Amor

Apparatus for Multi Option Reflectometry

Amor commands glossary

aby	**aby** device analyzer electromagnet, **a**nalyzer with magnetic field **b** with field direction **y**, i.e. perpendicular to the direction of the neutron beam **dr aby var** drives field at analyzer to value var
address	**adress** user address.
alias	
amorhmconf	
amorpar	
amorset	
amorstatus	
anadist	
analyzerstate	
ananame	
aom	**aom** **a**ngle **om**ega of analyzer mirror **dr aom var** drives analyzer mirror angle to value var
aoz	**aoz** translation of **a**nalyzer mirror along **z** **dr aoz var** drives analyzer mirror height to value var
ath	
atz	**atz** translation of **a**nalyzer stage along **z** **dr atz var** drives analyzer stage height to value var
aw	**aw** read the current value of the acceptance window. **aw val** set the acceptance window to val.
backup	**backup file** saves the current values of SICS variables and selected motor and device parameters to the disk file specified as parameter. If no file parameter is given the data is written to the system default status backup file. The format of the file is a list of SICS commands to set all these parameters again. The file is written on the instrument computer relative to the path of the SICS server. This is usually /home/INSTRUMENT/bin. **backup motSave** toggles a• which controls saving of motor positions. If this• is set, commands for driving motors to the current positions are included in the backup file. This is useful for instruments with slipping motors.
batchroot	**batchroot arg** defines the root directory of which jobfiles can be started, e.g. batchroot /home/amor/job

batchrun	**batchrun arg** starts a jobfile when batchroot has been defined batchrun test.job
broadcast	
c3z	**c3z** motor translation of single detector (counter) in z-direction **dr c3z var** drives single detector height to value var
center	**center** the command center drives the first scan variable to the peak center of the last scan.
cfgenv	
ch1ph	**ch1ph** returns value of chopper 1 phase **dr ch1ph var** drives chopper 1 phase to value var
ch2ph	**ch2ph** returns value of chopper 2 phase **dr ch2ph var** drives chopper 2 phase to value var
choco	
chopper1phase	**ch1ph** returns value of chopper 1 phase **dr chopper1phase var** drives chopper 1 phase to value var
chopper2phase	**chopper2phase** returns value of chopper 2 phase **dr chopper2phase var** drives chopper 2 phase to value var
chopperrotation	**chopperrotation** displays chopper rotation speed.
chopperspeed	
chosta	**chosta** displays chopper status
chsp	**chsp** displays chopper speed
clientput	**clientput sometext1 ...** writes everything after clientput to the client which started the script. This is needed as SICS supresses the output from intermediate commands in scripts. Except error messages and warnings. With clientput this scheme can be circumvented and data be printed from within scripts.
com	**com** motor detector (counter) angle omega **dr com var** drives detector angle omega to value var
commandlog	**commandlog** displays the status of the commandlog. **commandlog new filename** starts a new commandlog writing to filename. Any prior files will be closed. The log file can be found in the directory specified by the ServerOption LogFileDir. Usually this is the log directory. **commandlog close** closes the commandlog file. **commandlog auto** switches automatic log file creation on. This is normally switched on. Log

		switches automatic log file creation on. This is normally switched on. Log files are written to the log directory of the instrument account. There are time stamps any hour in that file and there is a new file any 24 hours. **commandlog tail n** prints the last n entries made into the command log. n is optional and defaults to 20. Up to 1000 lines are held in an internal buffer for this command.
	con1	
	con2	
	con3	
	con4	
	con5	
	con6	
	conf	
	config	**config** command configures various aspects of the current client server connection. Basically three things can be manipulated: The connections output class, the user rights associated with it, and output files. **config OutCode val** sets the output code for the connection. By default all output is sent to the client. But a graphical user interface client might want to restrict message to only those delivering requested values and error messages and suppressing anything else. In order to achieve this, this command is provided. Possible values Values for val are Internal, Command, HWError,InError,Status, Error, Value. This list is hierarchical. For example specifying InError for val lets the client receive all messages tagged InError, Status, Error and Value, but not HWError, Command and Internal messages. **config Rights Username Password** Each connection between a client and the SICS server has user rights assocociated with it. These user rights can be configured at runtime with the command config Rights Username Password . If a matching entry can be found in the servers password database new rights will be set. **config File name** Scientists are not content with having output on the screen. In order to check results a log of all output may be required. The command config File name makes all output to the client to be written to the file specified by name as well. The file must be a file accessible to the server, i.e. reside on the same machine as the server. Up to 10 logfiles can be specified. Note, that a directly connected line printer is only a special filename in unix. **config close num** closes the log file denoted by num again. **config list** lists the currently active values for outcode and user rights.
	connect_sea	
	continue	
	count	**count mode preset** does a count operation in mode with a preset of preset. The parameters are optional. If they are not given the count will be started with the current setting in the histogram memory object. After the count, StoreData will be automatically called.
	counter	
	cox	**cox** motor detector (counter)translation along x-axis **dr cox var** drives motor detector translation along x-axis to value var
		coz

coz	motor detector (counter)translation along z-axis (height) **dr coz var** drives motor detector translation along z-axis to value var
cscan	**cscan var center delta np preset** center scan is a convenience command which starts a scan around a specified center value. This mostly used for centering purposes. All parameters must be specified. The parameters and their meanings: • · var is the variable which is to be center scanned. Only one can be specified. • · center is the value to use as center of the scan. • · delta is the step width to use for the scan. • · np is the number of points to scan in each direction. • · preset is the preset to use for the counter. As the counter mode, the mode currently configured active in the scan object is used.
d1b	**d1b** motor diaphragm (slit) 1, gives position translation of slit 1 in z-direction **dr d1b val** drives position in z of slit 1 to value var
d1dist	
d1l	**d1l** motor diaphragm (slit) 1 width left, gives the position of left border of width of slit 1 **dr d1l val** drives left border of width of slit 1 to value var
d1r	**d1r** motor diaphragm (slit) 1 width right, gives the position of right border of width of slit 1 **dr d1r val** drives right border of width of slit 1 to value var
d1t	**d1t** motor diaphragm (slit) 1, gives the opening height of slit 1 **dr d1t val** drives opening of slit 1 to value var
d2b	**d2b** motor diaphragm (slit) 2, gives position translation of slit 2 in z-direction **dr d2b val** drives position in z of slit 2 to value var
d2dist	
d2l	**d2l** motor diaphragm (slit) 2 width left, gives the position of left border of width of slit 2 **dr d2l val** drives left border of width of slit 2 to value var
d2r	**d2r** motor diaphragm (slit) 2 width right, gives the position of right border of width of slit 2 **dr d2r val** drives right border of width of slit 2 to value var
d2t	**d2t** motor diaphragm (slit) 2, gives the opening height of slit 2 **dr d2t val** drives opening of slit 2 to value var
d3b	**d3b** motor diaphragm (slit) 3, gives position translation of slit 3 in z-direction **dr d3b val** drives position in z of slit 3 to value var
d3dist	

d3l	**d3l** motor diaphragm (slit) 3 width left, gives the position of left border of width of slit 3 **dr d3l val** drives left border of width of slit 3 to value var
d3r	**d3r** motor diaphragm (slit) 3 width right, gives the position of right border of width of slit 3 **dr d3r val** drives right border of width of slit 3 to value var
d3t	**d3t** motor diaphragm (slit) 3, gives the opening height of slit 3 **dr d3t val** drives opening of slit 3 to value var
d4b	**d4b** motor diaphragm (slit) 4, gives position translation of slit 4 in z-direction **dr d4b val** drives position in z of slit 4 to value var
d4dist	
d4l	**d4l** motor diaphragm (slit) 4 width left, gives the position of left border of width of slit 4 **dr d4l val** drives left border of width of slit 4 to value var
d4r	**d4r** motor diaphragm (slit) 4 width right, gives the position of right border of width of slit 4 **dr d4r val** drives right border of width of slit 4 to value var
d4t	**d4t** motor diaphragm (slit) 4, gives the opening height of slit 4 **dr d4t val** drives opening of slit 4 to value var
d5b	**d5b** motor diaphragm (slit) 5, gives position translation of slit 5 in z-direction **dr d5b val** drives position in z of slit 5 to value var
d5dist	
d5l	**d5l** motor diaphragm (slit) 5 width left, gives the position of left border of width of slit 5 **dr d5l val** drives left border of width of slit 5 to value var
d5r	**d5r** motor diaphragm (slit) 5 width right, gives the position of right border of width of slit 5 **dr d5r val** drives right border of width of slit 5 to value var
d5t	**d5t** motor diaphragm (slit) 5, gives the opening height of slit 5 **dr d5t val** drives opening of slit 5 to value var
db	
dbs	**dbs** motor diaphragm (slit) after shielding **dr dbs val** drives position in z of diaphragm (slit) at shielding to value var

definealias	
detectordist	
devexec	
dimetix	
dir	**dir** a single word command which lists all objects available in the SICS system in its current configuration.
dolater	
dr	**dr var newval var newval** **drive var newval var newval** can be called with one to n pairs of object new value pairs. This command will set the variables in motion and wait until the driving has finished. A drive can be seen as a sequence of a run commands immediately followed by a success command.
drivavable	
drive	**drive var newval var newval** **dr var newval var newval** can be called with one to n pairs of object new value pairs. This command will set the variables in motion and wait until the driving has finished. A drive can be seen as a sequence of a run commands immediately followed by a success command.
driverlist	
email	**email** user e-mail address.
emon	**emon** the environment monitor emon takes for the monitoring of an environment device during measurements. It also initiates error handling when appropriate. The emon understands a couple of commands. **emon list** this command lists all environment devices currently registered in the system. **emon register name** this is a specialist command which registers the environment device name with the environment monitor. Usually this will automatically be taken care of by evfactory. **emon unregister name** this is a specialist command which unregisters the environment device name with the environment monitor. Usually this will automatically be taken care of by evfactory. Following this call the device will no longer be monitored and out of tolerance errors on that device no longer be handled.
evfactory	**evfactory new name type par par ...** evfactory is responsible for configuring and deconfiguring sample environment devices into SICS. The syntax creates a new sample environment device. It will be known to SICS by the name specified as second parameter. The type parameter decides which driver to use for this device. The type will be followed by additional parameters which will be evaluated by the driver requested. **evfactory del name** deletes the environment device name from the system.
exe	
fax	**fax** user fax number
fileeval	**fileeval name** tries to open the file name and executes the script in this file. Then there are some special commands which can be used within macro-sripts.

fmaset	**fmaset** initalizes 1 T electro magnet **fmaset on** set electro magnet on **fmaset off** set electro magnet off
fom	**fom** angle omega of frame overlap mirror **dr fom var** drives angle omega of frame overlap mirror to value var
fomdist	
fomname	
ftz	**ftz** frame overlap mirror translation along z-axis **dr ftz var** drives frame overlap mirror along translation z-axis to value var
fulltransact	
getconfdialog	
getint	
getlog	**getlog** The SICS server logs all its activities to a logfile, regardless of what the user requested. This logfile is mainly intended to help in server debugging. However, clients may register an interest in certain server events and can have them displayed. This facility is accessed via the GetLog command. It needs to be stressed that this log receives messages from all active clients. **getlog all** achieves that all output to the server logfile is also written to the client which issued this command. **getlog kill** stops all logging output to the client. **getlog outcode** request that only certain events will be logged to the client issuing this command. Enables only the level specified. Multiple calls are possible.
getmode	
graph	
gumput	
help	
hm	**hm genbin start step nBins** generate the time binning in SICS. This generates an equidistant time binning starting at time start, with stepwidth step and nBins time bins. **hm init** configure the histogram memory to use the new time binning. **hm timebin** prints the currently active time binning array. **hm setbin inum value** sometimes unequally spaced time binnings are needed. These can be configured with this command. The time bin iNum is set to the value value. **hm clearbin** deletes the currently active time binning information.
hmc	
hsy	**hsy** field of electro magnet **dr hsy var** drives field of electro magnet to value var
installhdb	

installdb	
instrument	
interneval	
killfile	
lastscancommand	
list	**list** listing of motorparameters **list name** listing of motor parameters of motor name
listexe	
m2t	**m2t** drives polarizing mirror to angle 2theta. This command moves all optcal devices along the reflected beam path of the neutrons after the polarizing mirror into reflected beam position. **dr m2t var** drives polarizing mirror angle to value var
makemcstascontroller	
makemulti	
makeobj	
makesicspol	
makeswhpmotor	
mdif	
mom	
motavable	
motb	
motc	
moz	
mty	
mtz	
nxscript	
o2t	
opti	
p1	
pause	**pause** pauses the measurement until the problem has been resolved.
pby	
peak	**peak** the command peak prints the position, FWHM and maximum value of the peak in the last scan.
performance	
phone	**phone** user phone number

polarizerstate	
poldist	
polname	
psdconfigure	**psdconfigure hm xsize ysize** The resolution of the PSD in pixels can be tailored to the experiment at hand. To this purpose the command psdconfigure is available. xsize and ysize are the resolution of the detector in x direction (beam width) and y direction (two theta).
publish	
remob	
removeobject	
repeat	**repeat num mode preset** calls count num times. num is a required parameter. The other two are optional and are handled as described above for count.
resetserver	**resetserver** resets the server after an interrupt.
restore	**restore file** reads a file produced by the backup command described above and restores SICS to the state it was in when the status was saved with backup. If no file argument is given the system default file gets read.
run	**run var newval var newval …** can be called with one to n pairs of object new value pairs. This command will set the variables in motion and return to the command prompt without waiting for the requested operations to finish. This feature allows to operate other devices of the instrument while perhaps a slow device is still running into position.
s2t	
samenv	
sample	**sample** sample name.
sample_mur	
sampledist	
scan	**scan clear** clears current scan parameters. **scan list** lists current scan parameters. **scan var name start step** defines a variable (motor) to be scanned. The name of the variable, a start value and a stpe width need to be given. More then one scan variable can be specified. **scan modvar name start step** modifies the scan parameters for scan variable name to the new values given. **scan getvars** returns a list of currently active scan variables terminated with the string -END-. **scan np num** sets the number of scan points. **scan preset val** sets the preset value for the scan. Without a parameter, inquires the current value. **scan mode val** sets the count mode for the scan. Without a parameter, inquires the current value. Possible values are timer or monitor. **scan run** executes the scan.

	executes the scan. **scan cinterest** this call enables automatic printing of scan counts to your connection when new values arise. This command is primarily of interest for status display clients. **scan pinterest** this function makes the scan command send a notification (the string ScanVarChange) to you whenever the scan variables get modified. This command is primarily of interest for status display clients.
scancounts	
scaninfo	
scanmode	
sch	
scriptcallback	
sea	
serialport	
setint	**setint newval** GetInt sets SICS interrupts from macro scripts. Not recommended! Possible return values or new values are: continue, abortop, abortscan, abortbatch, halt, free, end. This command is only permitted in macros. Should only be used by SICS programmers.
setstatus	**setstatus newval** sets the SICS status to one of: Eager, UserWait, Count, NoBeam, Driving, Running, Scanning, Batch Hatl or Dead. This command is only available in macros.
sftime	
shutter	**shutter** The command shutter without arguments returns the status of the shutter. This can be one of open, closed, Enclosure is broken. **shutter open** opens the shutter when possible. **shutter close** closes the shutter.
sics_exitus	**sics_exitus** single word commands which shuts the server down. Only managers may use this command.
sicsbound	**sicsbound var newval** checks if the new value newval lies within the limits for variable var. Returns an error or OK depending on the result of the test.
sicscron	
sicsdatafactory	
sicsdatanumber	
sicsdatapath	
sicsdatapostfix	
sicsdataprefix	
sicsdescriptor	
sicsidle	
sicsprompt	
	sicsstatus var

sicsstatus		SICS devices such as counters or motor may be started and left running while the program is free to do something else. This command inquires the status of such a running device. Return values are internal SICS integer codes. This command is only of use for SICS programmers.
sicstime		
sicstype		**sicstype** object allows to query the type of the object specified by object. Possible return values are • · DRIV if the object is a SICS drivable object such as a motor • · COUNT if the object is some form of a counter. • · COM if the object is a SICS command. • · NUM if the object is a number. • · TEXT if object is something meaningless to SICS.
som		
soz		
sp		
sps1		
sscan		**sscan var1 start end var2 start end ... np preset** simple scan is a convenience command which starts a scan for one to several variables with a simplified syntax. All parameters must be specified. The parameters and their meanings: • · var1 start end This is how the variables to scan are specified. For each variable scanned the name of the variable, the start value and the end value of the scan must be given. More then one triplet can be given in order to allow for several scan variables. • · np is the number of points to scan. • · preset is the preset to use for the counter. As the counter mode, the mode currently configured active in the scan object is used.
starttime		
statistics		
status		**status** a single word command which makes SICS print its current status. Possible return values can be: Eager to execute commands, Scanning, Counting, Running, Halted. Note that if a command is executing which takes some time to complete the server will return an ERROR: Busy message when further commands are issued. **status interest** initiates automatic printing of any status change in the server. This command is primarily of interest for status display client implementors
stb		
stdscan		
sth		
stopexe		
storeamor		
storeamornew		
stt		
stz		
success		**success** waits and blocks the command connection until all pending operations

	have finished (or an interrupt occured).
t1	
t2	
td	**td** read the current value of the time to delay to start. **td val** set the time delay to start to val.
tecs	
temperature	
textstatus	
title	**title** measurement title
tofmode	
transact	
tt	
udpquieck	
update_remob	
user	**user** user name
wait	**wait time** waits time seconds before the next command is executed. This does not stop other clients from issuing commands.
wwwdata	
wwwgetaxis	
wwwgetdata	
wwwsics	
xxxscan	