DBE Memo#12.1
Mark 5 Memo #090.1

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
HAYSTACK OBSERVATORY
WESTFORD, MASSACHUSETTS 01886

5 June, 2012

Telephone: *781-981-5951*
Fax: *781-981-0590*

TO: Distribution

FROM: Chester Ruszczyk, Mikael Taveniku

SUBJECT: Digital Backend Software Command Set – Ver. 1.2

1. Introduction

This document describes the command set that the program to be used as the primary software interface on the second generation VLBI digital backends must support. This program will be the command and control interface for the embedded device. The name of the application is rdbe_server, for DBE command and control server daemon, where RDBE refers to the ROACH Digital Backend.

The standard method of communication with the application will be via a TCP connection to port 5000 via the standard 10/100/1000 Mbps Ethernet interface. Multiple simultaneously connections, up to a maximum supplied by a command line argument, are allowed, with commands and queries being executed in the order received, regardless of their origin.

The VSI-S specification defines the syntax of communication into and out of the RDBE.

2. Notes on RDBE Command Set

Note the following with respect to the command set:

1. All of the commands/queries expect the VSI-S communications protocol and command/response syntax.
2. Commands/queries are case *insensitive*.

3. VSI-S Command, Query and Response Syntax

The following explanation of the VSI-S syntax may be useful in understanding the structure of commands, queries, and their respective responses. This explanation has been lifted directly from the VSI-S specification.

3.1. *Command Syntax*

Commands cause the system to take some action and are of the form

<keyword> = <field 1> : <field 2> : …. ;

where <keyword> is a VSI-S command keyword. The number of fields may be either fixed or indefinite; fields are separated by colons and terminated with a semi-colon. A field may be of type decimal integer, decimal real, integer hex, character, literal ASCII, or a VSI-format time code. White space between tokens in the command line is ignored, however most character fields do not allow embedded white space.

3.2. *Command-Response Syntax*

Each command elicits a response of the form

!<keyword> = < return code > [:<RDBE-specific return> :….] ;

where

<keyword> is the command keyword

<return code> is an ASCII integer as follows:
0 - action successfully completed
1 - action initiated or enabled, but not completed
2 - command not implemented or not relevant to this RDBE
3 - syntax error
4 - error encountered during attempt to execute
5 - currently too busy to service request; try again later
6 - inconsistent or conflicting request
7 - no such keyword
8 - parameter error

<RDBE-specific return> - one or more optional fields specific to the particular RDBE, following the standard fields defined by VSI-S; fields may be of any type, but should be informative about the details of the action or error.

3.3. *Query and Query-response Syntax*

Queries return information about the system and are of the form

<keyword> ? <field 1> : <field 2> : …. ;

with a response of the form

!<keyword> ? <field 1(return code)> : <field 2> : <field 3> : …: [<RDBE-specific return>];

where

<return code> is an ASCII integer as follows:

0 - query successfully completed
1 - action initiated or enabled, but not completed
2 - query not implemented or not relevant to this RDBE
3 - syntax error
4 - error encountered during attempt to execute query
5 - currently too busy to service request; try again later
6 - inconsistent or conflicting request
7 - no such keyword
8 - parameter error
9 - indeterminate state

Note: *A 'blank' in a returned query field indicates the value of the parameter is unknown. A '?' in a returned query field indicates that not only is the parameter unknown, but that some sort of error condition likely exists.*

4. RDBE Command / Query Summary (by Category)

The RDBE system has some of the functionality that was once located in the Mark5 unit. The RDBE command set reflects this by utilizing similar style commands that were previously issued to the Mark5B system. Additionally, newer commands were required to add support for features that did not previously exist.

The command set will be presented as a set of commands and queries separated into categories, followed by alphabetical listing of the commands.

4.1. General

dbe_execute	20	Execute a specific system level command
dbe_hw_version?	22	Get the hardware version numbers of the RDBE (query only)
dbe_status?	30	Get system status (query only)
dbe_sw_version?	31	Get the software version information from the RDBE (query only)

4.2. System Setup and Monitoring

dbe_1pps_mon	8	Set the 1pps monitoring broadcast
dbe_quantize	28	Set channel quantization
dbe_dot?	15	Get the Data Observable Time (DOT) clock information (query only)
dbe_dot_inc	18	Increment the DOT clock time on the next 1pps tick
dbe_dot_set	19	Set the DOT clock on the next 1pps tick
dbe_fs	21	Resynchronize the 1pps and 5MHz signal on the frequency

		synthesizer board
dbe_personality	28	Set the RDBE FPGA bit code personality
dbe_tsys_diode_ctl	32	Set Tsys diode control settings
dbe_tsys_mon	33	Set the Tsys monitoring broadcast

4.3. Data Communication Commands

dbe_arp	11	Set the IP to MAC address resolution
dbe_mac	26	Set the 10G Core source MAC address
dbe_ifconfig	23	Set RDBE 10G network interface configuration

4.4. Packet Format Commands

dbe_data_format	13	Get the VLBI packet format (query only)
dbe_packet	27	Get the packet structure
dbe_ioch_assign	25	Set the input to output channel assignments for the VLBI payload

4.5. Data Transfer

dbe_data_connect	12	Set the destination IP
dbe_data_send	14	Set the time interval for transmitting data

4.6. ALC Commands

dbe_alc	9	Set the ALC attenuator setting for INPUT 0/1
dbe_alc_fpgaver?	10	Get the ALC board FPGA bit code version (query only)

4.6. DDC Commands

dbe_dc_cfg	15	Set the digital down converters configuration
dbe_ddc_quantize	16	Manage the digital down converters quantization
dbe_xbar	35	Set the DDC crossbar switch positions

5. RDBE Command/Query (Alphabetical)

dbe_1pps_mon	8	Set the 1pps monitoring broadcast
dbe_alc	9	Set the ALC attenuator setting for INPUT 0/1
dbe_alc_fpgavers	10	Get the ALC board FPGA bit code version (query only)
dbe_arp	11	Set the IP to MAC address resolution
dbe_data_connect	12	Set the destination IP
dbe_data_format?	13	Get the VLBI packet format (query only)
dbe_data_send	14	Set the time interval for transmitting data
dbe_dc_cfg	15	Set the digital down converters configuration
dbe_ddc_quantize	16	Manage the digital down converter quantization
dbe_dot?	17	Get the Data Observable Time (DOT) clock information (query only)
dbe_dot_inc	18	Increment the DOT clock time on next 1pps tick
dbe_dot_set	19	Set the DOT clock time on next 1pps tick
dbe_execute	20	Execute a specific system level command
dbe_fs	21	Resynchronize the 1pps and 5MHz signal on the frequency synthesizer board
dbe_hw_version?	22	Get the hardware version numbers from the RDBE (query only)
dbe_ifconfig	23	Set RDBE 10G network interface configuration
dbe_ioch_assign	25	Set the input to output channel assignments for the VLBI payload
dbe_mac	26	Set the 10G Core source MAC address
dbe_packet?	27	Get the packet structure (query only)
dbe_personality	28	Set the RDBE FPGA bit code personality
dbe_quantize	28	Set channel quantization
dbe_status?	30	Get system status (query only)
dbe_sw_version?	31	Get the software version information from the RDBE
dbe_tsys_diode_ctl	32	Set Tsys diode control settings
dbe_tsys_mon	33	Set the Tsys monitoring broadcast
dbe_xbar	33	Set the DDC crossbar switch positions

6. RDBE Command Set Details

This section contains a complete description of all RDBE commands / query in alphabetical order. Those commands with “TBD” are commands for features to be developed.

dbe_1pps_mon – Set the 1pps monitoring broadcast state

Command: → dbe_1pps_mon = <state>: <multicast IP address>:[<port>];

← !dbe_1pps_mon = <return code>;

Query: → dbe_1pps_mon?;

← !dbe_1pps_mon ? <return code>: <state> :<multicast IP address>:<port>;

Purpose: This command enables or disables the broadcast of the 1pps DOT time, as seen by the FPGA, out the standard 1G Ethernet interface.

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
state	ASCII	enable/disable	disable	This is the state of the1pps monitoring function
multicast IP Address	ASCII	00.00.00.00.00	239.0.2.20	Multicast IPv4 address to which the 1pps will be broadcast (see note 1)
port	int	2000-65556	20020	UDP Destination port number

Monitor Parameters:

Parameter	Type	Allowed values	Comments
state	ASCII	enable / disabled	active – the 1pps monitoring is enabled disabled – the 1pps is disabled
multicast IP Address	ASCII	xx.xx.xx.xx	The multicast IPv4 address
port	int	2000-65566	UDP destination port

Notes:

1. The multicast IP address must be in the range 239.0.1.0-239.255.255.255

dbe_alc – Set the ALC attenuator setting for INPUT 0/1

Command: → dbe_alc = [<input>] : <attcmd> : <solar>;

← !dbe_alc = <return code>;

Query: → dbe_alc?[input];

← !dbe_alc ? <return code> : <input> : <attcmd> : <solar> : [<input>] : [<attcmd>] : [<solar>];

Purpose: This command controls the amount of attenuation applied to the incoming signal within the ALC unit before it is passed to the sampler card.

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
input	int	0 | 1	-	Settings are for either INPUT 0 or INPUT 1(see note 1,2)
attcmd	int	0-31	16	Attenuator setting in dB (see note 3)
solar	ASCII	on | off	off	Solar mode setting (see note 4)

Monitor Parameters:

Parameter	Type	Values	Comments
input	int	0 | 1	Settings are for either INPUT 0 or INPUT 1 (see note 2)
attcmd	int	0-31	Attenuator setting in dB
solar	ASCII	on | off	Solar mode setting

Notes:

1. If an input field is not provided it is assumed that the setting in attcmd and solar are applied to both inputs

2. Specification of the [input] is optional; if not specified then both INPUT parameters are assumed

3. The ALC, by design, amplifies the incoming signal approximately by 26 db. This must be taken into account when choosing the value of the attenuator setting to apply.

4. Solar "on" provides an additional 20dB of attenuation to the base attenuator setting

dbe_alc_fpgaver – Get the ALC board FPGA bit code version number (query only)

Query :	→ dbe_alc_fpgaver?

← !dbe_alc_fpgaver? <return code> : <fpgavers>;

Purpose: This command gets the ALC FPGA bit code version number

Monitor Parameters:

Parameter	Type	Values	Comments
fpgavers	int		ALC FPGA version number

dbe_arp – Set the IP to MAC address resolution

dbe_arp

Command: → dbe_arp = <state> : <IP1>: [< MAC1>] : [<IP2>]: [<MAC2 >]: … ;

← !dbe_arp = <return code>;

Query: → dbe_arp?;

← !dbe_arp ? <return code>:<IP1> :< MAC1> : [<IP2>]: [<MAC2 >]: ...;

Purpose: This command is required since the 10G interface is presently unidirectional in our design and cannot respond to ARP queries. This command is equivalent to running the 'arp' command on Linux. This command is valid only for layer 4, UDP/IP encapsulated, data from the 10G interface.

Settable Parameters:

Parameter	Type	Allowed Values	Defaults	Comments
state	ASCII	add | delete | flush | force	null	add – adds the specific IP to MAC entry to the ARP table delete – removes the specific IP entry from the ARP table flush – removes all entries in the ARP table force – if an entry exists for the least significant address field, overwrite it (see note 1)
IP	ASCII	xx.xx.xx.xx		This is the destination IPv4 address
MAC	ASCII	00.00.00.00.00.00		IP1 MAC address to be used in data transmission (see note 2)

Notes:

1. The index of entries in the present ARP table uses the least significant address field, e.g. for XX.XX.XX.YY the value YY. If an entry is at that location, an error code will be returned to the client. A 'force' will overwrite that specific field.
2. If the RDBE is connected directly to a recording device, e.g. Mark5C, the MAC address will be the Conduant's 10GE MAC address. If the connection between the RDBE and the Mark5C is through a router, then the MAC address has to be the MAC address of the gateway to which the RDBE will use.

dbe_arp

dbe_data_connect – Set the destination IP

Command: → dbe_data_connect = <IP1>: [<INPUT >]:[<threadID>] : [<IP2>]: [<INPUT>]:[<threadID>] … ;

← !dbe_data_connect = <return code>;

Query: → dbe_data_connect?;

← !dbe_data_connect ? <return code>:<IP1> :[<INPUT>] :[<threadID>]: [<IP2>]: [<INPUT>]: :[<threadID>]...;

Purpose: This command is used when the VLBI data is encapsulated in an UDP/IP, or layer 4, and sets the destination IP address in the IP header field.

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
IP	ASCII	xx.xx.xx.xx		This is the destination IPv4 address (see note 1)
INPUT	int	0/1	0	INPUT [0/1] source for the specific destination address (see note 2)
threadID	int	threadID		If VDIF data format, then the thread id associated with output payload (see note 2)

Monitor Parameters

Parameter	Type	Allowed values	Comments
state	ASCII	active / closed	active – The destination IP address assignment is being utilized closed – A default destination IP address is assigned if data is being transmitted (see note 3)
IP	ASCII	xx.xx.xx.xx	This is the destination IPv4 address
INPUT	int	0/1	INPUT source's destination IP address
threadID	int	threadID	If VDIF data format, then the thread id associated with output payload

Notes:

1. The default destination IP address is dependent on the FPGA personality loaded. For version1.4 this is 192.168.5.11

2. For Mark5B emulation formatted packets, the IP address signifies the destination IP address of the packet. If the INPUT field is left blank, then all inputs will be addressed to this one IP address. If there are multiple IP addresses, then each INPUT will be destined to the corresponding IP address. For version 1.4 bit code, there is only 1 input that is available so the default is 0 and it cannot be changed.

3. The threadID can be a single id, e.g. 1, or it can be a range of threads addressed to the same destination, e.g. 1-16

dbe_data_format – Get the VLBI packet format (query only)

Query: → dbe_data_format?;

← !dbe_data_format ? <return code>: <data mode> : <data submode1 >:[<data submode2>];

Purpose: Get the packet format mode, either a VDIF Mark5C profile mode or Mark5B compatibility mode, used by the FPGA personality. Due to the FPGA personality implementation this command is not settable and only supports queries (see note 1)

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
payload	char	vdif | mark5B	Mark5B	transmit using VDIF (Mark5C profile) transmit in Mark5B format
data submode1	int	0|1	0	If a user specified input/channel payload assignment is being used versus the default[2]
data submode2	int	X ≥ 0 | 1,2,4,8,F	0 8	VDIF - profile to be used Mark5B - Number of channels to be written in a payload for a DDC personality. Using only 1 channel will give DC0, 2 channels = DC0, DC1, 4 channels = DC0, DC1, DC2, DC3. 0xF gives a test pattern in the packet.[3]

Notes:

1. The philosophy used with the design of the FPGA bit code is that for each different format supported a new personality would be used, instead of a general purpose formatter. This would reduce the amount of resources required within the FPGA.

dbe_data_send – Set the time interval for transmitting data

Command: → dbe_data_send = < state > : [< t_s >] : [<t_e>] : [<delta>] :[<threadID>];

← !dbe_data_send = <return code>;

Query: → dbe_data_send?;

← !dbe_data_send ? <return code>: <state> : <t_s> :<t_e> : <present DOT time> ;

Purpose: To start or stop the transfer of data from the RDBE out of the 10G Ethernet interface.

Settable Parameters:

Parameter	Type	Allowed Values	Defaults	Comments
state	ASCII	on | off	off	on - transmission of data should begin at t_s off - cease or abort transmission of data on a 1pps boundary, t_e
t_s	time	YYYYDOYHHMMSS		Optional start time of valid data on 1pps boundary
t_e	time	YYYYDOYHHMMSS		Optional end time of valid data on 1pps boundary
delta	time	≥1		Optional duration of valid data (integer seconds) (see note 1)
threadID	int	0-n	0	Thread id associated with the send command (VDIF payload ONLY)

Monitor Only Parameters:

Parameter	Type	Values	Comments
status	char	on | off | waiting	on – transmission active off – transmission inactive waiting – dbe_data_send command received and waiting for start time.
ts	time	YYYYDOYHHMMSS	start time of valid data on 1pps boundary
te	timr	YYYYDOYHHMMSS	end time of valid data on 1pps boundary
time	time	YYYYDOYHHMMSS	The present time

Notes:

1. The parameter delta can be used instead of the end time, or the end time can be specified, since one implies the other.
2. For more information about use cases for dbe_data_send, please refer to RDBE Data Transmission Use Cases memo

dbe_dc_cfg – Set the digital down-converters configuration

Command: → dbe_dc_cfg = < DC > : < rate > : <LO freq> : <ts>;

← !dbe_ dc_cfg = <return code>;

Query: → dbe_ dc_cfg?;

← !dbe_ dc_cfg? <return code>: < DC0 rate > : <DC0 LO> : ... :< DC7 rate > : <DC7LO > ;

Purpose: Provide an interface to the configurable portions of each down-converter; decimation rate and LO frequency can both be adjusted.

Settable Parameters:

Parameter	Type	Allowed Values	Defaults	Comments
DC	int	0-7	-	Select down converter to be configured
rate	int	4-2048	-	Decimation value for selected down-converter (see note 2)
LO freq	float	0.06-128.0	-	LO frequency for selected down-converter (Mhz)
t_s	time	YYYYDOYHHMMSS	-	integer second at which the LO frequency will take affect (see note 3)

Monitor Only Parameters:

Parameter	Type	Values	Comments
DCX rate	char	4-2048	Decimation rate for selected down-converter
DCX LO	time	0.06-128.0	LO frequency for selected down-converter (MHz)

Notes:

1. After setting the decimation rate, there will be a delay before valid data are again present at the filter output. This delay includes two clock cycles of 256MHz for the interface state machine, plus a larger delay due to the delta in decimation values.
2. In version 0.5, all down-converters must be set to the same decimation value.
3. The config command will be executed on the 1pps boundary

dbe_ddc_quantize – Manage the digital down-converters quantization

Command: → dbe_ddc_quantize = [<channel>] : <+thresh> : <zthresh> : <-thresh >;

← !dbe_ dc_cfg = <return code>;

Query: → dbe_ dc_cfg?;

← !dbe_ dc_cfg? <return code>: [<channel>] : <+thresh> : <zthresh> : <-thresh >;

Purpose: Provide an interface to set the quantization levels of the digital down converter.

Settable Parameters:

Parameter	Type	Allowed Values	Defaults	Comments
channel	int	0-7	0	Select down converter to be configured
+thresh	int		120	The plus threshold value
zthresh	int		0	The zero threshold value
-thresh	int		-120	The minus threshold value

Monitor Only Parameters:

Parameter	Type	Values	Comments
channel	int	0-7	The down converter channel threshold settings
+thresh	int		The plus threshold value
zthresh	int		The zero threshold value
-thresh	int		The minus threshold value

Notes:

1. After setting the thresholds, there will be a delay before valid data is again present at the filter output. This delay includes two clock cycles of 256MHz for the interface state machine.

2. The DDC personality handles quantization different then the PFBG personality and hence a separate command for setting the values.

dbe_dot – Get the Data Observable Time (DOT) clock information (query only)

Query: → dbe_dot?;

← !dbe_dot ? <return code> : <current DOT reading> : <sync status> : <current OS time> : <DOT-OS difference> : [< Actual DOT Time>];

Purpose: Get the DOT clock information with respect to the OS time.

Monitor Only Parameters:

Parameter	Type	Values	Comments
current DOT reading	time	YYYYDDDHHMMSS	Current value of DOT clock. (see note 2)
sync status	char	not_synced | syncerr_eq_0| syncerr_le_3 | syncerr_gt_3	'not_synced' – DOT 1pps generator has not yet been sync'ed. (see note 3) 'syncerr_eq_0' - DOT 1pps integer value is the same as the OS time. 'syncerr_le_3' – DOT 1pps integer value within +/-2 integer seconds from OS time 'syncerr_gt_3' – DOT 1pps integer value more +/-3 integer seconds from OS time
current OS time	time		Corresponding OS time
DOT-OS difference	time		<current DOT reading> minus <current OS time>
Actual DOT Time	time		Current time inserted in VLBI Header (see note 4)

Notes:

1. If you are looking from differences to the incoming 1pps in clock cycles it has moved to the dbe_fs command.

2. This is the DOT clock time converted to the standard time format YYYYDDDHHMMSS.

3. A <sync status> of 'not_synced' indicates that the DOT hardware 1pps generator has not been sync'ed to a 1pps tick. All other <sync status> returns indicate that the DOT 1pps generator has been sync'ed with a 'DOT_set' command.

4. The current DOT clock time depends on the packet format, e.g. Mark5B or VDIF, specified within the FPGA personality. If the packet format is Mark5B emulation mode, then the DOT clock reference is the VLBA BCD Time Code ('JJJSSSSS'), with the MSB representing seconds 10000's digit and the LSB seconds 1's digit. This is the time interval elapsed since the beginning of the given MJD. For VDIF frames it will be the seconds from the reference epoch.

dbe_dot_inc – Increment the DOT clock time on the next 1pps tick

Command: → dbe_dot_inc = <inc > ;

← !dbe_dot_inc = <return code>;

Query: → dbe_dot_inc?;

← !dbe_dot_inc? <return code>: <inc> ;

Purpose: Increment or decrement the DOT clock time by a specified number of seconds

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
inc	int	0 > X > 0	1	Number of integer seconds to increment DOT clock (may be positive or negative). >0 will advance the DOT clock setting; <0 will retard the DOT clock setting

Monitor Only Parameters:

Parameter	Type	Values	Comments
inc	int		The last setting made using dbe_dot_inc command

Notes:

1. The dbe_dot_inc command should be used to adjust an error in the DOT clock only after the DOT clock time has been initialized using “dbe_execute=init” and set with the ‘dbe_dot_set’ command

dbe_dot_set – Set the DOT clock time on next 1pps tick

Command: → dbe_dot_set = <time > : [<option>];

← !dbe_dot_set = <return code>;

Query: → dbe_dot_set?;

← !dbe_dot_set? <return code>: <time> : <time offset> ;

Purpose: Set the initial value of the RDBE's DOT clock on the next tick of the selected 1pps source.

Settable Parameters:

Parameter	Type	Allowed Values	Defaults	Comments
time[1]	time	YYYYDDDHHMMSS	null	The DOT clock can only be set to an integer second value. If null, the dot clock is set according to the current OS time for the appropriate data format, e.g. Mark5B emulation mode, VDIF.
option	char	null | force	null	If "force", 1pps generator will be re-synced even though the DOT_synced status indicates it is not already sync'ed

Monitor Only Parameters:

Parameter	Type	Values	Comments
time	time		Includes all inferred higher order time that was not given explicitly in the command value.
time offset	time		estimated interval between DOT time and OS time

Notes:

1. If <time> does not specify higher-order time (i.e. year, day, etc), the current values from the OS time are used. The time format is YYYYDDDHHMMSS and will be converted into the appropriate format for hardware based on the data format. Since the RDBE application keeps higher-order time (above 1sec) in software, higher-order time will be lost if the rdbe_server application is restarted or if the system is rebooted. The DOT hardware 1pps generator will lose sync if re-initialized with the dbe_execute=init command, and will require a re-issued 'dbe_dot_set' command to set higher-order time. After the 'dbe_dot_set' command is issued, a 'dbe_dot?' should be issued to verify the time setting. If necessary, the 'dbe_dot_inc' command may be used to adjust the DOT clock to the correct second. If the DOT clock is set from OS time, care must be taken that the OS clock is reasonably well aligned with the external 1pps tick (to within a few tens of milliseconds, at worst). Otherwise, there is no requirement on the OS timekeeping required. Normally the dbe_dot_set operation is only done once at the beginning of an experiment.

dbe_execute – Execute a specific system level command

Command: → dbe_execute = <cmd>;

← !dbe_execute = <return code>;

Purpose: Provides the capability to execute a finite number of commands via the rdbe_server application.

Settable Parameters:

Parameter	Type	Allowed Values	Comments
cmd	ASCII	reboot | init	Reboot the RDBE system Initialize the RDBE's FPGA personality to a known state (see note 1).

Notes:

The 'init' cmd (command) causes the RDBE application to execute a known set of low level commands, specified by the FPGA developer, to initialize the personality into a known state.

dbe_fs – Resynchronize the 1pps and 5MHz signal on frequency synthesizer board

Command: → dbe_fs = <pps_sync> : <pll_reset>;

← !dbe_fs = <return code>;

Query: → dbe_fs?;

← !dbe_fs? <return code>: <1pps_status> : <pll_status> : <pps_sync> ;

Purpose: This command returns the status of the 1pps and 5MHz signal input to the frequency synthesizer board and allows for the remote resynchronization of the clocks on the synthesizer board to the sources.

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
pps_sync	ASCII	on | off	off	Resynchronize to the station 1pps
pll_reset	ASCII	on | off	off	Resynchronize the phase lock loop to the incoming 5MHz signal

Monitor Parameters:

Parameter	Type	Values	Comments
1pps_status	ASCII	1/0	Station 1PPS is present (1) / or not (0)
pll_lock	int	1/0	Indicates that the PLL on the frequency synthesizer is locked to the incoming 5MHz signal (1)/ or not (0)
pps_sync	int		The delta, in 1024MHz clock cycles, between the incoming 1pps signal and the frequency synthesizer output 1pps signal

dbe_hw_version – Get the hardware version numbers of the RDBE (query only)

Query: → dbe_hw_version?;

← !dbe_hw_version ? <return code>: <ROACH board version> : <Timing board version> : <ALC board version>;

Purpose: Returns the version of hardware boards in the RDBE systen.

Monitor Only Parameters:

Parameter	Type	Values	Comments
Roach Board version	ASCII		The version number of the ROACH board
Timing board version	ASCII		The version number of the frequency synthesizer board
ALC board version	ASCII		The version of the ALC board

dbe_ifconfig – Set RDBE 10G network interface configuration

Command: → dbe_ifconfig = <state> : <MTU> : <mode> : <IP address> : [<GW address>];

← !dbe_ifconfig = <return code>;

Query: → dbe_ifconfig?;

← !dbe_ifconfig ? <return code>: <state> : <MTU> : <mode > : <IP address> : < MAC addr > : <TX Packets> : <TX Errors> : <TX Dropped> : <TX_Overrun> : <txqueuelen> ;

Purpose: Set interface mode, enable or get status of the specific FPGA 10G network interface.

Settable Parameters:

Parameter	Type	Allowed Values	Defaults	Comments
state	char	up | down	up	Causes the interface to be enabled Causes the interface to be disabled
MTU	int	64 ≤ mtu ≤ 9000	9000	Maximum packet size to accept, default = 9000
mode	int	2 | 4	4	Layer 2 transmission (no UDP/IP headers) Layer 4 transmission (standard UDP/IP header used)
IP address	ASCII	xx.xx.xx.xx	-	IPv4 address to be assigned to this interface, where 0 <= xx <= 255
GW address	ASCII	xx.xx.xx.xx	-	The IPv4 address of the gateway, used for routing to external sources. 0 <= xx <= 255

Monitor Only Parameters:

Parameter	Type	Allowed Values	Comments
state	char	NA	The interface status (up/down)
MTU	int	NA	The maximum packet size the interface will send
mode	int	NA	Layer 2 or Layer 4 transmission
IP address	ASCII	NA	If mode is 4, then the IP address assigned to the interface, ignored mode 2
MAC addr[1]	ASCII	NA	MAC address of 10G core interface separated by "." instead of normal ":" (see note 2)
TX Packets	int	NA	The number of packets transmitted
TX Errors	int	NA	The number of transmission errors detected
TX dropped	int	NA	The number of packets dropped

Parameter	Type	Allowed Values	Comments
TX Overrun	int	NA	The numbers of packets dropped due to the tx queue length overflowing
txqueulen	int	NA	The transmission queue length

Notes:

1. There are four 10G CX4 interfaces available to the RDBE, and at the present time the assigned interface is set in the FPGA. Later versions of FPGA code will require the capability to set specific interfaces, and this command will be modified.

2. The MAC address can be assigned either using ‘dbe_execute = init’ or the dbe_MAC command. If the MAC address of ff.ff.ff.ff.ff.ff is returned, the source MAC address has not been configured. Please see the dbe_MAC command.

dbe_ioch_assign – Set the input and output channel assignment for the VLBI payload

Command: → dbe_ioch_assign = <input>:<channel(s)>:[<threadID>]:[<input>]:[<channel(s)>]:[<threadID>]……;

← !dbe_ ioch_assign = <return code>;

Query: → dbe_ ioch_assign?;

← !dbe_ ioch_assign? <return code>: <input>:<channel(s)>:[<threadID>]:[<input>]:[<channel(s)>]:[<threadID>]……;

Purpose: Assign the VLBI payload input and channel order

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
input	int	m > 0	0	Input assignment (or IF number) the channel will be selected from
channel(s)	int	m | m – n	-	Individual channel, or range of channels, from the specified input (see note 1, 2)
threadID	int	z >=0	-	Used only when the data format is VDIF. If the format is Mark5B emulation mode, then the threadID is ignored (see note 3)

Monitor Only Parameters:

Parameter	Type	Allowed Values	Comments
Input	int		Input assignment (or IF number) the channel will be selected from
channel(s)	int		Individual channel, or range of channels, from the specified input
threadID	int		Used only when the data format is VDIF. If the format is Mark5B emulation, mode then it is ignored

Notes:

1. A range of values will be allowed for the channels with the value 'm' holding the lower significant position and the value 'n' holding the highest significant position for that input assignment. For example dbe_ioch_assign=0:0-15 assigns 16 channels with input channel 0 being assigned in order to the VLBI Payload.
2. The channel ordering in the VLBI payload is directly related to the assignment combination of input and channel specified in this command. The present geodetic personality has 0:1: : 1:1: :0:3::1:3:…0:15::1:15; with the first input / channel combination (0:1) being assigned to the least significant bit position (bits 0 and 1) in the VLBI Payload and (1:15: ;) being assigned to the most significant bit position (bits 30 and 31).
3. The theadIDs assignments are part of the VDIF standard. For details, please visit the http://vlbi.org/vdif. If no threadID's are assigned in the command, an ordered output will be assigned by the personality, starting at threadID 0, 1, etc,

dbe_mac – set / get the 10G core source MAC address

Command: → dbe_mac = <MAC> ;

← !dbe_ mac = <return code>;

Query: → dbe_mac?;

← !dbe_ mac? <return code>: <MAC>;

Purpose: Assign a source MAC address to the 10G Ethernet Core used for data transmission

Settable Parameters:

Parameter	Type	Allowed Values	Default	Comments
MAC	ASCII	xx.xx.xx.xx.xx.xx	(see note 1)	Source MAC address to be used in data transmission

Monitor Only Parameters:

Parameter	Type	Allowed Values	Comments
Input	ASCII		Source MAC address to be used in data transmission

Notes:

1. By default, the default value is dependent on the personality loaded. For the pfbg version 1.4 it is ff.ff.ff.ff.ff.ff (which is not a valid MAC address).

dbe_packet – Get the packet structure (query only)

Query: → dbe_packet?;

← !dbe_packet ? <return code>: <DPOFST> : <DFOFST > : <length>: <PSN Mode>: <PSNOFST>;

Purpose: Get the structure of the transmitted packet that includes the byte offsets, after the MAC header, and the length of the VLBI data and its characteristics.

Monitor Only Parameters:

Parameter	Type	Allowed Values	Comments
DPOFST	int	>= 0	payload byte offset from after the MAC header to the beginning of the VLBI payload (see note 1)
DFOFST	int	>= 0	payload byte offset to beginning of recording
Length	int	>= 0	number of bytes to record per packet (note 2)
PSN Mode	int	0 | 1 | 2	Packet Serial Number (PSN) monitor mode (note 3)
PSNOFST	int	>= 0	payload byte offset from beginning of payload to PSN (PSN = 1 | 2)

Notes:

1. The VLBI data payload is defined as to be the VLBI data plus the header, and if a PSN is pre-pended to include this byte field.
2. The length of the VLBI data payload, including the VLBI data header, to be recorded must be a multiple of 8 bytes
3. PSN-Monitor mode 0 will cause the PSN not to be written into the data. PSN Monitor mode 1 inserts the PSN at a byte position offset PSNOFST. The primary use of PSN Mode 1 is for Mark5B emulation mode so the hardware correlator can process the disk modules. PSN-monitor mode 2 marks packets as invalid so that the destination does not record the packet when the most significant bit of the PSN is set, when the MSB is not set, the data is stored as described by PSN Mode 1.

dbe_personality – Set the RDBE FPGA bit code personality

Command: → dbe_personality = <type> : [<personality>] ;

← !dbe_personality = <return code>;

Query: → dbe_personality?;

← !dbe_personality ? <return code>: <type> : <personality> : <version> : <status>;

Purpose: Specify the FPGA bit code personality to be loaded.

Settable Parameters:

Parameter	Type	Allowed Values	Comments
type	char	DDC | PFBG | PFBA	Digital Down Converter bit code type PFB Geodetic bit code type PFB Astronomy bit code type
personality	char		file and system path (see note 1)

Monitor Only Parameters

Parameter	type	Values	Comments
type	char		Represents the type of FPGA bit code loaded
personality	char		File name of bit code loaded, which includes the version of bit code
version[2]	char		This is the version information stored within the FPGA registers (see note 2)
status	char	loaded | error | not loaded	Status of the personality (see note 3)

Notes:

1. There will be a default location and file name of the bit code load. If there is another location that the bit code should be loaded from it should be specified here, along with the filename. The filename should be of the form INST_type_characteristics_MajVersNumber_MinorVersionNumber, e.g. HAY_PFBG_5B2G_1_0.bin. This would correspond to a Haystack Poly phase filter bank design with Mark5B emulation mode at 2Gbps version 1.0.
2. The filename of the binary does contain version information, but since filenames can be renamed, the bit code contains registers with the appropriate personality major and minor version that can be read.
3. A feature of the system is that on reboot the last personality loaded will be indicated and therefore the status of "not loaded".

dbe_personality

dbe_quantize – Set channel quantization

Command: → dbe_quantize = <state> ;

← !dbe_quantize = <return code>;

Query: → dbe_quantize?: [<input>]:[<chid>];

← !dbe_quantize ? <return code>: <input> : <channel > : <+thresh>:<-thresh>:<gain settings>: <state count>;

Purpose: Initiate threshold settings (2 bit quantization) for PFB personality.

Settable Parameters:

Parameter	Type	Allowed Values	Comments
state	char	hold_set | reset	Use the existing channel quantization values as the initial starting values for the quantization algorithm to re-quantize the PFB personality Replaces the dbe_execute = quantize command, and uses 0.5 as the initial value for the quantization algorithm

Monitor Only Parameters:

Parameter	Type	Allowed values	Comments
input	int	0/1	INPUT 0 or INPUT1
chid	int	0-15	Specific channel id that the quantize values are given
+thresh	float		2 bit quantization positive threshold
-thresh	float		2 bit quantization negative threshold
gain settings	fixed		2 bit quantization gain settings
state count	int		2 bit quantization state count

Notes:

1. The command is to be used with the PFBX type personalities only.
2. For the DDC personlity use the dbe_ddc_quantize command.

dbe_status – Get system status (query only)

Query: → dbe_status?;

← !dbe_status ? <return code>: <status word> ;

Purpose: Get general system status

Monitor Only Parameters:

Parameter	type	Bit Value	Comments
Status word	hex	0x0001	Bit 0 – system ready Bit 1 – error messages pending Bit 2 – system command has control of FPGA Bit 3- 'delayed completion' commands pending Bit 4- 'delayed completion' queries pending Bit 5- Bit 6- Transmitting valid data Bit 7- Transmitting data marked invalid Bit 8 – FPGA state: 0 - FPGA not loaded, 1 – FPGA loaded

dbe_sw_version – Get the RDBE command and control software version numbers (query only)

Query: → dbe_sw_version?;

← !dbe_sw_version ? <return code>: <RDBE application version> : <HAL library version> : [<OS / kernel version>];

Purpose: Determine the version of software application and supporting software running on the RDBE.

Monitor Only Parameters:

Parameter	type	Comments
RDBE application version	ASCII	The version number of the RDBE command and control application
HAL library version	ASCII	The version of the HAL library the RDBE application was compiled against
OS/kernel version	ASCII	Distribution and kernel version of Linux running on the RDBE

dbe_tsys_diode_ctl – Set Tsys diode control settings

Command: → dbe_tsys_diode_ctl = <diode_rate> : <blank_time> ;

← !dbe_tsys_diode_ctl = <return code>;

Query: → dbe_tsys_diode_ctl?;

← !dbe_tsys_diode_ctl? <return code>: <diode_rate> : <blank_time> : <accum_cnt>;

Purpose: To configure the Tsys diode control.

Settable Parameters:

Parameter	Type	Allowed values	Default	Comments
diode_rate	int	10-100	80	Diode switch rate in Hertz
blank_time	int	-	200	Time in microsecs, based on a 64MHz clock (see note 1)

Monitor Parameters:

Parameter	Type	Allowed values	Comments
diode_rate	int	1-100	Diode switch rate in Hertz
blank_time	int		Time in microsecs, based on a 64MHz clock
accum_cnt	int	>1	number of accumulations processed

Notes:

1. The blanking time should never exceed 50% of the diode period "on" time

dbe_tsys_mon – Set Tsys monitoring broadcast settings

Command: → dbe_tsys_mon = <state>: <Multicast IP address>:[<port>]:[<interval>];

← !dbe_tsys_mon = <return code>;

Query: → dbe_tsys_mon?;

← !dbe_tsys_mon ? <return code>: <state> :<Multicast IP address>:<port>:<interval>;

Purpose: To enable or disable the broadcast of the Tsys data from the FPGA over the standard 1G Ethernet interface.

Settable Parameters:

Parameter	Type	Allowed values	Default	Comments
state	ASCII	enable/disable	disable	Enabling / disabling of feature
Multicast IP Addr	ASCII	xx.xx.xx.xx		Multicast IP address that the 1pps port will be broadcast on (see note 1)
port	int	> 2000	20040	UDP destination port of broadcast
interval	int	>0	6	Interval at which the cumulative data will be broadcast in seconds

Monitor Parameters:

Parameter	Type	Allowed values	Comments
state	ASCII	enabled / disabled	enabled – the destination IP address assignment is being utilized disabled – no destination IP address is assigned if data are being transmitted
IP	ASCII	xx.xx.xx.xx	Destination IPv4 address
port	int	> 2000	UDP destination port of broadcast
interval	int	>0	Interval at which the cumulative data will be broadcast in seconds

Notes:

1) The multicast IP address must be in the range 239.0.1.0-239.255.255.255

2) The broadcast packet will start with 32 bit integration interval in seconds, followed by the 32 bit time value associated with the mid-point of the integration time for that interval. The 2nd thru 34th word will contain 8 bits for IF identification {0/1}, 8 bits for channel identification {0-15}, followed by tsys on data sqrt(V_{on}^2) averaged over the integration interval (48bits), tsys off data sgrt(V_{off}^2) averaged over the integration interval (48bits).

dbe_xbar – set / get DDC crossbar switch positions[1]

Command: → dbe_xbar = <DC0 source>:<DC1 source>: … :<DC6 source>:<DC7 source>;

← !dbe_ xbar = <return code>;

Query: → dbe_ xbar?;

← !dbe_ xbar ? <return code>: <DC0 source>:<DC1 source>: … :<DC6 source>:<DC7 source>;

Purpose: To control the DDC crossbar switch, which allows any sub-band from either INPUT to be connected to any of the eight down-converters?

Settable Parameters:

Parameter	Type	Allowed values	Default	Comments
DC0 source	integer	0-7	-	Source for down-converter 0
DC1 source	integer	0-7	-	Source for down-converter 1
DC2 source	integer	0-7	-	Source for down-converter 2
DC3 source	integer	0-7	-	Source for down-converter 3
DC4 source	integer	0-7	-	Source for down-converter 4
DC5 source	integer	0-7	-	Source for down-converter 5
DC6 source	integer	0-7	-	Source for down-converter 6
DC7 source	integer	0-7	-	Source for down-converter 7

Monitor Parameters:

DC0 source	integer	0-7	Source for down-converter 0
DC1 source	integer	0-7	Source for down-converter 1
DC2 source	integer	0-7	Source for down-converter 2
DC3 source	integer	0-7	Source for down-converter 3
DC4 source	integer	0-7	Source for down-converter 4
DC5 source	integer	0-7	Source for down-converter 5
DC6 source	integer	0-7	Source for down-converter 6
DC7 source	integer	0-7	Source for down-converter 7

Notes: 1. Each Down Converter can only be connected to one source (sub-band).