CARESS Document Set

User Commands

Busse/Brehmer/Sauer

Printed on December 13, 2013
SVN-Revision: 1701 (2010-09-01)

Abstract: This document, part 2 of the CARESS document set, lists all the CARESS user commands. It is both the command reference and the command user's guide and has been updated to reflect the changes in the current CARESS release.
Users new to CARESS will find here descriptions of the command's structure and purposes. All available commands are listed with some notes about the meaning and a 'real world' example.

Contents

1 Overview

Commands serve different purposes:

Gerneral setup define an user experiment, names, titles, comments and parameters

Special setups dealing with polarisation, coordinate transformations and some others

Data acquisition performing basic actions und run scans

Special actions like reference movements and universal device access

Special hardware access to deal with complex hardware systems

Flow control do loops and control the behaviour

Miscellaneous diverse useful pieces

2 Command Structure

CARESS command input is requested by the prompt

```
CAR>
```

in case of exceptional situations with

```
EXC>
```

An input line may contain only one command, but a command may spread several lines. If the last character is '-', a continuation line is expected. Comment lines start with ';'. A comment starts with the command *COM*. A comment on a command line starts with '!'.
Any sequence of CARESS commands can be read from a text file generated using an editor or the CARESS GUI. Input from such a command file (e.g. `mycommands.com`[1]) is invoked by

```
CAR>@mycommands
```

Command files may be nested.

A command starts with a command key which is optionally followed by a parameter list, the parameter list may be

- a list of values, e.g. `DS 3.3542 3.3542`
- a list of subkeys (mostly module names) with or without values, e.g.

```
TOL OMGS=0.02,PHIS=0.01
SL1 TIM1,SDET
```

Some[2] commands are *additive*, i.e. information in several occurrences of the same command is added (merged) e.g.

```
BLS OMGS=-100 100
BLS PHIS=0 75
```

have the same effect as

[1] Note: the suffix *.com* is mandatory.

[2] *valid for: BLS, DEFOFF, DEFON, DEFTURN, PRW, PRWA, SOF, TMO, TOL, WIND, WINDA, SELDET, DEFPOS, DEFINE, DEFSAM*

```
   BLS OMGS=-100 100 PHIS=0 75
```

All other commands are not additive, and the information of the last occurrence of such a command is relevant, only.
To obtain the current coverage of a command (with no matter if additive or not) it is usefull to call the CARESS `show` command.
For the example above:

```
   CAR> SHOW BLS
```

yields to this message on the protocol window:

```
BLS OMGS=-100 100 PHIS=0 75
```

3 Commands

This chapter lists all the CARESS commands. Which commands are accepted at a specific instrument (or in a specific situation) depends on the functionality and setup.
Typographical notes concerning the following command definitions:

{. . .}	means *repetition, at least one occurrence*
[. . .]	means *optional*
$\langle real\rangle$	::= *PASCAL* notation, in addition: $\{\langle digit\rangle\}\vert\{\langle digit\rangle\}.$
$\langle identifier\rangle$	::= $\langle letter\rangle\{\langle letter\ or\ digit\ or\ \_\rangle\}$, max. 20 characters
$\langle text\rangle$	::= $\{\langle symbol\rangle\}$, max. 68 characters
$\langle symbol\rangle$	::= $\langle program\ name\rangle$
$\langle string\rangle$	::= $\langle identifier\rangle\vert\,'\langle text\rangle'$
$\langle file\_name\rangle$	::= $\langle identifier\rangle$
$\langle peakinfo\rangle$	::= $ON\vert OFF\vert$
	$LL\ \langle beta\rangle\langle delta\_b\rangle\langle peak\ form\ factor\rangle\vert$
	$NLL\ \langle left\ points\rangle\langle right\ points\rangle$

Each command is listed with:

- command purpose (short)

  command syntax

  ```
  command example
  ```

4 General setup commands

Most of the commands of this group are essential for all of the supported instruments.

4.1 Definition of an User Experiment

- Instrument selection information

  EXPTYPE ⟨*instrument*⟩⟨*axes or mode*⟩⟨*detector*⟩⟨*exptype*⟩⟨*supplementary equipment*⟩

  ```
  EXPTYPE E1 A3 SDET EULER TEMP
  EXPTYPE V1, A2, ADET, POWD
  ```

- List of modules depending on EXPTYPE command

  ONL {⟨*module*⟩}

  ```
  ONL OMGS TTHS TEMP
  ```

4.2 Definition of Names, Title, Comment, etc.

- User Name

  USN ⟨*identifier*⟩

  ```
  USN MILLER
  ```

- Title

  TITLE ⟨*text*⟩

  ```
  TITLE This is a title.
  ```

- Sample Name

  SAM ⟨*identifier*⟩

  ```
  SAM CU
  ```

- List of samples (for V4)

  DEFSAM {⟨*pos_no*⟩⟨*string*⟩⟨*value*⟩⟨*value*⟩}

  if sample changer

  ```
  DEFSAM 1 ‘HMI#005‘ 0.0 0.0, 2 B60 0.0 0.0
  ```

otherwise

```
DEFSAM 1 ‘HMI#005‘ 0.0 0.0
```

- Comment

  COM ⟨*text*⟩

```
COM This is a comment.
```

4.3 Parameters for Driving Modules

- Limits (Range)

  BLS {⟨*module*⟩ = ⟨*lower limit*⟩, ⟨*upper limit*⟩}

```
BLS OMGM=0,100 TTHS= -50,70 OMGS=40,80
```

- Offset (real position = encoder_value + offset_value)

  SOF {⟨*module*⟩ = ⟨*real*⟩}

```
SOF TTHM=0 OMGS=20.5
```

- Tolerance

  TOL {⟨*module*⟩ = ⟨*tolerance value*⟩}

```
TOL OMGS=0.02 PHIS=0.01
```

- Timeout in seconds (for devices with known velocity: supplementary timeout)

  TMO {⟨*module*⟩ = ⟨*timeout value*⟩}

```
TMO OMGS=10, TEMP=200
```

4.4 Definition of Acquisition Parameters

- Master (preset) Counter

  MM1 ⟨*module*⟩ ⟨*preset value*⟩

```
MM1 MON 10000
```

- Slave (acquisition) Modules

  SL1 {⟨*module*⟩}

```
SL1 TIM1, TTHS, LDET, TEMP
```

- Protocol Modules (read at begin and at termination of (partial) data acquisition)

  PRO {⟨*module*⟩}

```
PRO TEMP
```

- Definition of Supplementary Modules to be journalised like a SET Module during a Scan.

  SCANV {⟨*module*⟩}

  ```
  SCANV MAGF
  ```

- Definition of Permanent Movement during Acquisition

  ACQPMOVE ⟨*module*⟩ = ⟨*pos*1⟩⟨*pos*2⟩

  ```
  ACQPMOVE OMGS=-100 100
  ```

4.5 Detector related parameters

- The user can SELect one or more (up to 4) DETectors out of the single detectors and/or multi detector for special purpose (define cell or region of cells). This command applies to the Data Acquisition Unit (DAU) only. The predefined names for selected detectors are D1, D2, D3 and D4. You can omit parameters, they are assumed as zero. For a single detector the y_chan should be zero in any case.

  SELDET {*Di* = ⟨*x_chan*⟩[⟨*y_chan*⟩[⟨*radius*⟩]]}

  select 4 cells of a linear detector:
  ```
  SELDET D1=17 D2=20 D3=21 D4=2
  ```
  select a region: cells 14 ... 20 of a 1-dim multi detector:
  ```
  SELDET D1= 17, 0, 3
  ```
  select a region of an area detector:
  ```
  SELDET D1 = 10, 20, 5
  ```

- Erase the selection of detector cell or region of cells of a detector.

  RESET SELDET

  ```
  RESET SELDET
  ```

- Up to 4 Windows for Linear Multi Counter (i = 1 ... 4)

  WIND {Wi = ⟨*lower channel*⟩, ⟨*upper channel*⟩}

  ```
  WIND W1=20 80 W2=150 250
  ```

- Up to 4 Windows for Area Detector (i = 1 ... 4)

  WINDA {Wi = ⟨*lower_x_chan*⟩, ⟨*upper_x_chan*⟩, ⟨*lower_y_chan*⟩, ⟨*upper_y_chan*⟩}

  ```
  WINDA W1 = 20 80 60 128
  ```

- Correction of Multi Detector Counts (CARESS will append extension .dat to the filename)

  CALIB ⟨*module*⟩ = ⟨*file_name*⟩

  ```
  CALIB LDET = e2_ldet_corr
  ```

- Correspondence Detector Angle ⇔ channel number of multi counter

  RELA $\{\langle module\rangle = \langle channels\rangle, \langle ref\_channel\rangle, \langle cell\_distance\rangle\}$ –
  $[, \langle distance\ sample - detector\rangle = 0]$

  1-dim multi detector with fixed distance sample-detector, cell_distance in degrees:
  ```
  RELA TTHS = 400 1 0.2
  ```
  area detector with variable distance sample-detector, cell_distance in cm:
  ```
  RELA TTHS = 128 1 0.1, NYS=128,1,0.1, YSD=0
  ```

  Note:

  > `RELA` sets the parameters for the detector given by `EXPTYPE` (refer to page 6). If more than one area detector is involved, use `MRELA`.

- Correspondence Detector Angle ⇔ channel number of multi counters

  MRELA $\langle detector\rangle\{\langle module\rangle = \langle ref\_channel\rangle, \langle offset\rangle\}$

  two 2-dim multi detectors, absolute distances:
  ```
  MRELA ADET1 TTHS=128 0, NYS=128,0, YSD=0,0
  MRELA ADET2 TTHS=128 0, NYS=128,0, YSD=0,0
  ```
  two 2-dim multi detectors, relative distances:
  ```
  MRELA ADET1 TTHS=128 64, NYS=128,128, YSD=0,100
  MRELA ADET2 TTHS=128 TTH1, NYS=128,64 YSD=0,100
  ```

  Note:

  > Each area detector is to define with one `MRELA` command. A complete setup includes a corresponding entry in `corbadevice.dat` (`startcommands += MRELA`).
  > The `offset` is relative to the `reference channel` of the `module`.

4.6 Define Online Protocol and Plot of Scan Data

- Protocol Parameters

  PRT ON/OFF/FUL/SHT
  FUL: full, SHT: short

  ```
  PRT FUL
  ```

- Define modules to be printed in the protocol terminal

  PRTMOD $\{\langle device\rangle\}$

  ```
  PRTMOD OMGX, RESIST, MON2
  ```

  Note:

  > This command has no effect on the data file it just reduces the number of *online* printed protocol modules to fit screen sice etc. The files contain all *SL1* and *PRO* modules.

- Define Protocol Windows one dimensional (in case of SDET or LDET)

  PRW $\{Wi = \langle lower\_channel\rangle\langle upper\_channel\rangle\}$
  (i=1 . . . 4)

  ```
  PRW W1= 5 196
  ```

  two dimensional (in case of ADET)

  PRWA $\{Wi = \langle lower\_x\_ch\rangle\langle upper\_x\_ch\rangle\langle lower\_y\_ch\rangle\langle upper\_y\_ch\rangle\}$
  (i = 1 .. 4)

  ```
  PRWA W1 = 1 128 1 80
  ```

- Peak calculation
  LL: method lehman-larsen;
  NLL: not LL

  PEAK $\langle peakinfo\rangle$

  ```
  PEAK ON
  PEAK LL,-1,-1,0.5
  PEAK NLL 8,5
  ```

- Switch OFF/ON Online Plot, define Refresh Time (in seconds)

  DISP OFF/ON/LIV

  ```
  DISP LIV 10
  ```

- Define Histogram Axes

  SPEC $[\langle x\_axis\rangle]\langle y\_axis\rangle$
  $\langle x\_axis\rangle$ *: any scan module,*
  $\langle y\_axis\rangle$ *: any slave module*

  ```
  SPEC OMGA SDET
  SPEC LDET
  ```

- Supervise environment modules by saving their values periodically

  LOGDEV PERIOD = $\langle seconds\rangle\{\langle environment\ module\rangle\}$
  (save action not yet implemented)

  ```
  LOGDEV PERIOD=60 TEMP,MAGF
  ```

4.7 Reset internal setup lists

- Reset / Delete Command Parameters
  with commands: CALIB, SELDET, PERMOVE, PASS, TOF, SCANV, WAIT1, WAIT2, ONL, OFL, SL1, PRO, WIND, WINDA, PRW, PRWA, RESOSTEP, SPEC

RESET {⟨*commandkey*⟩}

```
RESET PASS
```

5 Special setups

The following commands will apply to some special configurations at some instruments.

5.1 Polarisation

- Definition of OFF/ON Values for Power Supplies

  DEFOFF {⟨*module*⟩ = ⟨*off position*⟩}
  or
  DEFON {⟨*module*⟩ = ⟨*on position*⟩}

  ```
  DEFON HFI1=1.5 HFI2=1.0 HFI3=1.5
  ```

- Definition of Spin Turner IN and OUT (3D polarisation, only E1)

  DEFTURN YIN = ⟨9 *values*⟩*ZIN* = ⟨9 *values*⟩*VFI*1 = ⟨9 *values*⟩ -
  YOUT = ⟨9 *values*⟩*ZOUT* = ⟨9 *values*⟩*HFI*1 = ⟨9 *values*⟩

- Definition of a Polarisation Cycle with POLB (polarisation begin) and POLE (polarisation end)

  POLB
  SET {⟨*polarisation_module*⟩ = ⟨*targetposition*⟩}   if V6
  SET {⟨*polarisation_module*⟩ = *ON*/*OFF*}   otherwise
  MES [⟨*presetvalue*⟩]
  ...
  POLE

  ```
  POLB
  SET HF=OFF VF=OFF FL1=OFF FL2=OFF
  MES
  SET HF= ON VF=OFF FL1=OFF FL2=OFF
  MES 2500
  POLE
  ```

- Switch ON/OFF Polarisation to Execute the Polarisation Cycle

  POL ON/OFF

  ```
  POL ON
  ```

  Note:

The definition and execution of a measuring cycle (POLB, POLE, POL) may be freely used (independent of polarisation!).

5.2 Coordinate Transformations (Angles ⇔ H,K,L, ...)

- Lattice Parameter

  LAT ⟨6 *values*⟩

  ```
  LAT 7.235 7.235 1.8242 90 90 90
  ```

- UB matrix
  (number of significant parameters depends on instrument command e.g.
  E2 with FLAT: only 1st, 2nd, 4th, 5th parameter meaningful)

  UBM ⟨9 *values*⟩

  ```
  UBM ...
  ```

- Wave Length

  WAV ⟨*wavelength*⟩

  ```
  WAV 2.43138
  ```

- Method of Conversion H,K,L,E into Angle Values
  (conversion method depends on instrument and EXPTYPE command)

  MSH ⟨*conversion method*⟩[, ⟨*alternative*⟩]

  with conversion method:
  0: bisectric (PHIS ≠ 0), or 4 circles
  1: bisectric (PHIS = 0), or 2 circles
  2: parallel
  3: multi counter inclination (flat cone)
  4: 2 circles, OMGX
  5: goniometer (only E4)

  ```
  MSH 0
  MSH 1, 2[1]
  ```

- Passive Modules (angle positions calculated from e.g. H, K, L will not be driven)

  PASS {⟨*module*⟩}

  ```
  PASS TTHS
  ```

- Calculate Angles

  CALC {⟨*module*⟩ = ⟨*value*⟩}

[1] alternative 2 is selected

```
CALC H=1 K=2 L=1 E=0.1
```

- Calculate H, K, L (only for A2, EULER)

  $CALCHKL\ OMGS = \langle value\rangle TTHS = \langle value\rangle PHIS = \langle value\rangle CHIS = \langle value\rangle$

```
CALCHKL OMGS=12.4 TTHS=2.2 PHIS=16.8 CHIS=0.1
```

5.3 Diverse special

- D Spacing monochromator[, analyzer]

  DS ⟨*value*⟩[⟨*value*⟩] different number of significant parameters

```
DS 3.3542 3.3542
```

- Scattering Sense monochromator, sample[, analyzer]
  1: left, -1: right (reverse for E10)

  SS ⟨2 *values*⟩[⟨*value*⟩] different number of significant parameters!

```
SS 1 1 1
```

- Fix initial (0) or final (1) Energy wave vector, wave length

  FX 0/1

```
FX 1
```

- Energy Unit
  (0: THz, 1: meV)

  EU 0/1

```
EU 1
```

- Definition of Resolution Steps

  RESOSTEP ⟨*number of steps*⟩

```
RESOSTEP 15
```

- Definition of Partial Resolution Acquisition

  RESOPART ⟨*reso_steps*⟩⟨*start_step*⟩⟨*end_step*⟩
  (start_step⟨= end_step⟨= reso_steps)

```
RESOPART 25 13 25
```

- Select the *omega* module used for scans and conversions

  OMEGA_MOD ⟨*omega module*⟩

```
OMEGA_MOD OMEGAX
```

Note:

The default is *OMGS*; a definition could be reset via *RESET OMEGA_MOD.*

- Coupling set modules

  CSM {⟨modules⟩}

  ```
  CSM TTHS FKM
  CSM TEMP MAGF POSX
  ```

  Note:

  This command defines dependencies of modules according to a lookup table; multiple sets of different modules are allowed; subsequent set and scan commands will use these definitions to determine the setpoint positions of dependend modules; the set command must not contain more than one module of a csm definition;
  all sets are cleared via *RESET CSM.*

- Select action after a fatal error (a device requested a RESET)

  FATAL_ERROR ⟨action⟩
  with action:
  IGNORE: try to continue scan
  HALT: (default) halt scan and ask user
  ABORT: abort scan and wait for next command

  ```
  FATAL_ERROR HALT
  ```

6 Data acquisition and other actions

This set of commands causes actions with real hardware access, e.g. readout or movement to new set points of devices.

6.1 Perform Basic Actions

- Read and Print Current Positions

  READ {⟨*module*⟩}

  ```
  READ OMGS TTHM TEMP
  ```

- Drive Modules to Target Position

  SET {⟨*module*⟩ = ⟨*value or identifier*⟩}
  SETC {⟨*module*⟩ = ⟨*value or identifier*⟩} *continue / no waiting*

  ```
  SET OMGS=3.23 PHIS=-23.02
  SET H=1 K=2 L=1
  SET HF=ON, MATRIX=XY
  ```

- Data Acquisition

  MES [⟨*preset*⟩]

  ```
  MES
  MES 2500
  ```

  Note:

  > This command starts a single data acquisition without incrementing the run number and overwrites the temporary measurment file! For regular measurment files refer to (next) section `scans`.
  > A given preset is temporary used insted of the default defined by command `MM1`.

6.2 Scans

- Scan around a central position

  SGEN STEP = ⟨*steps*⟩{⟨*module*⟩ = ⟨*central_pos*⟩}{⟨*d_module*⟩ = ⟨*step_width*⟩}

  ```
  SGEN STEP = 11 OMGS=10.3 DOMGS=0.2 TTHS=20.6 DTTHS=0.4
  ```

```
SGEN STEP = 21 DOMGS=0.2
```
[1]

```
SGEN STEP=31 H=1 K=2 L=1 DH=0.1 DK=0 DL=0
```

- Scan from start to end position

  SGEN1 STEP = ⟨*steps*⟩{⟨*module*⟩ = ⟨*start_pos*⟩⟨*end_pos*⟩}

  ```
  SGEN1 STEP =21 OMGS=20.5, 31.8
  SGEN1 STEP=13 H=1, 1.2 K=2,2.2 L=3,3.2 E=0
  ```

- Definition of an arbitrary Scan
  using SCANB (scan begin) and SCANE (scan end)

  SCANB {⟨*action*⟩} *SCANE*

  ```
  SCANB
  SET OMGS=5.3
  SET MAGF=2.5
  MES
  ...
  SCANE
  ```

  Note:

  > It is not allowed to change (redefine) the `master module (MM1)` inside the scan.

Continuous scans There are two variants of continuous scans. Both at first set ⟨*motor*⟩ to the position ⟨*start*⟩ and subsequently continue to move ⟨*motor*⟩ to position ⟨*target*⟩. At the moment ⟨*motor*⟩ reaches position ⟨*start_mes*⟩ CARESS starts the measurment. The scan terminates if ⟨*motor*⟩ reaches position ⟨*target*⟩ or stops for any other reason.

- Continuous Scan with *fixed* number of steps

  SGEN2 ⟨*motor*⟩ = ⟨*start*⟩, ⟨*start_mes*⟩, ⟨*end_mes*⟩, ⟨*target*⟩*STEP* = ⟨*steps*⟩

  ```
  SGEN2 OMGX=11.7, 12.1, 42.8, 44.3 STEP=27
  ```

  Note:

  > There is no master counter involved (MM1 is to be ignored!) and all counters are used as slave modules. CARESS is doing ⟨*steps*⟩ measurment steps without stopping the movement nor the counters. The parameter *STEP* is defined to be the last parameter of the command.

- Continuous Scan with *variable* number of steps

  MM1 ⟨*master counter*⟩ = ⟨*preset*⟩
  SGEN2 ⟨*motor*⟩ = ⟨*start*⟩, ⟨*start_mes*⟩, ⟨*end_mes*⟩, ⟨*target*⟩

  ```
  MM1 TIM1=10000
  SGEN2 OMGX=11.7, 12.1, 42.8, 44.3
  ```

[1] scan around actual position of OMGS

Note:

> The master counter (MM1) is used to determine the measurment steps while all slave counters and the ⟨*motor*⟩ are continuously running.

6.3 Initialisation

- Initialise Hardware

  INITH

  ```
  INITH
  ```

  Note:

  > This command updates all CARESS internal information about the hardware configuration. Use `inith` to read in updated hardware description files.

- Reset / Delete Command Parameters
  with commands: CALIB, CSM, OFL, ONL, PASS, PERMOVE, PRO, PRW, PRWA, RESOSTEP, SCANV, SELDET, SL1, SPEC, TOF, WAIT1, WAIT2, WIND, WINDA

  RESET {⟨*commandkey*⟩}

  ```
  RESET PASS
  ```

7 Special actions

This section lists commands to deploy some special hardware setups/access.

7.1 Reference movements

- Move one or more motor axis, each to one reference point

  REF {⟨*module*⟩ = ⟨*direction*⟩}
  with direction=1 ≡ positive limit switch,
  direction=2 ≡ negative limit switch

  ```
  REF OMEGA=1, SLIT1_X=2
  ```

- Move one or more motor axis, each to one reference point and define this point as given position value

  REFSET {⟨*module*⟩ = ⟨*direction*⟩, ⟨*position*⟩}
  with direction=1 ≡ positive limit switch,
  direction=2 ≡ negative limit switch,
  position=new position value after movement

  ```
  REFSET OMGS=1,27
  REFSET OMGS=2,127 TTHS=1,-21.7
  ```

- Definition of a permanent 'Movement' of a magnetic field

  PMOVE ⟨*module*⟩ = ⟨*ccu*⟩⟨*ccd*⟩⟨*wcu*⟩⟨*wcd*⟩⟨*rcu*⟩⟨*rcd*⟩

  with: ccu/ccd = cycle current up/down
  wcu/wcd = wait current up/down
  rcu/rcd = rate current up/down

  ```
  PMOVE MAGF=0.5 1 0.7 1.2 2 2
  ```

  Note:

  > This command currently works for magnetic fields controlled by a bruker bec-1 power supply (CARESS device kind 68).

7.2 Universal device access

- receive an ASCII string from a gpib (iec bus) device

READGPIB ⟨X⟩ : ⟨Y⟩

with X = logical unit number of the gpib controller and
Y = address of the gpib device

or

READGPIB ⟨Z⟩

with Z = logical unit number of the gpib device.

```
READGPIB 1:30
READGPIB 27
```

- send an ASCII string to a gpib (iec bus) device

  WRITEGPIB ⟨X⟩ : ⟨Y⟩ ′string′

  with X = logical unit number of the gpib controller and
  Y = address of the gpib device

  or

  WRITEGPIB ⟨Z⟩ ′string′

  with Z = logical unit number of the gpib device.

  ```
  WRITEGPIB 1:19 "MEASure:VOLTage:DC? 20, 0.001"
  WRITEGPIB 27 "SETup:TXPower:COUNt:STATe ON"
  ```

- send an ASCII string to a device/module

  LOADTEXT ⟨module⟩ = ⟨string⟩
  ⟨string⟩ ::= ⟨identifier⟩|′⟨text⟩′ (no ’ in text)

  ```
  LOADTEXT ADET=‘LISTMODE ON’
  ```

  Note:

  > To interpret the sent information is exclusively up to the device (driver).

- send the contents of a file to a device/module

  LOADFILE ⟨module⟩ = ⟨filename⟩
  ⟨string⟩ ::= ⟨identifier⟩|′⟨filename⟩′

  ```
  LOADFILE ADET=‘adet-special.cfg’
  ```

- Set an input register bit to either logical 0 or 1

  SETBIT ⟨bit − number⟩⟨value⟩

  ```
  SETBIT 2 1
  ```

  Note:

  > This command is used at a special wired register at V4. (cf. WAIT4BIT at page 24)

7.3 Complex actions

- Sophisticated command at instrument `M2`

  SETCM2 ⟨device⟩⟨motoraxis⟩⟨setpoint⟩⟨rangelow⟩⟨rangehigh⟩⟨mode⟩⟨interval⟩

  ```
  SETCM2 COU, TTHS, 123, 5, 124, 0, 1.7
  ```

  Note:

  > This command initiates a very special algorithm at instrument `M2` by sending several data to the named `device`. It starts a measurment within `range` while moving from current to `set point` position of `motor axis`. `mode` and `interval` are additional measurment parameters.
  > This command is blocking (see `ENDCM2` at on this page)

- Terminate the `SETCM2` command

  ENDCM2

  ```
  ENDCM2
  ```

  Note:

  > Since `SETCM2` is a blocking command it is necessary to terminate it explicitly with `ENDCM2`.

8 Control special hardware devices

A set of commands to deal with more sophisticated hardware systems is shown in the following paragraphs.

8.1 Definition of Compound Positions, etc.

- Sample Changer Positions (V4)

  DEFPOS {⟨*pos_no*⟩⟨*pos_x*⟩⟨*pos_y*⟩}

  ```
  DEFPOS 1 2.3 0, 2 0.5 0, 3 0.52 0.1, ...
  ```

- Define List Position

  DEFLPOS LPOS = ⟨*pos_no*⟩{⟨*module*⟩ = ⟨*value*⟩}

  ```
  DEFLPOS LPOS=1 H1=2.0 H3=1.5 H10=-0.5 ...
  ```

- Change (defined) List Position

  CHGLPOS LPOS = ⟨*pos_no*⟩{⟨*module*⟩ = ⟨*value*⟩}

  ```
  CHGLPOS LPOS=1 H3=1.6
  ```

- Define and name special Positions (E2, E7, V4, M1, M2) or Distances (V6)

  DEFINE {⟨*module*⟩_⟨*name*⟩⟨*value*⟩}

  ```
  DEFINE BSX_OUTX 53.1 BSY_OUTY 30.95
  ```

8.2 Define Data Acquisition Unit DAU

The DAU is a special hardware developed at Hahn-Meitner-Institut Berlin and used at instruments V5/SPAN only.

- Enable detectors and define data width (16 bit or 32 bit) for incrementing
  SI: single detector banks and/or
  MU: area detector,
  I2: 16 bit (2 bytes), I4: 32 bit (4 bytes)

  CONFIG {⟨*detector*⟩ = ⟨*width*⟩}

  ```
  CONFIG SI=I4 MU=I2
  CONFIG SI=I2
  ```

- Define Time-Of-Flight Parameters (note the DAU has 384KB memory for detectory data)
  TB: time base (0. . . 4: 0=125ns/8MHz, 1=250ns, 2=0.5μs, 3=1μs, 4=2μs),
  DT: delay time (units of TB for delay between pick-up signal and start of time window),
  CN: number of channels (1. . . 4096),
  CW: channel width in units of TB (⟨⟩ =1)

  TOF TB = ⟨*value*⟩*DT* = ⟨*value*⟩*CN* = ⟨*value*⟩*CW* = ⟨*value*⟩

  ```
  TOF TB=0 DT=20 CN=1024 CW=10
  ```

  defines TOF mode and TOF parameters
  time base 125ns (8MHz), delay time 2.5ms, 1024 channels of 1.25ms

- Switch from TOF Mode to NSE Mode

  RESET TOF

  ```
  RESET TOF
  ```

8.3 Neutron Chopper System

- This command is used to set up several parameters of a chopper system. The real activation of these parameters is done with 'SET CHOP = RUN'
  Define State
  0: synch,
  1: asynch,
  Speed (RPM),
  Speed Ratio and
  Phase in relation to Chopper 1

  CHOPPER STATE = {⟨*chop_no*⟩ ⟨*state*⟩} *SPEED* = {⟨*chop_no*⟩ ⟨*speed*⟩}
  RATIO = {⟨*chop_no*⟩ ⟨*ratio*⟩} *PHASE* = {⟨*chop_no*⟩ ⟨*phase*⟩}

  ```
  CHOPPER STATE=2 0, 3 0, 4 0, SPEED=1 8000 -
  RATIO=2 4 PHASE= 2 0, 3 0, 4 0
  ```

9 Flow control

A group of commands to control the actual command execution is to find here.

9.1 Wait commands

- Wait for seconds (unconditional)
  WAIT $\langle seconds \rangle$

  ```
  WAIT 10
  ```

- Wait before start of a scan
  WAIT1 $\langle seconds \rangle$

  ```
  WAIT1 20
  ```

- Wait between accomplished drive and start of acquisition in a scan (repeatedly)
  WAIT2 $\langle seconds \rangle$

  ```
  WAIT2 15
  ```

- Wait for an external signal (hardware, i/o-module trigger)
  WAITTRIG

  ```
  WAITTRIG
  ```

- Wait for either logical 0 or 1 at an input register bit
  WAIT4BIT $\langle bit - number \rangle \langle value \rangle$

  ```
  WAIT4BIT 2 1
  ```

  Note:

  > This command is used at a special wired register at V4. (cf. SETBIT at page 20)

- Wait until logical expression becomes true
  WAIT4EXPR $\langle expression \rangle$

  ```
  WAIT4EXPR ABS(TEMP-T_SAM)<=0.25
  ```

9.2 Input Utilities

Used for Variables, Procedures, Loops, Dummy Command Parameters

- Definition of Variables, Arithmetic Operations (+, -, *, /), Mathematical Functions (sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, exp, log, log10, ceil, floor, sqrt, fabs) and Assignment

  LET ⟨variable⟩ = ⟨expression⟩

  ```
  LET A = 5.4
  LET START = -1
  LET WIDTH = (2+OMGS)*SQRT(A) + START
  ```

- Definition of Procedure

  PROCEDURE ⟨proc_name⟩(⟨formal_parameter_list⟩)
  sequence of CARESS commands
  ENDP

  ```
  PROCEDURE MYPROC(STEPS,CENTRE,STEP_WIDTH)
  SGEN STEP=STEPS OMGS=CENTRE DOMGS=STEP_WIDTH
  ...
  ENDP
  ```

- Call of Procedure

  DO ⟨proc_name⟩(⟨actual_paramter_list⟩)

  ```
  DO MYPROC(11,5,0.1)
  ```

- List currently defined Assignments or Procedures

  LIST ⟨command⟩ (LET or PROCEDURE)

  ```
  LIST LET
  ```

- Loops (loops may be nested)

  REPEAT ⟨repeat count⟩
  {any sequence of commands}
  ENDR

  ```
  LET I = 0
  LET H1_START = 1.5
  LET H1_DELTA = 0.2
  REPEAT 10
  LET H1_POS = H1_START + I*H1_DELTA
  SET H1 = H1_POS
  MES
  LET I = I+1
  ENDR
  ```

- Open a Command Parameter File

  PARF ⟨filename⟩ !(max.20 characters, CARESS appends extension .par)

  ```
  PARF myhkl
  ```

- Replace dummy Command Parameters ($) by Values from the Command Parameter File myhkl.par

  SET commandparameters

  ```
  SET H=$ K=$ L=$
  SET FL=$
  ```

- Exit active Loop, if end-of-record (EOR) or end-of-file EOF

  IF ⟨flag⟩ EXIT

  ```
  REPEAT 10
  SET H=$ K=$ L=$
  MES
  IF EOR EXIT
  IF EOF EXIT
  ENDR
  ```

9.3 Control CARESS by an external user program, ...

- Call External User Program

  EXTERN ⟨text⟩

  ```
  EXTERN myprog
  EXTERN center
  ```

- Call External PV-WAVE User Program

  EXTERN WAVE ⟨text⟩

  start PV-WAVE program myprog.pro:
  ```
  EXTERN WAVE myprog
  ```
  start PV-WAVE (and type in PV-Wave commands):
  ```
  EXTERN WAVE
  ```

- Return from external control to normal CARESS control

  INTERN

  ```
  INTERN
  ```

  Note:

  > INTERN is not allowed in command files, procedures, and repeat sequences.
  > EXTERN is ignored, if CHECK is active.

EXTERN will interrupt an active command file.
If EXTERN becomes active in a procedure or repeat sequence out of a command file, no command file may be opened by the external process.
An external process may not define/use procedures nor repeat sequences (it will be more convenient to use the facilities of the programming language!).

- Set a semaphore and abandon current control

  BLOCK ⟨GID⟩ (with $0 \leq GID \leq 9$*)*

  ```
  BLOCK 1
  ```

  Note:

  This command is used for synchronisation reasons in a multi-client setup. `BLOCK` abandons the current control and defines a semaphore group. The **only** way to get control back is to use `UNBLOCK` (description on the current page) from another CARESS client.

- Clear semaphore in multi-client environments

  UNBLOCK [GID]

  ```
  UNBLOCK
  UNBLOCK 3
  ```

  Note:

  This command is used for synchronisation reasons in a multi-client setup. `UNBLOCK` clears all semaphores, with an optional group id only this semaphore group is cleared. Semaphores are defined with `BLOCK` (description on this page).

9.4 Interrupt / Kill / Continue / Terminate CARESS Activities

- kill current activit

  KILS

  ```
  KILS
  ```

- delete data file (after KILS, only)

  KILD

  ```
  KILD
  ```

- cancel command input queue including all open command file

  KILC

  ```
  KILC
  ```

- cancel active repeat loops and/or procedure

  KILP

  ```
  KILP
  ```

- continue normal processin

  CON

  ```
  CON
  ```

- abort a cancel action (KILS, KILC, KILP, INTERN and CON) (command version, only)

  ABO ⟨killcommand⟩

  ```
  ABO KILP
  ```

- abort a cancel action (KILS, KILD, KILC, KILP, INTERN and CON)

  ABOD ⟨killcommand⟩

  ```
  ABOD KILD
  ```

- terminate all CARESS activities (not allowed in command files, procedures, and repeat sequences)

  EX or EXIT

  ```
  EXIT
  ```

  Note:

  > To interrupt an activity click the HALT button within the GUI or type CNTRL/h within the command version of CARESS.

10 Miscellaneous commands

Some Commands do not fit into categories before and will be listed here.

- Check commands

  CHECK ON/OFF [options]

  ```
  CHECK ON
  @mycommands
  CHECK OFF
  ```

  Options to restrict check depth are:

  > $MAXREPEAT = \langle n \rangle$ limit repetition of loops to $\langle n \rangle$
  >
  > rangle (aliases to $MAXREPEATS$ and $MAXLOOP$)
  > $MAXFILE = \langle n \rangle$
  > rangle limit nesting script level to $\langle n \rangle$
  >
  > rangle (aliases to $MAXFILES$ and $NESTED$)
  > $NEWONLY = ON/OFF$
  > rangle if *ON*, include script files only once
  >
  > rangle (aliases to $NEWFILEONLY$, $NEWFILESONLY$
  >
  > rangle and $UNIQUE$)

  Note:

  > The enclosed commands are only checked, the parameters are not stored! Procedures are being defined, independent of CHECK! The enclosed commands and CHECK OFF are not written into the CARESS session protocol. CHECK ON/OFF is not allowed in command files, procedures, and repeat sequences. Typing CHECK OFF within procedure or repeat definition and CHECK active will cancel all open procedure/repeats. A warning message is printed. By this mechanism errors in command files (e.g. missing ENDR) may be detected, if checked by the GUI. Caution! Don't forget to edit the faulty command file before executing it!

- Switch OFF/ON Archiving of Data on central archive system

  ACV OFF/ON

  ```
  ACV OFF
  ```

- Bypass Condition(s) which prevents Data Acquisition e.g. SHUTTER closed (V4)

  BYPASS ON|OFF

  ```
  BYPASS ON
  ```

- Disable *keep position* of a Motor Control or all Motors
  i.e. allow persistent manual positioning (Y66 motor control)

  MANUAL [⟨*module*⟩]

  ```
  MANUAL
  MANUAL OMGS
  ```

- Set Modules "offline"

  OFL {⟨*module*⟩}

  ```
  OFL TTHA
  ```

- Disable/enable Protected Input
  (CARESS command version only, not for GUI)

  PASSW ON/OFF

  ```
  PASSW ON
  ```

- List the current Occurrence of a Command

  SHOW ⟨*command key*⟩

  ```
  SHOW BLS
  ```

Index

User’s Notes