

SOCKET INTERFACE SPECIFICATION V4.1.0

MYTHEN2 DETECTOR SYSTEM

Document Version: V4

Document Version V8

© Copyright 2020

DECTRIS Ltd

Contents

1. General Information	6
1.1. Contact and Support	6
1.2. Explanation of Symbols	6
1.3. Explanation of Terms	7
1.4. Warranty Information	7
1.5. Disclaimer	7
2. Safety Instructions	8
3. Introduction	9
4. Operation	10
4.1. Connecting to the Server	10
4.3. Settings	10
4.3.1. Predefined Settings	10
4.1.2. Setting Threshold and Energy	10
4.4. Frame Rates	11
4.5. Acquisition Control	11
4.7. Triggering and Gating	12
4.8. Debugging	12
5. Example Applications	13
6. Commands	14
6.1. Command List	14
6.1.1. General Commands	15
6.1.2. Acquisition Control	17
6.1.3. Detector Settings	19
6.1.4. Data Correction	20
6.1.5. Trigger/Gate	16
6.1.6. Debugging	24
7. Error Codes	25
8. Release Notes	27
8.1. Version 4.1.0	26
8.2. Version 4.0.1	27
8.3. Version 4.0.0	27
8.4. Version 4.0.0-T03	27
8.4.1. Modifications Compared to 4.0.0-T02	27
8.5. Version 4.0.0-T02	27
8.5.1. Modifications Compared to 4.0.0-T01	27
8.6. Version 4.0.0-T01	27
8.6.1. Modifications	28
8.6.2. New Commands	28
8.6.3. New Error Codes	29

8.6.4. Removed Commands	29
8.7. Version 3.0.0	29
8.7.1. Modifications	29
8.7.2. Migration Notes	30
8.7.3. Legacy2 Mode	30
8.7.4. Known Issues	30
8.8. Version 2.1.0	30
8.9. Version 2.0.5	31
8.9.1. Known Issues	31
8.10. Version 2.0.4	31
8.11. Version 2.0.3	32
8.12. Version 2.0.2	32
8.13. Version 2.0.1	32
8.14. Version 2.0.0	32
8.14.1. New Functionalities	32
8.14.2. Migration Notes	32
8.14.3. Legacy Mode	33

Document History

Current Document

Version	Date	Status	Prepared	Checked	Released
4	24.04.2020	released	LG	RM, LW	

Changes

Version	Date	Changes
3	05.06.2019	Updated Corporate Design
2	22.12.2017	Updated Web Links
1	05.07.2016	First Version

1. General Information

1.1. Contact and Support



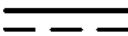



Address: DECTRIS Ltd.
 Taefernweg 1
 5405 Baden-Daettwil
 Switzerland

Phone: +41 56 500 21 02
Fax: +41 56 500 21 01

Homepage: <http://www.dectris.com/>
Email: support@dectris.com

Should you have questions concerning the system or its use, please contact us via telephone, e-mail or fax.

1.2. Explanation of Symbols

Symbol	Description
	Important or helpful notice
	Caution. Please follow the instructions carefully to prevent equipment damage or personal injury.
	DC-current
	AC-current
	Ground
	Functional earth

1.3. Explanation of Terms

Term	Description
ASIC	Application-Specific Integrated Circuit
DAC	Digital Analog Converter
GUI	Graphical User Interface
DCS4	Detector Control System for up to 4 detector modules
Detector Module	The smallest fully functional unit of the detector (1280 or 640 channels)
P2P	Peer to Peer connection (fixed IP address)
MCB	Module Control Board
FIFO	Storage working according to First In – First Out
keV	Kilo electron Volt

1.4. Warranty Information

Should your detector require warranty service, contact DECTRIS for further information. Before shipping the system back, please contact DECTRIS to receive the necessary transport and shipping information. Make sure that the original packaging is used when returning the system.



Do not ship the system back before you receive the necessary transport and shipping information.

When returning the detector system for repair, be sure to fill out and include the service form at the back of this document to provide the support division with the necessary information.

1.5. Disclaimer

DECTRIS has carefully compiled the contents of this manual according to the current state of knowledge. Damage and warranty claims arising from missing or incorrect data are excluded.

DECTRIS bears no responsibility or liability for damage of any kind, also for indirect or consequential damage resulting from the use of this system.

DECTRIS is the sole owner of all user rights related to the contents of the manual (in particular information, images or materials), unless otherwise indicated. Without the written permission of DECTRIS it is prohibited to integrate the protected contents in this publication into other programs or other websites or to use them by any other means.

DECTRIS reserves the right, at its own discretion and without liability or prior notice, to modify and/or discontinue this publication in whole or in part at any time, and is not obliged to update the contents of the manual.

2. Safety Instructions



Please read these safety instructions before operating the detector system.

- Before turning the power supply on, check the supply voltage against the label on the power supply. Using an improper main voltage will destroy the power supply and damage the detector.
- Use only the power supply delivered by DECTRIS
- Power down the detector system before connecting or disconnecting any cable.
- Make sure the cables are connected and properly secured.
- Do not use any other cables than delivered by DECTRIS
- Avoid pressure or tension on the cables.
- The detector system should have enough space for proper ventilation. Operating the detector outside the specified ambient conditions could damage the system.
- The detector is not specified to withstand direct beam at a synchrotron. Such exposure will damage the exposed pixels.
- Place the protective cover on the detector when it is not in use to prevent the detector from accidental damage.
- Opening the detector or the power supply housing without explicit instructions from DECTRIS will void the warranty.
- Do not install additional software or change the operating system.
- Do not touch the entrance window of the detector.
- Do not connect any device to the USB plugs of the DCS4 except the detector modules.
- Do not connect the detector module to any other USB device than the DCS4.

3. Introduction

The Detector Control System (DCS4) hosts an embedded Linux system, on which the MYTHEN Socket Server is running. A client can open a socket connection to the server and control the MYTHEN2 detector by sending the commands listed in Chapter 6. Example applications in C++ for Linux and Windows are provided with the system documentation. A *spec* implementation based on the Socket Interface is available from <http://www.certif.com> for some versions of the Socket Server.

To take advantage of new functionality and bug fixes, it is possible to update to a newer version of the MYTHEN Socket Server within the same major release. The required files can be downloaded from the DECTRIS Ltd. website <http://www.dectris.com>.

An upgrade to a new major release usually requires new hardware or a recalibration of the detector by DECTRIS Ltd. and is therefore not possible to be performed by the customer. If you are interested in a major upgrade please contact sales@dectris.com.

4. Operation

4.1. Connecting to the Server

As a first step, the client has to open a socket connection to the server. The user can choose between UDP or TCP protocol. It is recommended to use TCP because it is more reliable.

To open the connection, the user has to specify the IP of the DCS4, which is by default 192.168.0.90. The port of the socket server is 1030 for the UDP and 1031 for the TCP protocol.

The commands sent to the server are ASCII strings. The response is a byte array, which has to be interpreted as string, integer array, or float array, depending on the command. All commands return responses of well defined lengths. Since the response might be fragmented on its way through the network, the client might have to read several packages, until the expected response length is reached.

In case of an error, the server will respond with a negative number. The meaning of the error codes can be found in Section 7.

4.2. Initialization

First of all, it should be checked that the socket server has the expected version. This is done with help of the command "-get version".

Afterwards, the detector system should be initialized by invoking the "-reset" command. This takes about two seconds, plus half a second per module. This command loads settings for Cu X-rays and enables the flat-field correction and the interpolation for bad channels (more details on the default values after initialization can be found in Section 6).

The DCS4 automatically detects how many modules are connected when sending the reset command. Nonetheless, it is possible to read out only a few connected modules. The number of active modules to be used can be specified using the "-nmodules" command.

By default the commands affect all active modules. Some commands can be invoked on single modules. This can be achieved by selecting a specific module with the "-module" command. The same command can also be used, to restore the default behavior by selecting all active modules.

4.3. Settings

4.3.1. Predefined Settings

The simplest way to operate the detector is to load predefined settings for a copper or molybdenum tube with the "-settings" command. Depending on the module type (sensor geometry; please see User Manual), predefined settings for chromium to silver X-rays are also available.

4.3.2. Setting Threshold and Energy

The predefined settings can be customized by explicitly specifying the energy threshold and the X-ray energy with the command "-kthreshenergy". The allowed ranges for these parameters depend on the module type (sensor geometry; please see User Manual).

4.4. Frame Rates

The frame rate of MYTHEN2 systems has an upper limit depending on the performance of your detector modules (X or R).

The highest allowed frame rates should be achievable independent of the chosen protocol (UDP/TCP), the applied data corrections, or the number of bits read out. However, for frame rates above hundred frames per second, it will be necessary to read out several frames per readout command (by using $n > 1$ in `-readout [n]`).

To minimize the readout time between two frames, during which no X-rays are registered, the number of bits to be read out can be reduced by the command `-nbits`. However the benefit is quite small, since the readout time between two frames only drops by one third.

4.5. Acquisition Control

Before starting an acquisition, the number of frames and the duration of the frames have to be programmed. The former can be done with the `-frames` command, the latter with the `-time` command. Since the duration is specified in units of 100 ns, the argument can be a large number. Therefore, the user should make sure to handle these numbers correctly. For example in the C/C++ programming language, it will be necessary to use the data type *long long*, which comprises 64 bits. If the acquisition consists of more than one frame, the sum of the duration of one frame and the delay between two frames must not be shorter than the maximum allowed frame rate (for details, please see User Manual). When programming a sequence exceeding the maximum allowed frame rate, the error code for an invalid argument will be returned.

An acquisition is started with the `-start` command. Afterwards, the frames are read out with the `-readout [n]` command. The readout command returns a byte array, which has to be interpreted as an array of signed 32-bit integers. The first integer encodes the number of detected photons in channel 0, the next integer the number of detected photons in channel 1, and so on. A readout command that is sent while the corresponding frames are still acquired will return the data as soon as it gets available. Nonetheless it is recommended to send the readout command only when the data is ready. If the exposure time is known, the client can simply sleep during this period, otherwise the status of the acquisition can be polled with the `-get status` command until the data becomes ready.

An acquisition can be interrupted with the `-stop` command. If the stop command occurs during an ongoing exposure, the value of the counters at that instant of time can be retrieved by sending an additional readout command after the stop command.

During acquisition parameters affecting the acquisition sequence, the detector settings, or the data corrections must not be changed. When trying to do so, the error code for an unfinished acquisition will be returned.

4.6. Data Corrections

For channels marked as bad, the server will replace the number of counts with the average number of counts of the next lower and the next upper working channels. With the `-badchannelinterpolation` command the user can turn off this correction. In this case, the number of counts will equal to -2 for these channels.

The server automatically applies a flat-field correction to the data. This behavior can be disabled with the `-flatfieldcorrection` command.

For high incoming X-ray fluxes (more than about $5 \cdot 10^4$ counts per second and channel), the rate correction can be turned on with the "-ratecorrection" command.

4.7. Triggering and Gating

The standard and the continuous trigger modes can be enabled by the "-trigen" and "-conttrigen" commands. The timing of the measurements can be adjusted with the "-delbef" and "-delafter" commands.

The gating mode is turned on by the "-gateen" command. The number of gates can be programmed with the "-gates" command.

4.8. Debugging

To verify the correct interpretation of the data, the -testpattern command can be used. This command returns a test data set with the number of counts for each channel equal to the channel number.

After sending the command "-log start", the socket server will log its activities into a file on the DCS4 until a "-log stop" command is sent. The content of the file is sent to the user as a response to the "-log read" command. Since there is only limited disk and memory space on the DCS4, the logging functionality should only be turned on for short periods of time.

For debugging problems related to the transmission of the data packages over the socket, the freely available network protocol analyzer WireShark can be used.

5. Example Applications

As a starting point, a Unix and a Windows C application with some comments are provided with the documentation. They set up a socket connection to the DCS4 and read back the module serial numbers. After a short acquisition is performed, the result is printed on the command prompt. The Windows version was developed with the freely available Microsoft Visual C++ 2008 Express Edition.

6. Commands

6.1. Command List

Synopsis: Specifies the format of the ASCII command. Values in angle brackets have to be replaced by the user: *b* stands for a boolean (*i.e.* a 0 or 1), *n* for an integer, *f* for a float, *time* for a 64-bit integer (*long long* or `int64_t`, used for the timing commands). Values in square brackets are optional.

Arguments: Gives the meaning of the arguments. Default values after initialization are set in *italic* type.

Description: Describes the effect of the command.

Return Type: Specifies the data type of the returned byte array. By default, the command returns an integer: a zero indicating successful execution of the command, or a negative number in case of an error (see section Error Codes). If the return value is of type float, the error code is returned as a float. A character comprises 1 byte, an integer 4 bytes, a float 4 bytes, and a long long 8 bytes.

Return Size: Specifies the number of data elements in the returned byte array. For example, a response of return type *int* and return size 1280 is 5120 bytes long. N_{MOD} stands for the number of active modules (set by "-nmodules"). N_{CHAN} stands for the total number of channels of the active modules (*e.g.* 2560 for two modules with 1280 channels each).

6.1.1.General Commands

Table 1: GENERAL COMMANDS.

<i>Synopsis</i>	<i>Argument</i>	<i>Description</i>	<i>Return Type</i>	<i>Return Size</i>
-get assemblydate		Returns the assembly date of the system as character array.	char	50
-get badchannels		Returns the status of each channel. A value of 0 means that the channel is working, a value of 1 means that it is defective.	int	N _{CHAN}
-get commandid		Returns the current command identifier. The command identifier increases by one whenever a client sends a command to the server. This can be used to make sure that no other clients sent any command to the server.	int	1
-get commandsetid		Returns the current set command identifier. The set command identifier increases by one whenever a client sends a command to the server, which does not start with “-get”. It can be used to make sure that no other clients changed the state of the system.	int	1
-get dcstemperature		Returns the temperature of the SoC die on the DCS4 in degrees Kelvin.	float	1
-get frameratemax		Returns the maximum allowed frame rate for the active detector modules.	float	1
-get fwversion		Returns the version string of the FPGA firmware running on the DCS4, which is of the format “01.03.00”. The returned string is terminated by a Null character.	char	9
-get humidity		Returns the relative humidity measured on each module.	float	N _{MOD}
-get hv		Returns the high voltage at which the modules are operated.	Int	N _{MOD}
-get modchannels		Returns the number of channels for each module.	int	N _{MOD}
-get modfwversion		Returns the concatenated version strings of the FPGA firmware	char	N _{MOD} *8+1

		running on the modules, each of the format "01.03.00". The returned string is terminated by a Null character.		
-get modnum		Returns the serial numbers of the modules, which have to be displayed as hex-numbers.	int	N_{MOD}
-get module		Returns either the index of the selected module or 65535 when all modules are selected.	int	1
-get nmaxmodules		Returns the maximal number of modules that can be connected to this DCS4.	int	1
-get nmodules		Returns the number of active modules that are connected to DCS4.	int	1
-get sensormaterial		Returns an identifier for the sensor material of each module. 0: Silicon	int	N_{MOD}
-get sensorthickness		Returns the nominal sensor thickness of each module in μm .	int	N_{MOD}
-get sensorwidth		Returns the nominal sensor width of each module in μm .	int	N_{MOD}
-get systemnum		Returns the serial number of the MYTHEN2 system.	int	1
-get temperature		Returns the temperature of all active modules in degrees Kelvin.	float	N_{MOD}
-get testversion		In case of a development version returns an additional identifier to the software version.	char	7
-get version		Returns the software version of the socket server, which is of the form "M3.0.0". The returned string is terminated by a Null character.	char	7
-module <n>	Module index $0 \leq n < N_{MOD} \mid$ 65535	Selects the module at the given index as the target of module specific commands. The sequence of modules follows the sequence, in which they are connected to the USB connectors. An argument of 65535 selects all active modules. After initialization all modules are selected.	int	1
-nmodules <n>	Number of modules	Sets the number of active modules (N_{MOD}). All modules are set back to default settings. After initialization the number of active modules is	int	1

set to the number of connected modules.

-reset	Sets the detector back to default settings. This command takes about two seconds, plus half a second per module. A possibly running acquisition is stopped.	int	1
--------	---	-----	---

6.1.2. Acquisition Control

Table 2: ACQUISITION CONTROL COMMANDS.

<i>Synopsis</i>	<i>Argument</i>	<i>Description</i>	<i>Return Type</i>	<i>Return Size</i>
-delafter <time>	Delay in units of 100 ns. Default is 0 ns.	Sets the delay between two subsequent frames (“delay after frame”). Setting the delay below the readout time for the current settings has no effect. When doing this, the effective time between two subsequent frames is equal to the readout time. For acquisitions with multiple frames, the programmed sequence must not exceed the maximal frame rate. Only relevant for the internal and the single trigger mode.	int	1
-frames <n>	Number of frames. Default is 1.	Sets the number of frames within an acquisition. For acquisitions with multiple frames, the programmed sequence must not exceed the maximal frame rate.	int	1
-get delafter		Returns the programmed delay between two subsequent frames. If the programmed delay is below the readout time for the current settings, the effective time between two subsequent frames is equal to the readout time.	long long	1
-get frames		Returns the number of frames.	int	1
-get nbits		Returns the number of bits to be read out.	int	1

-get readouttimes		Returns the readout times for the four different bit modes in units of 100 ns. The first element gives the readout time for the 24-bit mode, followed by the values for the modes with 16, 8, and 4 bits.	long long	4
-get status		Returns the status of the DCS4 as a bit pattern: Bit 0: Acquisition running Bit 3: Acquisition running but exposure inactive Bit 16: No data available for readout (empty buffer)	int	1
-get time		Returns the programmed exposure time of one frame in units of 100 ns.	long long	1
-nbits <n>	n = 4 8 16 24	Sets the number of bits to be read out, thereby determining the dynamic range and the readout time. After initialization 24 bits are read out.	int	1
-readout [n]	Number of frames.	Returns the number of detected photons in each detector channel. The DCS4 can internally buffer a large number of frames. This command returns the data of the n oldest frames in the buffer (<i>i.e.</i> the frames are returned in the same order as they were acquired). If no argument is specified, one frame is read out. If there is an ongoing measurement and the n requested frames are not yet ready, the server sends the data as soon as the frames get available. If the readout fails for some reason, all count values are set to -1. If the bad channel interpolation is disabled, bad channels get a value of -2.	int	n*N _{CHAN}
-start		Starts an acquisition with the programmed number of frames or gates.	int	1
-stop		Stops the current acquisition. If the acquisition is stopped during an ongoing exposure, the counter values at that time can be read out with a subsequent readout command.	int	1

-time <time>	Exposure time in units of 100 ns. Default is 1 s.	Sets the exposure time of one frame. For acquisitions with multiple frames, the programmed sequence must not exceed the maximal frame rate.	int	1
--------------	---	---	-----	---

6.1.3. Detector Settings

Table 3: DETECTOR SETTING CONTROLLING COMMANDS.

<i>Synopsis</i>	<i>Argument</i>	<i>Description</i>	<i>Return Type</i>	<i>Return Size</i>
-energy <f>	$E_{\min} \leq f \leq E_{\max}$ in units of keV. Default is 8.05 keV.	Sets the X-ray energy for the selected modules while keeping the current threshold energy. The energy is used to provide an optimal flat-field correction. The supported energy range depends on the module type. This command takes about half a second per module.	int	1
-get energy		Returns the X-ray energies of the selected modules.	float	N_{MOD}
-get energymax		Returns the maximum supported X-ray energy of the detector modules in units of keV.	float	N_{MOD}
-get energymin		Returns the minimum supported X-ray energy of the detector modules in units of keV.	float	N_{MOD}
-get kthresh		Returns the threshold of the modules in keV.	float	N_{MOD}
-get kthreshmax		Returns the maximum supported threshold energy of the detector modules in units of keV.	float	N_{MOD}
-get kthreshmin		Returns the minimum supported threshold energy of the detector modules in units of keV.	float	N_{MOD}
-kthresh <f>	$\text{Thr}_{\min} \leq f \leq \text{Thr}_{\max}$ in units of keV. Default is 6.4 keV.	Sets the energy threshold for the selected modules while keeping the current X-ray energy. The supported threshold range depends on the module type. This command takes about half a second per module.	int	1

-kthresenergy <f1> <f2>	$\text{Thr}_{\min} \leq f1 \leq \text{Thr}_{\max}, E_{\min} \leq f2 \leq E_{\max}$ in units of keV. Defaults are $f1 = 6.4$ keV and $f2 = 8.05$ keV.	Sets the energy threshold and the X-ray energy for the selected modules. The supported ranges depend on the module type. This command takes about half a second per module.	int	1
-settings <name>	name = Cu Mo Cr Ag	Loads predefined settings for the selected modules and the specified X-ray radiation. The energy threshold is set to a suitable value depending on the module type (for more details consult the User Manual). The availability of Cr and Ag settings depends on the module type. This command takes about half a second per module. After initialization settings for Cu X-rays are loaded.	int	1

6.1.4. Data Correction

Table 4: DATA CORRECTION COMMANDS.

<i>Synopsis</i>	<i>Argument</i>	<i>Description</i>	<i>Return Type</i>	<i>Return Size</i>
- badchannelinterpolation 	0: Disable 1: Enable	Turns on or off the interpolation routine for bad channels. When enabled, the number of counts for a bad channel is set to the average number of counts of the next lower and the next upper working channels. If there is no next lower (upper) working channel, the number of counts of the next upper (lower) working channel is used. When disabled the number of counts for a bad channel is set to -2.	int	1

-flatfield <slot> <flat-field>	slot = 0 1 2 3 flat-field: binary block with N_{CHAN} elements of type uint32. The complete data set is 13 byte long: the command in ASCII string format, trailed by a space. This is followed by $4*N_{\text{CHAN}}$ bytes of binary data.	Stores a customer specific flat-field to the given slot. The uploaded flat-field gets immediately active and is applied to the data in case the flat-field correction is enabled (cf. command -flatfieldcorrection). Issuing an -energy or -settings command will reactivate the default flat-field provided by DECTRIS.	int	1
-flatfieldcorrection 	0: Disable 1: Enable	Enables or disables the flat-field correction. After initialization the flat-field correction is enabled.	int	1
-get badchannelinterpolation		Returns whether the bad channel interpolation is enabled (1) or not (0).	int	1
-get cutoff		Returns the maximal count value before flat-field correction.	int	1
-get flatfield		Returns the currently loaded flat-field calibration of all modules.	int	N_{CHAN}
-get flatfieldcorrection		Returns whether the flat-field correction is enabled (1) or not (0).	int	1
-get ratecorrection		Returns whether the rate correction is enabled (1) or not (0).	int	1
-get tau		Returns the dead time constant used by the rate correction for all active modules in units of ns.	float	N_{MOD}
-loadflatfield <slot>		Activates a customer specific flat-field stored in the given slot. The flat-field is only applied to the data if the flat-field correction is enabled (cf. command -flatfieldcorrection). Issuing an -energy or -settings command will reactivate the default flat-field provided by DECTRIS.	int	1

-ratecorrection 	0: Disable 1: Enable	Enables or disables the rate correction. After initialization the rate correction is disabled.	int	1
-tau <f>	f > 0 or f = -1.0 in units of ns	Sets the dead time constant (used by the rate correction) for the selected modules. If the argument is equal to -1.0 the system uses predefined values appropriate for the loaded settings.	int	1

6.1.5. Trigger / Gate

Table 5: TRIGGER AND GATING CONTROL COMMANDS.

<i>Synopsis</i>	<i>Argument</i>	<i>Description</i>	<i>Return Type</i>	<i>Return Size</i>
-conttrigen 	0: Disable 1: Enable	Enables or disables the continuous trigger mode. In this mode, each frame requires a trigger signal. For acquisitions with multiple frames, the programmed sequence must not exceed the maximum frame rate.	int	1
-delbef <time>	Delay in units of 100 ns. Default is 0 ns.	Sets the delay between a trigger signal and the start of the measurement ("delay before frame"). For acquisitions with multiple frames, the programmed sequence must not exceed the maximum frame rate. Only relevant for the single and the continuous trigger modes.	int	1
-gateen 	0: Disable 1: Enable	Enables or disables the gated measurement mode. For acquisitions with multiple frames, the programmed sequence must not exceed the maximum frame rate.	int	1
-gates <n>	Number of gates. Default is 1.	Sets the number of gate signals within one frame.	int	1
-get conttrig		Returns whether the continuous trigger mode is enabled (1) or not (0).	int	1
-get delbef		Returns the delay between a trigger signal and the start of the measurement.	long long	1
-get gate		Returns whether the gated mode is enabled (1) or not (0).	int	1
-get gates		Returns the number of gates.	int	1
-get trig		Returns whether the trigger mode is enabled (1) or not (0).	int	1

-trigen 	0: Disable 1: Enable	Enables or disables the single trigger mode. The acquisition starts after the trigger signal. Subsequent frames are started automatically. For acquisitions with multiple frames, the programmed sequence must not exceed the maximum frame rate.	int	1
-------------	-------------------------	---	-----	---

6.1.6. Debugging

Table 6: DEBUGGING COMMANDS.

<i>Synopsis</i>	<i>Argument</i>	<i>Description</i>	<i>Return Type</i>	<i>Return Size</i>
-log start		Instructs the socket server to log its activities into a file on the DCS4. Do not enable logging except for debugging purposes, since it will slow down the server and because there is only limited space for the log file on the DCS4.	int	1
-log status		Returns whether the logging is running (1) or not (0).	int	1
-log stop		Stops the logging functionality of the socket server. The command returns the size of the log file.	int	1
-log read		Instructs the socket server to send back the content of the log file. The length of the server response has to be inferred from the return value of the "-log stop" command.	char	
-testpattern		Returns a test data set with the number of counts for each channel equal to the channel number. Can be used to verify the client implementation of the readout mechanism.	int	N _{CHAN}

7. Error Codes

If the response of the server consists of a single, negative number, an error has occurred during the execution of the command. The following table lists the corresponding error codes.

Table 7: LIST OF ERROR CODES.

<i>Error code</i>	<i>Description</i>
-1	Unknown command
-2	Invalid argument
-3	Unknown settings
-4	Out of memory
-5	Module calibration files not found
-6	Readout failed
-7	Acquisition not finished
-8	Failure while reading temperature and humidity sensor
-9	Invalid license key
-10	Flat-field file not found
-11	Bad channel file not found
-12	Energy calibration file not found
-13	Noise file not found
-14	Trimbit file not found
-15	Invalid format of the flat-field file
-16	Invalid format of the bad channel file
-17	Invalid format of the energy calibration file
-18	Invalid format of the noise file
-19	Invalid format of the trimbit file
-20	Version file not found
-21	Invalid format of the version file
-22	Gain calibration file not found
-23	Invalid format of the gain calibration file
-24	Dead time file not found
-25	Invalid format of the dead time file
-26	High voltage file not found
-27	Invalid format of high voltage file
-28	Energy threshold relation file not found
-29	Invalid format of the energy threshold relation file
-30	Could not create log file

-31	Could not close log file
-32	Could not read log file
-50	No modules connected
-51	Error during module communication
-52	DCS initialization failed
-53	Could not store customer flat-field

8. Release Notes

8.1. Version 4.1.0

- Support for customer specific flat-fields. Flat-fields can be stored in four slots with the `-flat-field` command. These flat-fields are permanently stored on the DCS4. They can be loaded with the `-loadflat-field` command. The default flat-fields provided by DECTRIS get reactivated after a `-settings` or `-energy` command.
- New error code -53

8.2. Version 4.0.1

- New command `-log status`
- `-get modfwversion` now returns the correct module FPGA firmware string. Version 4.0.0 by mistake always returned a build number of 2.

8.3. Version 4.0.0

There are no modifications with respect to 4.0.0-T03.

8.4. Version 4.0.0-T03

8.4.1. Modifications Compared to 4.0.0-T02

- New command `-get commandsetid` to check whether another client changed the state of the system.
- New error codes -51, -52
- The command `-get hv` returns the measured high voltage. Older test versions returned the nominal high voltage.
- Check for disconnected modules during acquisition
- Improved error handling and error propagation
- Improved server stability

8.5. Version 4.0.0-T02

8.5.1. Modifications Compared to 4.0.0-T01

- The commands `-get kthreshmax`, `-get kthreshmin`, `-get energymax`, `-get energymin` return the corresponding information for each module separately.
- New error codes -28, -29

8.6. Version 4.0.0-T01

The major version 4 supports the new MYTHEN2 hardware, especially modules with 640 channels.

8.6.1. Modifications

- The DCS is now capable of buffering several thousands of frames. It is therefore less important to immediately readout a frame after it has been acquired.
- The command “-readout [n]” optionally takes the number of frames to be read out as an argument.
- A readout command waiting for the data of an ongoing exposure can be interrupted by sending a stop command. In previous versions the server was not able to react to a stop command before a readout command completed.
- After stopping an acquisition, the current counter value can be read out by a subsequent readout command.
- In case of an invalid license key all commands starting with “-get” still work. Other commands will return the newly added error code for an invalid license key.
- MYTHEN2 systems have an upper limit on the frame rate depending on the module performance. The maximum supported frame rate of the active modules can be retrieved by the command `-get frameratemax`.
- During an acquisition, parameters affecting the acquisition sequence, the detector settings, or the data corrections must not be changed. When trying to do so, the error code for an unfinished acquisition will be returned.
- MYTHEN2 systems automatically detect the number of connected modules during initialization. In case that no modules are connected, commands depending on the number of connected modules return the newly added error code for no connected modules. The `-nmodules` command always sets all modules back to default settings.
- The commands `-get sensormaterial` and `-get sensorthickness` return the corresponding information for each module separately.
- The meaning of bit 3 of the status command is slightly changed compared to older versions. For MYTHEN systems, it was high when the system was waiting for a trigger signal. For MYTHEN2 systems it is high, whenever an acquisition is running but the exposure is inactive. So, when the system is waiting for a trigger, the bit is high as before. Additionally it is also high during a possibly delay before the exposure starts as programmed by the command `-delbef`. It is also high during inactive periods in gating mode.
- In the gating mode, the rate correction uses not any more the programmed acquisition time, but the actual exposure time to correct the data.

8.6.2. New Commands

- The number of channels of each module can be retrieved with the command `-get modchannels`
- The command `-get sensorwidth` returns the width of the sensors in micrometers.
- The modules of MYTHEN2 systems include a combined temperature and humidity sensor. The corresponding values can be read out with the new commands `-get temperature` and `-get humidity`.
- The temperature of the SoC die on the DCS can be read out with the command `-get dcstemperature`.
- The maximum supported frame rate can be retrieved by the command `-get frameratemax`.

- The FPGA firmware versions of the DCS and the modules can be queried by the commands `-get fwversion` and `-get modfwversion`.
- The readout times of the different readout modes can be retrieved by the command `-get readouttimes`.
- The high voltage at which the modules are operated can be retrieved by the command `-get hv`.

8.6.3. New Error Codes

The new error codes -7, -8, -9, -26, -27, and -50 have been added.

8.6.4. Removed Commands

- The `-readoutraw` command, formerly required for achieving maximal frame rates, is no longer supported because of the improved performance of the DCS4.
- The DCS4 does not support changing the input and output polarities of the trigger and enable signals. Therefore the following commands have been removed: `-get inpol`, `-get outpol`, `-inpol`, `-outpol`.
- Commands marked as deprecated in version 3.0.0 have been removed: `-autoreadouttime`, `-autosettings`, `-flipchannels`, `-get autoreadouttime`, `-get flipchannels`, `-get settings`, `-get settingsmode`, `-legacy`, `-legacy2`, `-read`, `-settings StdCu | StdMo | HgCr | HgCu | FastCu | FastMo`, `-trimfile`.

8.7. Version 3.0.0

The main objective of the new release is a substantially improved detector calibration in parallel with a considerably simplified detector usage. The user has no longer to worry about choosing suitable gain settings, this task is now entirely handled by the detector. Depending on the energy threshold, the server continuously adjusts the gain settings to ensure noiseless measurements with the smallest possible detector dead-time.

8.7.1. Modifications

- Predefined settings are loaded by simply specifying the X-ray energy in the form `"-settings Cu"`. No gain has to be specified.
- For optimal flat-field correction, the server has to be informed about the X-ray energy. It is recommended to either use predefined settings or to set the energy threshold and the energy with help of the new `"-kthreshenergy"` command. The energy can also be set with the new command `"-energy"`.
- Improved precision of the rate correction.
- Many new commands to retrieve the state of the system: `"-get contrig"`, `"-get cutoff"`, `"-get delafter"`, `"-get delbef"`, `"-get frames"`, `"-get gate"`, `"-get gates"`, `"-get inpol"`, `"-get module"`, `"-get nmaxmodules"`, `"-get outpol"`, `"-get trig"`.
- When increasing the number of active modules with the `"-nmodules"` command, all modules are set back to default settings.
- Multi-module systems can be controlled without looping over individual modules:
 - Commands retrieving module specific information return by default the values for all modules.

- Commands setting module specific parameters affect by default all modules.
- Instead of additional gain settings, extended threshold ranges can be requested by the customer. Therefore the commands to set the energy or threshold only accept arguments within the supported range. The upper and lower limits can be retrieved by “-get energymin”, “-get energymax”, “-get kthreshmin”, “-get kthreshmax”.
- New command “-get commandid” to retrieve a command identifier. This can be used to make sure that no other clients changed the state of the system.
- New command “-testpattern” for debugging purposes.
- Legacy2 mode for backwards compatibility with version 2.

8.7.2. Migration Notes

If you migrate from version 2 take care of the following changes:

- For optimal flat-field correction, the X-ray energy has to be specified.
- After initialization all modules are selected
- Commands retrieving module specific information return the values for all values.
- The “-kthresh” commands only accepts arguments within the range supported by the specific MYTHEN system.
- Since the gain automatically changes with the threshold energy, the number of bad channels can change with the specified threshold energy.
- The “-autosettings” command is deprecated. Please use “-kthreshenergy” instead.
- The commands “-get settings” and “-get settingsmode” are deprecated.
- The arguments “StdCu”, “HgCr”, ... of the “-settings”command are deprecated.

8.7.3. Legacy2 Mode

If for some reasons you are not going to migrate, you can enable the Legacy2 mode with the “-legacy2” command. After having sent this command, the interface to the server is identical with version 2 and your software will still run. Although the interface is the same, keep in mind that the calibration of the detector will be different than for a system delivered with version 2. For example you will not see any differences in data acquired with “StdCu” and “FastCu” settings, because these deprecated arguments are internally mapped to the same detector settings. Furthermore there are upper and lower limits on the allowed threshold and energy ranges.

8.7.4. Known Issues

- Sending commands other than “-readout”, “-status” or “-stop” during an acquisition can result in misbehavior of the detector.

8.8. Version 2.1.0

- The bad channel interpolation now supports neighboring bad channels. In this case the interpolated value is equal to the average number of counts of the next lower and the next upper

working channels. If there is no next lower (upper) working channel, the number of counts of the next upper (lower) working channel is used.

- Different modules can now have different dead time constants for the rate correction. This fixes the first known issue of version 2.0.5.
- Increasing the number of active modules now works flawlessly, fixing the second known issue of version 2.0.5.
- The rate correction is now applied before the flat-field correction.

8.9. Version 2.0.5

New command to get the assembly date of the system („-get assemblydate“).

8.9.1. Known Issues

- When operating a multi-module system with differing gain settings (Std, Hg, Fast) for the various modules, the rate correction uses the dead time (“tau”) constant of the currently selected module for all modules. Correctly it would use different dead time constants depending on the gain of each module. Therefore the rate correction should only be used when all modules of a multi-module system have the same gain settings.
- When setting the number of active modules to a value larger than the original number of delivered modules the server will stuck when loading settings for the additional modules. To correctly initialise additional modules, the serial number of the modules have to be added to the configuration file /etc/conf.d/version on the DCS.

8.10. Version 2.0.4

New commands (“-get sensorthickness”, “-get sensormaterial”, “-get systemnum”) to retrieve information about the sensor and the system. If the information cannot be found, one of the new error codes -20 and -21 is returned. If you upgrade an older system to version 2.0.4, the sensor information will not be available.

Several improvements and bug fixes for multi-module systems (especially for the MYTHEN 24K system) have been implemented:

- Faster execution of the reset command.
- After execution of the reset command the first module (module 0) is selected (instead of the last one).
- Correct sending of data if its length is a multiple of the UDP/TCP packet size.
- The initialization sets the number of modules to the number of delivered modules. For systems which are delivered with less modules than the DCS maximally supports (e.g. a MYTHEN 3K with 3 modules on a DCS6), the user no longer needs to specify the number of modules. The command to set the number of active modules is only necessary for running a system with less (or more) than the number of delivered modules. In contrast to earlier versions (2.0.2 and 2.0.3), the corresponding command has to be executed after the reset command, since a reset now sets the number of modules to the number of delivered modules.

8.11. Version 2.0.3

The server is now able to handle optimized high gain settings. This change is purely internal and has no consequences for the user.

8.12. Version 2.0.2

The reset command does not change the number of active modules any more. This prevents the issuing of error messages by the reset command when working with a not fully equipped multi-module system.

8.13. Version 2.0.1

Two bugs in version 2.0.0 have been fixed:

- Due to a type conversion problem, the server-side flat-field correction truncated the correction factors to integers. As a consequence many channels returned zero counts in all readouts. The problem only occurred when the server-side flat-field correction was turned on.
- The result of the "-get badchannels" did not have the format as described in this specification.

8.14. Version 2.0.0

Version 2.0.0 is a complete redesign of the socket server and the user is encouraged to adapt his older implementation to the new version, since it offers better stability and more functionality.

8.14.1. New Functionalities

- The server can be controlled over a TCP socket
- Support for up to 24 modules
- Fast settings are available to the user
- Built-in support for flat-field correction
- Built-in support for rate correction
- Automatic loading of optimal settings based on the required threshold
- Many new commands to read out the current settings and parameters
- Logging functionality

8.14.2. Migration Notes

If you migrate from version 1.3.0, take care of the following changes:

- By default the automatic readout of the first frame in an acquisition is disabled.
- The flipping of the channels is turned off after initialization.
- By default the flat-field correction is performed directly on the DCS.
- The "-read" command is deprecated and should be replaced by the "-get flat-field" command.
- The "-trimfile" command is deprecated and should be replaced by the "-settings" command.

8.14.3. Legacy Mode

If for some reasons you are not going to migrate, you can enable the legacy mode with the "-legacy" command. After having sent this command, the server should behave identically to version 1.3.0 up to bug fixes and the more detailed error responses.