VRM2WB Command Reference v29Mar2015

Table of Contents

VRM2WB Mainframe Commands

baud	Serial port baud rate
block	Mainframe frequency block
desc	Mainframe description string
detect	Mainframe module detection status
hwstat	Hardware status
id	Mainframe id string
lcdbl	LCD backlight brightness
lockstate	Front panel lock status
phantpwr	Antenna phantom power
portctl	Communication port control flags
serial	Mainframe serial number
version	Mainframe firmware version

Termination: all commands are terminated with an ASCII **carriage return** character (hex code 0x0D), represented by **<CR>** in the examples below. All responses are terminated with an ASCII **carriage return, line feed** pair (hex codes 0x0D, 0x0A), represented by **<CRLF>** in the examples below. An ellipsis (**...**) represents members of an array that have been omitted from an example for the sake of brevity.

Verbose response: commands prefixed with an exclamation point (bang) character result in a "verbose" response containing both the name of the property being addressed and its current value (if any). The verbose response returns the property/value pair in the "assignment" form, for example **`OK ingn(2)=40 <CRLF>`**. This supports certain 3rd party control programming styles where the response needs to be self-describing.

baud (Serial port baud rate)

This command may be used as a query to determine the baud rate setting for the serial port, or as an update to set it. The data is an integer type. The following values are allowed:

- 9600
- 19200
- 38400
- 57600

- 115200

Examples:

	REQUEST	RESPONSE
QUERY	**baud?<CR>**	**OK 57600<CRLF>**
UPDATE	**baud=57600<CR>**	**OK<CRLF>**

block (Mainframe frequency block)

This command may be used as a query to read the mainframe frequency block. The data type is integer and is a code that specifies the frequency block for the mainframe. It may be in the range 0 to 2, with the following meanings:

Code	Frequency Block
0	Low
1	Mid
2	High

Examples:

	REQUEST	RESPONSE
QUERY	**rxblock?<CR>**	**OK 0<CRLF>**

desc (Mainframe description string)

This command may be used as a query to read the user defined mainframe description, or as an update to set it. The data is a string type, with a limit of 30 characters.

Note: String arguments in commands need to be passed in **quoted** form, contained in a pair of **double-quote** (") characters. A problem arises when using the `desc` command to read or write a string that already contains double-quote characters, for example: `The "Hula" Room`. The solution is to **escape** the double quotes within `The "Hula" Room` so that it can be passed as a string argument for the `desc` command. This is done by preceding the double-quote characters with a **backslash** character like this: `The \"Hula\" Room`. Now it can be passed as a string argument to the `desc` command: `desc="The \"Hula\" Room"`. Since the **backslash** serves as the escape character in quoted-string arguments, it too must be escaped if it is part of the string, so `"foo\bar"` would become `"foo\\bar"`. If necessary, any character, printable or non-

printable, can be represented in the hexadecimal escaped form `\xHH` where `HH` is any 2-digit hexadecimal number. The special escaped character forms `\r` (carriage return), `\n` (new line) and `\t` (tab) are also recognized.

Examples:

	REQUEST	RESPONSE
QUERY	**desc?<CR>**	**OK "Aloha Room East"<CRLF>**
UPDATE	**desc="Studio #2"<CR>**	**OK<CRLF>**

detect (Mainframe module detection status)

This command may be used as a query to read the mainframe module detection status, or as an update to set it. The data type is integer, either "1" meaning that the mainframe is actively detecting modules, or "0" meaning that it is not. If set to 1 normal operation is suspended until the detection process is complete.

Examples:

	REQUEST	RESPONSE
QUERY	**detect?<CR>**	**OK 0<CRLF>**
UPDATE	**detect=1<CR>**	**OK<CRLF>**

hwstat (Hardware status)

This command may be used as a query to determine the status of device hardware. The data type is integer, and the value is a code representing one of the following states:

- **0** means normal operation
- **1** means that an antenna phantom power short has been detected

Example:

	REQUEST	RESPONSE
QUERY	**hwstat?<CR>**	**OK 0<CRLF>**

id (Mainframe ID string)

This command may be used as a query to read the mainframe id string. This is the "name" of the device used by the control protocol and is always "DR". The data is a string type.

Example:

	REQUEST	RESPONSE
QUERY	**id?<CR>**	**OK "DR"<CRLF>**

lcdbl (LCD backlight brightness)

This command may be used as a query to read the front panel LCD backlight brightness level, or as an update to set it. The data type is integer, in the range 1 to 4, where 1 is the minimum brightness and 4 the maximum brightness.

Examples:

	REQUEST	RESPONSE
QUERY	**lcdbl?<CR>**	**OK 1<CRLF>**
UPDATE	**lcdbl=0<CR>**	**OK<CRLF>**

lockstate (Front panel lock status)

This command may be used as a query to read the front panel status, or as an update to set it. The data type is integer, either 1 or 0, where 1 is "locked" and 0 is "not locked".

Examples:

	REQUEST	RESPONSE
QUERY	**lockstate?<CR>**	**OK 1<CRLF>**
UPDATE	**lockstate=0<CR>**	**OK<CRLF>**

phantpwr (Antenna phantom power)

This command may be used as a query to read the antenna phantom power status, or as an update to set it. The data type is integer and is a code that specifies the phantom power configuration. It may be in the range 0 to 3, with the following meanings:

Code	Phantom Power Configuration
0	No phantom power
1	Phantom power on antenna A only
2	Phantom power on antenna B only
3	Phantom power on both antenna A and B

Examples:

	REQUEST	RESPONSE
QUERY	**phantpwr?<CR>**	**OK 0<CRLF>**
UPDATE	**phantpwr=3<CR>**	**OK<CRLF>**

portctl (communication port control flags)

This command may be used as a query to read the port control flags, or as an update to set them. The communication port is specified by using the address syntax. Addresses must be in the range 0 to 4, representing the one of the following:

- **0** - USB port
- **1** - RS232 port
- **2** - TCP port 1
- **3** - TCP port 2
- **4** - HTTP port

The data type is integer, in the range 0 to 3. The value is a code representing the control settings for the communication port:

Code	Port Setting
0	Port is disabled
1	Port is receive only
2	Port is send only
3	Port is send/receive

If the port address is wildcarded, then the data type is an array of integer of size 4. By default, all communication ports are send/receive (full duplex) but in some advanced 3rd party control scenarios a port may need to be set otherwise. **Note: to preserve the ability to communicate with the device, changes to the USB port setting have no effect, the default of send/receive is always in force.**

Examples:

	REQUEST	RESPONSE
QUERY	**portctl(1)?<CR>**	**OK 3<CRLF>**
QUERY	**portctl(*)?<CR>**	**OK {3,3,3,3}<CRLF>**
UPDATE	**portctl(1)=2<CR>**	**OK<CRLF>**
UPDATE	**portctl(*)={3,3,3,3}<CR>**	**OK<CRLF>**

serial (Mainframe serial number)

This command may be used as a query to read the mainframe's serial number. The data is a string type.

Example:

	REQUEST	RESPONSE
QUERY	**serial?<CR>**	**OK "6100101"<CRLF>**

version (Mainframe firmware version)

This command may be used as a query to read the mainframe's firmware version number. The data is a string type.

Example:

	REQUEST	RESPONSE
QUERY	**version?<CR>**	**OK "1.0.1"<CRLF>**

VRM2WB Channel Receiver Commands

rxalevel	Receiver audio attenuator level
rxameter	Receiver audio level meter
rxblock	Receiver frequency block
rxcompat	Receiver compatibility mode
rxdivmode	Receiver diversity reception mode
rxfreq	Receiver frequency
rxid	Receiver ID
rxlink	Receiver link status
rxmute	Receiver mute status
rxmutetog	Receiver mute toggle
rxname	Receiver name
rxphase	Receiver audio phase
rxpresent	Receiver present status
rxpwr	Receiver power on/off
rxrmeter	Receiver RF level meter
rxscan	Receiver scan state
rxsmartnr	Receiver smart noise reduction
rxsquelch	Receiver squelch status
rxstat	Receiver status
rxtben	Receiver talkback enable
rxtboutch	Receiver talkback output channel
rxtone	Receiver output test tone
rxver	Receiver firmware version

Termination: all commands are terminated with an ASCII **carriage return** character (hex code 0x0D), represented by **<CR>** in the examples below. All responses are terminated with an ASCII **carriage return, line feed** pair (hex codes 0x0D, 0x0A), represented by **<CRLF>** in the examples below. An ellipsis (**...**) represents members of an array that have been omitted from an example for the sake of brevity.

Verbose response: commands prefixed with an exclamation point (bang) character result in a "verbose" response containing both the name of the property being addressed and its current value (if any). The verbose response returns the property/value pair in the "assignment" form, for example **OK rxphase(2)=0 <CRLF>**. This supports certain 3rd party control programming styles where the response needs to be self-describing.

rxalevel (Receiver audio attenuator level)

This command may be used as a query to read the receiver output level setting, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, representing the attenuation (or gain) in dB. The range is -35 to +8 dB. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **99** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxalevel(1)?<CR>**	**OK rxalevel(1)=-10<CRLF>**
QUERY	**!rxalevel(*)?<CR>**	**OK** rxalevel(*)={0,0,0,0,0,0}<CRLF>
UPDATE	**!rxalevel(5)=0<CR>**	**OK rxalevel(5)=0<CRLF>**
UPDATE	**!rxalevel(*)={0,0,-3, 0,99,99}<CR>**	**OK rxalevel(*)={0,0,-3,0,0,0}<CRLF>**

rxameter (Receiver audio level meter)

This command may be used as a query to read the receiver audio level meter. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, in the range -80 to +5, representing the audio level in dB, referenced to the transmitter's threshold of limiting. This is the raw level *before* application of the receiver's audio level setting (rxalevel), which controls the rear panel audio output level (if rxameter = 0dB and rxalevel = 0dB then the output level is 0dBu). If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxameter(1)?<CR>**	**OK rxameter(1)=-23<CRLF>**
QUERY	**!rxameter(*)?<CR>**	**OK rxameter(*)={-2,4,-10,-53,-71,-95}<CRLF>**

rxblock (Receiver frequency block)

This command may be used as a query to read the receiver frequency block. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is string.

	REQUEST	RESPONSE
QUERY	`!rxblock(1)?<CR>`	`OK rxblock(1)="A1"<CRLF>`

rxcompat (Receiver compatibility mode)

This command may be used as a query to read the compatibility mode, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **99** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command. The data type is integer, and is a code representing one of the following:

Code	Compatibility Mode
0	None (DSP Passthrough mode for testing)
1	Lectrosonics 100 Series
2	Lectrosonics 200 Series
3	Mode 3 (contact factory for details)
4	Lectrosonics 400 Series (Digital Hybrid - US and Canada)
5	Lectrosonics IFB Series
6	Mode 6 (contact factory for details)
7	Mode 7 (contact factory for details)
8	Lectrosonics 300 Series (European Union)
9	Lectrosonics 400 Series (Digital Hybrid - European Union)

Examples:

	REQUEST	RESPONSE

QUERY	**!rxcompat(3)?<CR>**	**OK rxcompat (3)=4<CRLF>**
QUERY	**!rxcompat (*)?<CR>**	**OK rxcompat (*)={4,4,4,4,4,4}<CRLF>**
UPDATE	**!rxcompat (3)=2<CR>**	**OK rxcompat (3)=2<CRLF>**
UPDATE	**!rxcompat (*)={4,4,2,99,99,99}<CR>**	**OK rxcompat (*)={4,4,2,4,4,4}<CRLF>**

rxdivmode (Receiver diversity reception mode)

This command may be used as a query to read the receiver diversity reception mode, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6, or the wildcard address. If the channel address is wildcarded, then the data type is an array of integer of size 3, and the data pertains to *diversity pairs*, not individual receiver modules. The data type is integer, either "1" meaning that the diversity pair is active, or "0" meaning that it is not. Diversity pairs consist of adjacent receiver modules, so channels 1 and 2 form the first pair, 2 and 3 the second pair, and 3 and 4 the third pair.

Note: An update will return an ERROR response if the addressed receiver module is not present, or if the members of the diversity pair it belongs to are not on the same frequency block.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxdivmode(3)?<CR>**	**OK rxdivmode(3)=1<CRLF>**
QUERY	**!rxdivmode(*)?<CR>**	**OK rxdivmode(*)={0,0,1}<CRLF>**
UPDATE	**!rxdivmode(2)=0<CR>**	**OK rxdivmode(2)=0<CRLF>**
UPDATE	**!rxdivmode(*)={0,0,0}<CR>**	**OK rxdivmode(*)={0,0,0}<CRLF>**

rxfreq (Receiver frequency)

This command may be used as a query to read the receiver frequency, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, representing the frequency in kHz. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **999999** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command.

Examples:

	REQUEST	RESPONSE

QUERY	**!rxfreq(1)?<CR>**	**OK rxfreq(1)=471200<CRLF>**
QUERY	**!rxfreq(*)?<CR>**	**OK rxfreq(*)={471200,520900,...,540000}<CRLF>**
UPDATE	**!rxfreq(1)=471200<CR>**	**OK rxfreq(1)=471.200<CRLF>**
UPDATE	**!rxfreq(*)={471200,520900,...,999999}<CR>**	**OK rxfreq(*)={471200,520900,...,540000}<CRLF>**

rxid (Receiver ID)

This command may be used as a query to read the receiver's ID string. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data is a string type.

Example:

	REQUEST	RESPONSE
QUERY	**!rxid(1)?<CR>**	**OK rxid(1)="DRM"<CRLF>**

rxlink (Receiver link status)

This command may be used as a query to read the link status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that a valid signal has been acquired, or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxlink(3)?<CR>**	**OK rxlink(3)=0<CRLF>**
QUERY	**!rxlink(*)?<CR>**	**OK rxlink(*)={0,1,0,0,0,0}<CRLF>**

rxmute (Receiver mute status)

This command may be used as a query to read the receiver mute status, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the output is muted, or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **99** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxmute(3)?<CR>**	**OK rxmute(3)=1<CRLF>**
QUERY	**!rxmute(*)?<CR>**	**OK rxmute(*)={0,0,0,0,1,0}<CRLF>**
UPDATE	**!rxmute(2)=0<CR>**	**OK rxmute(2)=0<CRLF>**
UPDATE	**!rxmute(*)={0,1,0,99,99,99}<CR>**	**OK rxmute(*)={0,1,0,0,0,0}<CRLF>**

rxmutetog (Receiver mute toggle)

This command may be used as a simple comand to toggle the receiver mute status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The response to a verbose command is the new mute status for that receiver.

Examples:

	REQUEST	RESPONSE
COMMAND	**!rxmutetog(4)<CR>**	**OK rxmute(4)=1<CRLF>**

rxname (Receiver name)

This command may be used as a query to read the receiver name, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is string, with a limit of 15 characters.

Note: String arguments in commands need to be passed in **quoted** form, contained in a pair of **double-quote** (") characters. A problem arises when using the `inlb` command to read or write a string that already contains double-quote characters, for example: `The "Hula" Room`. The solution is to **escape** the double quotes within `The "Hula" Room` so that it can be passed as a string argument for the `inlb` command. This is done by preceding the double-quote characters with a **backslash** character like this: `The \"Hula\" Room`. Now it can be passed as a string argument to the `inlb` command: `inlb(1)="The \"Hula\" Room"`. Since the **backslash** serves

as the escape character in quoted-string arguments, it too must be escaped if it is part of the string, so `"foo\bar"` would become `"foo\\bar"`. If necessary, any character, printable or non-printable, can be represented in the hexadecimal escaped form `\xHH` where `HH` is any 2-digit hexadecimal number. The special escaped character forms `\r` (carriage return), `\n` (new line) and `\t` (tab) are also recognized..

Examples:

	REQUEST	RESPONSE
QUERY	**!rxname(1)?<CR>**	**OK rxname(1)="Chairman"<CRLF>**
UPDATE	**!rxname(2)="David"<CR>**	**OK rxname(2)="David"<CRLF>**

rxphase (Receiver audio phase)

This command may be used as a query to read the receiver audio phase status, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the audio phase inverted (shifted by 180 degrees), or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **99** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxphase(3)?<CR>**	**OK rxphase(3)=0<CRLF>**
QUERY	**!rxphase(*)?<CR>**	**OK rxphase(*)={0,1,0,0,0,0}<CRLF>**
UPDATE	**!rxphase(2)=1<CR>**	**OK rxphase(2)=1<CRLF>**
UPDATE	**!rxphase(*)={0,0,1,99,99,99}<CR>**	**OK rxphase(*)={0,0,1,1,1,1}<CRLF>**

rxpresent (Receiver present status)

This command may be used as a query to read the receiver present status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the receiver is installed in the mainframe, or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxpresent(3)?<CR>**	**OK rxpresent(3)=1<CRLF>**
QUERY	**!rxpresent(*)?<CR>**	**OK rxpresent(*)={1,1,1,1,0,0}<CRLF>**

rxpwr (Receiver power on/off)

This command may be used as a query to read the receiver power on/off status, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the receiver is powered on, or "0" meaning that it is powered off. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **99** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxpwr(3)?<CR>**	**OK rxpwr(3)=0<CRLF>**
QUERY	**!rxpwr(*)?<CR>**	**OK rxpwr(*)={1,1,1,0,0,0}<CRLF>**
UPDATE	**!rxpwr(2)=1<CR>**	**OK rxpwr(2)=1<CRLF>**
UPDATE	**!rxpwr(*)={1,1,1,0,99,99}<CR>**	**OK rxpwr(*)={1,1,1,0,0,0}<CRLF>**

rxrmeter (Receiver RF level meter)

This command may be used as a query to read the receiver RF level meter. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, in the range 0 to 50, representing the RF level in dBuV. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxrmeter(1)?<CR>**	**OK rxrmeter(1)=45<CRLF>**
QUERY	**!rxrmeter(*)?<CR>**	**OK rxrmeter(*)={0,12,45,0,0,0}<CRLF>**

rxscan (Receiver scan state)

This command may be used as a query to read the receiver scan state, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the receiver is scanning, or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **99** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxscan(3)?<CR>**	**OK rxscan(3)=0<CRLF>**
QUERY	**!rxscan(*)?<CR>**	**OK rxscan(*)={1,0,0,0,0,0}<CRLF>**
UPDATE	**!rxscan(2)=1<CR>**	**OK rxscan(2)=1<CRLF>**
UPDATE	**!rxscan(*)={1,0,0,99,99,99}<CR>**	**OK rxscan(*)={1,0,0,0,0,0}<CRLF>**

rxsmartnr (Receiver smart noise reduction)

This command may be used as a query to read the smart noise reduction status, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. If the channel address is wildcarded, then the data type is an array of integer of size 6. In this case the value **99** may be used in an update to indicate that a particular receiver is to **remain unchanged** by the command. The data type is integer, and is a code representing one of the following:

Code	Noise Reduction Level
0	None
1	Normal
2	Full

Examples:

	REQUEST	RESPONSE
QUERY	**!rxsmartnr(3)?<CR>**	**OK rxsmartnr(3)=1<CRLF>**
QUERY	**!rxsmartnr(*)?<CR>**	**OK rxsmartnr(*)={1,1,1,1,1,0}<CRLF>**
UPDATE	**!rxsmartnr(2)=0<CR>**	**OK rxsmartnr(2)=0<CRLF>**
UPDATE	**!rxsmartnr(*)={0,1,0,99,99,99}<CR>**	**OK rxsmartnr(*)={1,1,1,1,1,1}<CRLF>**

rxsquelch (Receiver squelch status)

This command may be used to determine if a receiver is squelched. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the receiver is squelched (no audio output), or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxsquelch(1)?<CR>**	**OK rxsquelch(1)=1<CRLF>**
QUERY	**!rxsquelch(*)?<CR>**	**OK rxsquelch(*)={0,1,1,0,0,0}<CRLF>**

rxstat (Receiver status)

This command may be used as a query to read *real-time* receiver status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is array of integer, with a length of 9. The values contained in the array are interpreted by position as follows:

Position	Value
1	Receiver present status (0 = not present, 1 = present)
2	Power status (0 = powered down, 1 = powered up)
3	Link Status (0 = no link, 1 = link established)
4	Audio meter (in range -80 to +5 dB, referenced to the transmitters's threshold of limiting, *before* application of rxalevel setting)
5	RF meter (in range 0 to 50 dBuV)
6	Scan status (0 = normal operation, 1 = receiver scanning)
7	Antenna diversity status (1 = antenna A selected, 2 = antenna B selected)
8	Mute status (0 = unmuted, 1 = muted)
9	Squelch status (0 = unsquelched, 1 = squelched)

If a receiver is not present in the mainframe or is powered down, then the remaining data is invalid.

Example:

	REQUEST	RESPONSE

QUERY	**!rxstat(1)?<CR>**	**OK rxstat(1)={1,1,1,-12,22,0,1,0,0}<CRLF>**

rxtben (Receiver talkback enable)

This command may be used as an update to enable or disable the talkback feature on a channel. If the channel address is wildcarded, then the data type is an array of integer of size 6. The data type is integer, either "1" meaning that talkback is enabled, or "0" meaning that it is not.

Examples:

	REQUEST	RESPONSE
UPDATE	**!rxtben(2)=1<CR>**	**OK rxtben(2)=1<CRLF>**
UPDATE	**!rxtben(*)={1,0,0,0,0,0}<CR>**	**OK rxtben(*)={1,0,0,0,0,0}<CRLF>**

rxtboutch (Receiver talkback output channel)

This command may be used as a query to read a receiver's talkback output channel, or as an update to set it. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, representing the *channel to which talkback audio is routed* when talkback is invoked on the transmitter associated with the addressed receiver. This setting is meaningful only if talkback is enabled for the receiver. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!rxtboutch(1)?<CR>**	**OK rxtboutch(1)=5<CRLF>**
QUERY	**!rxtboutch(*)?<CR>**	**OK rxtboutch(*)={5,6,6,6,6,6}<CRLF>**
UPDATE	**!rxtboutch(1)=6<CR>**	**OK rxtboutch(1)=6<CRLF>**
UPDATE	**!rxtboutch(*)={6,6,6,6,6,6}<CR>**	**OK rxtboutch(*)={6,6,6,6,6,6}<CRLF>**

rxtone (Receiver output test tone)

This command may be used as an update to enable or disable the receiver audio test tone. If the channel address is wildcarded, then the data type is an array of integer of size 6. The data type is integer, either "1" meaning that the test tone is enabled, or "0" meaning that it is not.

Examples:

	REQUEST	RESPONSE
UPDATE	**!rxtone(2)=1<CR>**	**OK rxtone(2)=1<CRLF>**
UPDATE	**!rxtone(*)={1,0,0,99,99,99}<CR>**	**OK rxtone(*)={1,0,0,0,0,0}<CRLF>**

rxver (Receiver firmware version)

This command may be used as a query to read the receiver's firmware version number. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data is a string type.

Example:

	REQUEST	RESPONSE
QUERY	**!rxver(1)?<CR>**	**OK rxver(1)="1.0"<CRLF>**

VRM2WB Channel Transmitter Commands

txautoon	Transmitter auto-on
txbalert	Transmitter battery timer alert threshold
txbatt	Transmitter battery selection
txbl	Transmitter backlight control
txblevel	Transmitter battery level
txblock	Transmitter frequency block
txbtime	Transmitter battery timer
txbutton	Transmitter button mode
txbwarn	Transmitter battery warning
txgain	Transmitter audio gain
txlock	Transmitter panel lock status
txphase	Transmitter audio phase
txrolloff	Transmitter low frequency roll off
txstat	Transmitter status
txstepsize	Transmitter tuning step size

Termination: all commands are terminated with an ASCII **carriage return** character (hex code 0x0D), represented by **<CR>** in the examples below. All responses are terminated with an ASCII **carriage return, line feed** pair (hex codes 0x0D, 0x0A), represented by **<CRLF>** in the examples below. An ellipsis (**...**) represents members of an array that have been omitted from an example for the sake of brevity.

Verbose response: commands prefixed with an exclamation point (bang) character result in a "verbose" response containing both the name of the property being addressed and its current value (if any). The verbose response returns the property/value pair in the "assignment" form, for example **`OK rxphase(2)=0 <CRLF>`**. This supports certain 3rd party control programming styles where the response needs to be self-describing.

txautoon (Transmitter auto-on)

This command may be used as a query to read the transmitter auto-on status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that auto-on is enabled, or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!txautoon(3)?<CR>**	**OK txautoon(3)=1<CRLF>**
QUERY	**!txautoon(*)?<CR>**	**OK txautoon(*)={0,0,0,0,1,0}<CRLF>**

txbalert (Transmitter battery timer alert threshold)

This command may be used as a query to read the transmitter battery elapsed time. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, representing the elapsed time in minutes, in the range 0 to 599. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!txbalert(1)?<CR>**	**OK txbalert(1)=-10<CRLF>**
QUERY	**!txbalert(*)?<CR>**	**OK txbalert(*)={20,31,210,234,0,0}<CRLF>**

txbatt (Transmitter battery selection)

This command may be used as a query to read the transmitter battery selection. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. If the channel address is wildcarded, then the data type is an array of integer of size 6. The data type is integer and is a code that specifies the battery type. It may be in the range 0 to 3, with the following meanings:

Code	Battery Type
0	1.5V Alkaline
1	1.5V Lithium
2	9V Alkaline
3	9V Lithium

Examples:

	REQUEST	RESPONSE
QUERY	**!txbatt(3)?<CR>**	**OK txbatt(3)=0<CRLF>**

QUERY	**!txbatt(*)?<CR>**	**OK txbatt(*)={1,0,0,0,0,0}<CRLF>**

txbl (Transmitter backlight timeout)

This command may be used as a query to read the transmitter backlight control setting. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. If the channel address is wildcarded, then the data type is an array of integer of size 6. The data type is integer and is a code that specifies the backlight timeout. It may be in the range 0 to 2, with the following meanings:

Code	Backlight timeout
0	No timeout
1	Times out in 30 seconds
2	Times out in 5 minutes

Examples:

	REQUEST	RESPONSE
QUERY	**!txbl(3)?<CR>**	**OK txbl(3)=1<CRLF>**
QUERY	**!txbl(*)?<CR>**	**OK txbl(*)={1,1,0,0,1,0}<CRLF>**

txblevel (Transmitter battery level)

This command may be used as a query to read the transmitter battery level. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, in the range 0 to 31, representing the battery voltage in tenth-volt increments. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!txblevel(1)?<CR>**	**OK txblevel(1)=29<CRLF>**
QUERY	**!txblevel(*)?<CR>**	**OK txblevel(*)={29,27,30,0,0,0}<CRLF>**

txblock (Transmitter frequency block)

This command may be used as a query to read the transmitter frequency block. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is string.

Examples:

	REQUEST	RESPONSE
QUERY	**!txblock(1)?<CR>**	**OK txblock(1)="A1"<CRLF>**

txbtime (Transmitter battery time)

This command may be used as a query to read the transmitter battery elapsed time. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, representing the elapsed time in minutes, in the range 0 to 599. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!txbtime(1)?<CR>**	**OK txbtime(1)=-10<CRLF>**
QUERY	**!txbtime(*)?<CR>**	**OK txbtime(*)={20,31,210,234,0,0}<CRLF>**

txbutton (Transmitter button function)

This command may be used as a query to read the transmitter button function. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. If the channel address is wildcarded, then the data type is an array of integer of size 6. The data type is integer and is a code that specifies the button function. It may be in the range 0 to 2, with the following meanings:

Code	Button Function
0	No function
1	Mute on/off
2	Power on/off

Examples:

	REQUEST	RESPONSE

QUERY	**!txbutton(3)?<CR>**	**OK txbutton(3)=1<CRLF>**
QUERY	**!txbutton(*)?<CR>**	**OK txbutton(*)={1,1,0,0,1,0}<CRLF>**

txbwarn (Transmitter battery warning)

This command may be used as a query to read the transmitter battery warning level. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. If the channel address is wildcarded, then the data type is an array of integer of size 6. The data type is integer and is a code that specifies the battery warning level for the transmitter. It may be in the range 0 to 4, with the following meanings:

Code	Warning Level
0	Off
1	Less than 25% battery life left
2	Less than 10% battery life left
3	Reserved, not used
4	Battery time expired alarm

Examples:

	REQUEST	RESPONSE
QUERY	**!txbwarn(1)?<CR>**	**OK txbwarn(1)=0<CRLF>**
QUERY	**!txbwarn(*)?<CR>**	**OK txbwarn(*)={0,1,0,0,0,0}<CRLF>**

txgain (Transmitter audio gain)

This command may be used as a query to read the transmitter audio gain. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, representing the gain on a scale of 0 to 42. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE

QUERY	**!txgain(1)?<CR>**	**OK txgain(1)=22<CRLF>**
QUERY	**!txgain(*)?<CR>**	**OK txgain**(*)={24,26,20,27,22,22}<CRLF>

txlock (Transmitter panel lock status)

This command may be used as a query to read the transmitter panel lock status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the transmitter panel is locked, or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!txlock(3)?<CR>**	**OK txlock(3)=1<CRLF>**
QUERY	**!txlock(*)?<CR>**	**OK txlock(*)={0,0,0,0,1,0}<CRLF>**

txphase (Transmitter audio phase)

This command may be used as a query to read the transmitter audio phase status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, either "1" meaning that the audio phase inverted (shifted by 180 degrees), or "0" meaning that it is not. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!txphase(3)?<CR>**	**OK txphase(3)=0<CRLF>**
QUERY	**!txphase(*)?<CR>**	**OK txphase(*)={0,1,0,0,0,0}<CRLF>**

txrolloff (Transmitter low frequency rolloff)

This command may be used as a query to read the transmitter low frequency rolloff. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. If the channel address is wildcarded, then the data type is an array of integer of size 6. The data type is integer

and is a code that specifies the low frequency rolloff in Hz. It may be in the range 0 to 5, with the following meanings:

Code	Low Frequency Rolloff
0	35 Hz
1	50 Hz
2	70 Hz
3	100 Hz
4	120 Hz
5	150 Hz

Examples:

	REQUEST	RESPONSE
QUERY	**!txrolloff(3)?<CR>**	**OK txrolloff(3)=0<CRLF>**
QUERY	**!txrolloff(*)?<CR>**	**OK txrolloff(*)={1,1,2,1,0,0}<CRLF>**

txstat (Transmitter status)

This command may be used as a query to read *real-time* transmitter status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is array of integer, with a length of 8. The values contained in the array are interpreted by position as follows:

Position	Value
1	Link Status (0 = no link, 1 = link established)
2	Battery timer (elapsed time in minutes)
3	Battery warning status (0 = ok, 1 = less than 25%, 2 = less than 10%, 4 = timer alarm)
4	Front panel lock status (0 = unlocked, 1 = locked)
5	Audio meter (in range -80 to +5 dB, referenced to the threshold of limiting)
6	Audio limiter status (0 = not in limiting, 1 = in limiting)
7	Audio clipping status (0 = not clipping, 1 = clipping)

Data is valid only when link is established (link status = 1).

Example:

	REQUEST	RESPONSE
QUERY	**!txstat(1)?<CR>**	**OK txstat(1)={1,72,0,0,-18,0,0}<CRLF>**

txstepsize (Tranmitter tuning step size)

This command may be used as a query to read the transmitter frequency step size. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is integer, representing the frequency set size in kHz. Allowable values are 25 and 100 kHz. If the channel address is wildcarded, then the data type is an array of integer of size 6.

Examples:

	REQUEST	RESPONSE
QUERY	**!txstepsize(1)?<CR>**	**OK txstepsize(1)=100<CRLF>**
QUERY	**!txstepsize(*)?<CR>**	**OK txstepsize(*)={100,100,100,100,100,100}<CRLF>**

VRM2WB Channel Status Commands

chstat	Channel status

Termination: all commands are terminated with an ASCII **carriage return** character (hex code 0x0D), represented by **<CR>** in the examples below. All responses are terminated with an ASCII **carriage return, line feed** pair (hex codes 0x0D, 0x0A), represented by **<CRLF>** in the examples below. An ellipsis (**...**) represents members of an array that have been omitted from an example for the sake of brevity.

Verbose response: commands prefixed with an exclamation point (bang) character result in a "verbose" response containing both the name of the property being addressed and its current value (if any). The verbose response returns the property/value pair in the "assignment" form, for example **OK rxphase(2)=0 <CRLF>**. This supports certain 3rd party control programming styles where the response needs to be self-describing.

chstat (Channel status)

This command may be used as a query to read *real-time* channel status. The channel is specified by using the address syntax. Addresses must be in the range 1 to 6. The data type is array of integer, with a length of 15. The values contained in the array are interpreted by position as follows:

Position	Value
1	Receiver present status (0 = not present, 1 = present)
2	Receiver power status (0 = powered down, 1 = powered up)
3	Link Status (0 = no link, 1 = link established)
4	Audio meter (in range -80 to +5 dB, referenced to the transmitters's threshold of limiting, *before* application of rxalevel setting)
5	RF meter (in range 0 to 50 dBuV)
6	Scan status (0 = normal operation, 1 = receiver scanning)
7	Antenna diversity status (1 = antenna A selected, 2 = antenna B selected)
8	Receiver mute status (0 = unmuted, 1 = muted)
9	Receiver squelch status (0 = unsquelched, 1 = squelched)
10	Transmitter battery timer (elapsed time in minutes)
11	Transmitter battery warning status (0 = ok, 1 = less than 25%, 2 = less

	than 10%, 4 = timer alarm)
12	Transmitter front panel lock status (0 = unlocked, 1 = locked)
13	Transmitter audio limiter status (0 = not in limiting, 1 = in limiting)
14	Transmitter audio clipping status (0 = not clipping, 1 = clipping)

If a receiver is not present in the mainframe or is powered down, then all remaining data is invalid. Transmitter data is valid only when link is established (link status = 2).

Example:

	REQUEST	RESPONSE
QUERY	**!chstat(1)?<CR>**	**OK chstat(1)={1,1,2,-3,22,0,1,0,0,97,0,0,0,0}<CRLF>**

VRM2WB Network Setup Commands

defgate	Default gateway
dhcpen	DHCP enable
httpport	HTTP port number
ipaddr	IP address
macaddr	MAC address
netmask	Network mask
tcpport	TCP port number

Termination: all commands are terminated with an ASCII **carriage return** character (hex code 0x0D), represented by **<CR>** in the examples below. All responses are terminated with an ASCII **carriage return, line feed** pair (hex codes 0x0D, 0x0A), represented by **<CRLF>** in the examples below. An ellipsis (**...**) represents members of an array that have been omitted from an example for the sake of brevity.

Verbose response: commands prefixed with an exclamation point (bang) character result in a "verbose" response containing both the name of the property being addressed and its current value (if any). The verbose response returns the property/value pair in the "assignment" form, for example **`OK ingn(2)=40 <CRLF>`**. This supports certain 3rd party control programming styles where the response needs to be self-describing.

defgate (default gateway)

This command may be used as a query to read the Default Gateway address, or as an update to set it. The data type is string, containing the address in IP "dotted quad" format.

Example:

	REQUEST	RESPONSE
QUERY	**`defgate?<CR>`**	**`OK "172.16.4.1"<CRLF>`**
UPDATE	**`defgate="172.16.4.1"<CR>`**	**`OK<CRLF>`**

dhcpen (DHCP enable)

This command may be used as a query to read the DHCP enable status, or as an update to set it. The data type is integer, either "1" meaning that the DHCP client feature is enabled, or "0" meaning that it is not. If enabled, DHCP (Dynamic Host Configuration Protocol) is used at power up to obtain an IP address. **Note:** If this setting is changed, the new value takes effect the next time the device is powered up.

Example:

	REQUEST	RESPONSE
QUERY	**dhcpen?<CR>**	**OK 0<CRLF>**
UPDATE	**dhcpen=1<CR>**	**OK<CRLF>**

httpport (HTTP port number)

This command may be used as a query to read the HTTP port number assignment, or as an update to set it. The data type is integer, in the range 0 to 65535, representing the port number used for HTTP connections to the device. The default value is 80.

Example:

	REQUEST	RESPONSE
QUERY	**httpport?<CR>**	**OK 80<CRLF>**
UPDATE	**httpport=80<CR>**	**OK<CRLF>**

ipaddr (IP address)

This command may be used as a query to read the IP address of the device, or as an update to set it. The data type is string, containing the address in IP "dotted quad" format.

Example:

	REQUEST	RESPONSE
QUERY	**ipaddr?<CR>**	**OK "172.16.4.151"<CRLF>**
UPDATE	**ipaddr="172.16.4.151"<CR>**	**OK<CRLF>**

macaddr (MAC address)

This command may be used as a query to read the ethernet MAC address of the device. The data type is string, containing the address in IEEE MAC-48 format.

Example:

	REQUEST	RESPONSE
QUERY	**macaddr?<CR>**	**OK "00-24-34-32-00-22"<CRLF>**

netmask (network mask)

This command may be used as a query to read the Network Mask, or as an update to set it. The data type is string, containing the mask in IP "dotted quad" format.

Example:

	REQUEST	RESPONSE
QUERY	**netmask?<CR>**	**OK "255.255.255.0"<CRLF>**
UPDATE	**netmask="255.255.255.0"<CR>**	**OK<CRLF>**

tcpport (TCP port number)

This command may be used as a query to read the TCP port number assignment, or as an update to set it. The data type is integer, in the range 0 to 65535, representing the port number used for TCP connections to the device. The default value is 4080.

Example:

	REQUEST	RESPONSE
QUERY	**tcpport?<CR>**	**OK 4080<CRLF>**
UPDATE	**tcpport=4080<CR>**	**OK<CRLF>**