-265  
   pulse length A-161  
   recovery delay A-161  
   service tool A-288  
   shaped A-24  
   shimming A-25, A-220, A-282, A-287  
   static A-24  
   strength A-24  
 gradshim command A-220  
 group delay A-5, A-53, A-54  
 gs command A-11, A-12, A-34, A-41, A-192, A-  
 202, A-214

**H**

halt command A-31, A-135, A-200, A-204, A-212,  
 A-214, A-215, A-222  
 hard pulse A-86, A-87, A-161  
 hardware list A-64, A-66, A-240, A-286  
 hardware resolution A-15, A-18  
 high power  
   accessory A-38  
   amplifier A-25, A-33, A-157  
   cabinet A-257  
   output A-33, A-157  
   preamplifier A-25  
   routing A-58, A-100, A-159  
   tube transmitter A-242, A-257  
 holder position A-56  
 homodecoupling  
   duty cycle A-25  
   experiment A-17  
   power level A-161

homogeneity of the magnetic field A-114, A-125  
 hostname A-62  
 hosts file A-62, A-64  
 HPCU  
   parameters A-242  
   service tool A-288  
   unit A-257  
 HPPR preamplifier A-25, A-147, A-276  
   service tool A-288

**I**

iconnmr command A-206  
 ii command A-11, A-207  
 ij command A-142  
 impedance  
   of a probehead A-145, A-146, A-147  
 input parameters A-8  
 insert  
   airtime A-240, A-246, A-247, A-249, A-252  
   the sample A-142, A-246, A-247  
   the spinner A-247, A-249, A-252  
 integral ranges A-187, A-277  
 interface  
   of the spectrometer A-207  
 irradiation frequency A-23, A-32, A-34, A-35, A-  
 41, A-57, A-59, A-99, A-100, A-111, A-151, A-  
 156, A-182, A-202  
 irradiation frequency offset A-35, A-57, A-59, A-  
 105, A-182, A-202

**K**

kill command A-135, A-204, A-214, A-218, A-222

**L**

LC-NMR A-67, A-70, A-160  
 lfilter command A-106, A-108  
 lgain command A-55, A-106, A-108  
 liprosol command A-87  
 lock  
   channel A-276  
   display A-106, A-114  
   display file A-115  
   frequency A-104, A-114  
   gain A-110  
   irradiation frequency A-59  
   level A-126, A-127  
   nucleus A-28, A-104, A-114  
   parameters A-7, A-29, A-58, A-70, A-104, A-  
   110, A-116



phase A-59, A-105, A-106, A-110, A-114, A-116, A-128  
 power A-55, A-59, A-104, A-105, A-106, A-110, A-116  
 regulator cut-off frequency A-55, A-59, A-104  
 regulator filter A-108  
 regulator gain A-54, A-59, A-104, A-108  
 regulator time A-55, A-59, A-104, A-108  
 shift A-55  
 signal A-105, A-106, A-110, A-112, A-114, A-116  
 status A-56  
 sweep A-106  
 table A-104  
 lock command A-8, A-29, A-43, A-58, A-105, A-108, A-110, A-116, A-193, A-196, A-220  
 lockdisp command A-106, A-112, A-114, A-220  
 lock-in procedure A-58, A-110, A-111, A-116  
 logical frequency channel A-21, A-32, A-58, A-86, A-100, A-156, A-159  
 loop counter A-27  
 loop filter A-55, A-59, A-104, A-106, A-108, A-111, A-116  
 loop gain A-54, A-59, A-106, A-108, A-111, A-116  
 loop time A-55, A-59, A-104, A-106, A-108, A-111, A-116  
 loopadj AU program A-106, A-108  
 lopo command A-8, A-29, A-43, A-58, A-105, A-112, A-116  
 lopoi command A-29, A-116  
 ltime command A-106, A-108

## M

macros  
   in AU programs A-5, A-283  
   in XWIN-NMR A-5, A-187, A-260, A-262, A-265  
 magic angle A-239, A-249, A-252  
 magnet  
   air flow A-142  
   strength A-62  
   type A-66, A-245  
 MAS  
   commands A-253  
   control A-29, A-245, A-249, A-252  
   log file A-250  
   manual A-253  
   monitor A-55, A-255  
   parameters A-240, A-245

pneumatic unit A-70, A-240, A-245, A-246, A-247, A-249, A-252, A-286  
 spin rate A-29, A-55, A-249, A-253  
 spin rate list A-181, A-187  
 mas command A-29, A-245, A-249, A-252  
 mascontrol command A-252  
 mase command A-252  
 masg command A-252  
 mash command A-252  
 masi command A-252  
 masr command A-29, A-55, A-252  
 masrmon command A-55, A-249, A-255  
 matching knob A-146, A-147  
 matching the probehead A-51, A-134, A-135, A-136, A-145, A-192, A-193, A-196  
 mc command A-46  
 micro imaging A-101  
 mixed solvents A-111  
 mixing time A-86  
 multiefp AU program A-194  
 multimas AU program A-186  
 multizg AU program A-223

## N

new dataset A-168, A-193, A-196, A-220  
 NMR Superuser A-13, A-62, A-97, A-153, A-155, A-172  
 nuclei  
   list A-34, A-83  
   selection A-156  
   table A-13, A-65, A-83, A-286

## O

O1 spikes A-12  
 observe nucleus A-59, A-105  
 optimization  
   criterion A-208  
   of acquisition parameters A-202, A-208  
   of the lock parameters A-110  
   of the receiver gain A-213  
   protocol of popt A-208  
 output device parameters A-163, A-164, A-165, A-168  
 output parameters A-7, A-8  
 overflow A-18, A-35, A-213  
 oversampling A-16, A-17, A-18, A-20, A-23, A-53  
   dwell time A-17, A-20

**P**

- parameter
  - editor A-8, A-10
  - optimization A-202, A-208
  - sets A-97, A-98, A-163, A-167, A-168, A-218, A-260, A-264
  - types A-163, A-164, A-167, A-168, A-260
  - value types A-10
- paropt AU program A-210
- paroptlin AU program A-210
- paroptlog AU program A-210
- parray AU program A-210
- phase
  - alignment A-42, A-43, A-86
  - correction factor A-37
  - correction of a pulse A-37
  - correction of the receiver A-37
  - correction of the spectrum A-16, A-92
  - cycle A-19, A-37, A-200, A-204, A-222
  - list A-202
  - of a pulse A-268
  - of the lock signal A-59, A-105, A-106, A-111, A-114, A-116, A-128
  - of the receiver A-16, A-91, A-268
  - program A-36
  - values of a shaped pulse A-185
- phtran AU program A-37
- pk command A-194
- plot
  - parameters A-7, A-163, A-164, A-165, A-168
  - title A-163, A-164, A-165, A-168
- plot command A-194, A-197
- popt command A-208
- portfolio A-282
- power
  - level A-8, A-38, A-42, A-43, A-86, A-87, A-151, A-161, A-202, A-268, A-269
  - list A-183
  - mode A-38
- preamplifier
  - blanking A-89
  - gain A-39
  - modules A-25, A-33, A-38, A-58, A-96, A-100, A-157, A-158, A-159
  - of a cryoprobe A-276
  - of an MSL A-39
  - routing A-32, A-33, A-58, A-96, A-99, A-156, A-157, A-158, A-159
  - switching A-16, A-91
  - tuning and matching A-146
  - preferred output for 19F A-33, A-58, A-96, A-99, A-157, A-158, A-159
- preferred preamplifier A-33, A-58, A-96, A-99, A-157, A-158, A-159
- pre-scan
  - delay A-15, A-91, A-150
  - subdelays A-16, A-91
- probehead A-40, A-104
  - acquisition parameter A-56
  - ATM A-134, A-136
  - broadband A-136, A-147
  - commands A-133
  - cryo A-276
  - current A-104, A-138, A-139, A-220
  - dependent lock parameters A-87
  - dependent parameters A-8, A-43, A-59, A-85, A-160, A-197, A-220
  - dependent tune file A-87
  - impedance A-145
  - name A-138
  - parameter editor A-138
  - parameters A-138, A-139
  - resonance circuit A-145
  - resonance frequency A-136, A-145, A-147
  - table A-70, A-138
  - tune file A-193
  - tuning and matching A-51, A-134, A-136, A-145, A-193, A-196, A-220
  - type A-245, A-249, A-252
- processing
  - commands A-3, A-7, A-36, A-46, A-52, A-53, A-54, A-223
  - menu A-192
  - parameters A-7, A-163, A-164, A-165, A-168
- prosol command A-87
- prosol parameters A-7, A-8, A-39, A-85, A-160
- protein experiments A-160
- pulsdisp command A-11, A-268
- pulse channels A-268
- pulse length
  - CPD A-37, A-151, A-161
  - decrement A-26
  - display A-271
  - increment A-26
  - of a 180 degree pulse A-161
  - of a 90 degree pulse A-86, A-145, A-161
  - of a gradient pulse A-161
  - of a hard pulse A-86
  - of a soft pulse A-86

parameter A-35  
reset A-26  
pulse list A-181  
pulse program  
  Bruker defined A-177, A-188  
  compilation A-220  
  constants A-13  
  creation A-177, A-188  
  current A-177  
  define statements A-98  
  deletion A-260  
  display A-268  
  edit A-177, A-188  
  for AMX A-100  
  for wobble A-135  
  installation A-97  
  label A-27, A-30, A-49, A-183  
  rename A-265  
  timing A-268  
  user defined A-177, A-188  
pulsesort AU program A-160

## Q

QNP probe A-40  
qnpset AU program A-40  
quad images A-12  
quadrature detection A-12

## R

rackpow command A-257  
raw data A-5, A-7, A-9, A-30, A-36, A-45, A-152,  
  A-164, A-192, A-197, A-200, A-204, A-222, A-  
  224, A-280, A-281  
receiver  
  channel A-40  
  control A-268  
  gain A-40, A-150, A-202, A-203  
  gain optimization A-213  
  gate A-16, A-91  
  output A-41  
  phase A-16, A-37, A-91, A-268  
  rs232 channel A-65, A-287  
  service tool A-288  
  status A-268  
recycle delay A-19, A-212, A-221  
reference  
  50 ohm resistor A-145  
  frequency A-40, A-100  
  number A-93, A-138

peak A-98  
phase A-202  
shift A-59, A-105  
signal A-59, A-105  
substance A-13, A-111  
referencing the spectrum A-104  
relations  
  file A-87, A-160, A-280  
  prosol-acquisition A-87, A-160, A-280  
relaxation delay A-151  
rename  
  2D lookup tables A-265  
  AU programs A-264  
  gradient programs A-265  
  lists A-266  
  macros A-265  
  parameter sets A-264  
  pulse programs A-265  
renau command A-264  
rengp command A-265  
renlist command A-266  
renlut command A-265  
renmac command A-265  
renpar command A-264  
renpul command A-265  
resonance circuit A-145, A-147  
resonance frequency  
  of a probehead A-136, A-145, A-146, A-147  
resume command A-11, A-200, A-212  
rga command A-11, A-213  
ro command A-41, A-66, A-143  
ROESY experiment A-86, A-161  
rotation of the sample A-40, A-66, A-126, A-134,  
  A-143, A-144, A-146, A-249  
rotor material A-245  
routing  
  parameters A-34, A-96, A-158  
  table A-13, A-32, A-57, A-156, A-157  
rpar command A-17, A-34, A-36, A-40, A-163, A-  
  167, A-168, A-193, A-196, A-220  
rsh command A-122, A-128, A-220  
RX22  
  rs232 channel A-65, A-287  
  service tool A-288  
RX22 receiver A-16, A-91

## S

sample  
  changer A-66, A-70

- depth A-139
- diameter A-138, A-139, A-240, A-245
- eject A-142, A-247
- insert A-142, A-246, A-247
- position A-142
- rotation A-66, A-126, A-134, A-143, A-146, A-249
- rotation frequency A-40, A-144
- solvent A-42, A-59
- temperature A-19, A-221, A-230, A-234, A-236
- transfer tube A-247, A-249
- saturation of the lock signal A-106
- scaling region file A-98, A-187
- SCM unit A-111
- SE451 receiver A-41
- sensitivity of GS sliders A-203
- sequential mode A-12, A-18
- sethpcu command A-257
- setres command A-173, A-223
- setsh command A-124
- setti command A-164
- SGU verification A-74
- shape files A-42, A-86, A-98, A-101
- Shape Tool A-25, A-42
- shaped gradient
  - file name A-225
- shaped pulse
  - file name A-225, A-280
  - frequency offset A-99
  - in CPD programs A-38
  - list A-181, A-185
  - parameters A-42
  - power A-38
- shim
  - commands A-119, A-127
  - definition file A-125
  - file A-122, A-127, A-130, A-131, A-260
  - increment A-127
  - set A-119, A-122
  - unit A-122, A-124, A-127, A-128, A-130
  - value A-122, A-124, A-131, A-220
- shimming
  - automatically A-125, A-193, A-196
  - manually A-122, A-124
  - on the FID A-125
  - on the lock level A-126
  - on the lock signal A-114
  - using simplex A-127
- simplex algorithm A-127
- simplex AU program A-128
- simulation pulse sequence A-268
- simultaneous mode A-12
- single channel detection A-12
- sliders
  - on a probehead A-136, A-147
- smilies A-53
- soft pulse A-86, A-87
- solid state A-33, A-101, A-157, A-240, A-243
- solvent
  - acquisition parameter A-42
  - dependent lock parameters A-59, A-87, A-111
  - dependent parameters A-8, A-85, A-160, A-193, A-197, A-220, A-287
  - list A-93, A-104, A-110
  - table A-70, A-93
- solvloop command A-87
- spdisp command A-272
- spectral width
  - acquisition parameter A-43
  - adjustment A-20
  - in 2D data A-26, A-30
  - maximum A-44
  - optimization A-277
  - parameter A-43, A-44
- spectrometer
  - communication problems A-62
  - configuration A-61, A-62, A-70, A-96, A-286
  - constants A-89
  - frequency A-98, A-104
  - hardware A-64
  - installation A-64
  - interface A-207
  - internal CPU A-62
  - name A-62, A-64, A-286
  - routing A-32, A-34, A-57, A-95, A-156, A-158
  - type A-97, A-98
- spectrum
  - display A-34, A-41
  - edges A-53
  - noise A-200
  - phase correction A-16, A-37, A-92
  - referencing A-58
  - resolution A-45
- spin rate A-29, A-55, A-126, A-143, A-240, A-245, A-249, A-250, A-252, A-255
- spinlock tolerance A-240
- sref command A-29, A-43, A-59, A-105
- stator of an MAS probe A-240, A-247, A-249, A-252
- status of raw the data A-152

status of the raw data A-7, A-9  
stdisp command A-25, A-42  
steady state conditions A-19, A-221  
stop command A-135, A-147, A-200, A-204, A-212,  
A-214, A-215, A-222  
suspend command A-200, A-212, A-215  
Switchbox A-33, A-38, A-45, A-58, A-96, A-157,  
A-159

## T

T1 relaxation A-19, A-221  
te2get command A-230  
te2ready command A-234  
te2set command A-236  
teget command A-230  
temon command A-231  
temperature  
  acquisition parameter A-46  
  commands A-227  
  list A-50, A-181, A-184, A-280  
  maximum A-231  
  minimum A-231  
  monitor A-231  
  of the inlet capillary A-234  
  of the inlet capillary A-230  
  of the probe preamplifier A-276  
  of the probehead coil A-276  
  of the second regulator A-230, A-234, A-236  
  parameter set A-233  
  parameters A-228, A-229  
  stabilization A-19, A-221, A-234  
  unit A-46, A-185, A-227, A-228, A-229, A-230,  
  A-233, A-236  
  unit configuration A-68, A-70  
  unit rs232 channel A-65, A-287  
temporary parameters A-8  
tepar command A-233  
teready command A-234  
teset command A-46, A-234, A-236  
tf3 command A-53  
time domain A-45, A-150  
TMS A-111  
TOCSY experiment A-30, A-86, A-161  
tomography A-101  
tr command A-216, A-221  
transfer data to disk A-216  
transmitter  
  blanking A-89  
  frequency A-145

  gate A-14  
  power A-145  
  tuning A-242, A-257  
tube amplifiers A-257  
tune command A-87, A-121, A-124, A-125, A-127,  
A-193, A-196  
tune file A-87, A-121, A-125, A-126, A-193, A-196  
tuning knob A-146, A-147  
tuning the probehead A-51, A-134, A-135, A-136,  
A-145, A-192, A-193, A-196

## U

user defined  
  AU programs A-69, A-81, A-172, A-264  
  CPD programs A-15, A-176  
  experiments A-194  
  gradient files A-24  
  gradient programs A-179, A-260  
  hard pulses A-87  
  parameter sets A-163, A-164, A-165, A-260  
  pulse programs A-40, A-177, A-188, A-215, A-  
  260, A-265  
  relations file A-87  
  shape files A-42  
  soft pulses A-87  
  XWIN-NMR settings A-282  
User Interface A-173, A-176, A-179, A-188

## V

variable  
  amplitude list A-47, A-183, A-187  
  counter list A-48, A-187, A-280  
  dataset list A-19, A-187  
  delay list A-49, A-180, A-187, A-280  
  frequency list A-187, A-280  
  pulse list A-49, A-187, A-280  
  temperature list A-50, A-187, A-280  
vish command A-130

## W

well-plate A-56  
wide bore probe A-247, A-249, A-252  
wobb command A-51, A-134, A-136, A-145, A-  
192, A-193, A-196, A-220  
wobble  
  curve A-51, A-146, A-192  
  parameters A-135, A-137, A-148  
  procedure A-146  
  pulse program A-135, A-137, A-148

- steps A-51, A-148
- sweep width A-51, A-148
- wpar command A-164, A-167
- wsh command A-130, A-131

## **X**

- xau command A-97, A-217, A-260
- xaua command A-13, A-194, A-197, A-217
- xaup command A-217
- XCMD macro A-5
- xfb command A-197
- Xwin-nmr
  - command sequence A-5
  - display A-193, A-196, A-210, A-222
  - file handling A-3
  - graphics A-284
  - home directory A-4, A-63
  - installation A-39, A-63, A-96, A-97, A-172, A-188
  - macros A-5, A-265
  - menu A-6
  - parameters A-7
  - patchlevel A-286
  - popup menus A-6
  - resource settings A-282
  - status line A-126
  - version A-87, A-135, A-137, A-160, A-200, A-202, A-210

## **Z**

- zg command A-7, A-8, A-11, A-192, A-200, A-220, A-221