

# **SymNet™** | Network Audio Solutions



## External Control Protocol

**SymNet Designer Rev 8.x**

## Contents

<b>Introduction</b> .....	<b>1</b>	<b>Push Commands</b> .....	<b>11</b>
About this document .....	1	(PU) Global Push Enable/Disable.....	14
Conventions used in this document .....	1	(PUE) Push Enable.....	15
<b>General Notes</b> .....	<b>1</b>	(PUD) Push Disable.....	15
Connections .....	1	(GPU) Get Push-enabled Controllers.....	16
RS-232 Port Configuration.....	2	(PUR) Push Refresh .....	16
RS-232 Control .....	2	(PUC) Push Clear .....	17
Ethernet Control .....	2	(PUI) Set Push Interval.....	17
RS-485 Control .....	3	(PUT) Set Push Threshold.....	17
<b>Parameter Notes</b> .....	<b>3</b>	<b>Setup Commands</b> .....	<b>18</b>
Faders .....	3	(SP) Set Port.....	18
Buttons.....	3	(SB) Set Baud.....	19
Input and Output Selectors .....	3	(SQ) Set Quiet Mode .....	19
Meters .....	3	(EH) Set Echo Mode .....	19
Other Parameters .....	4		
Parameters that cannot be controlled externally .....	4		
<b>Getting Started</b> .....	<b>5</b>		
Protocol .....	5		
<b>Control Commands</b> .....	<b>5</b>		
(CS) Control RS-232/485.....	5		
(CSG) Control RS-232/485 Global .....	5		
(CC) Change RS-232/485 Controller.....	6		
(GS) Get RS-232/485 .....	6		
(GS2) Get RS-232/485 with Controller Number.....	7		
(GSB) Get RS-232/485 Block.....	7		
(GSB2) Get RS-232/485 Block with Controller Number.....	8		
(SV) Set Volume.....	8		
(GC) Get Configuration .....	8		
(LC) Load Configuration.....	9		
(LCG) Load Configuration Global.....	9		
(GPR) Get Preset.....	9		
(LP) Load Preset .....	9		
(LPG) Load Preset Global .....	10		
(SU) Set Unit .....	10		
(SUQ) Set Unit Quiet .....	10		
(FU) Flash Unit .....	11		

## Introduction

### About this document

The purpose of this document is to provide a technical understanding of the Symetrix SymNet Command Protocol. It will define and illustrate the protocol used to communicate with the SymNet via 3rd-party interface. As of SymNet Designer 7.0, this document will be known as the “External Control Protocol” document as SymNet not only allows 3rd-party control via RS-232, but also UDP/IP (Ethernet) as well.

SymNet Devices can be controlled by 3rd-party based controllers such as certain AMX or Crestron models, or any RS-232 or Ethernet equipped device that can be adapted to this protocol. The protocol consists of humanly readable text commands and responses.

Control is achieved by using the existing controller assignment system in SymNet Designer for RS-232/485 control over most processing parameters. Right-clicking on most faders and buttons of module views will allow assigning the parameter to an RS-232/485 controller. An appropriate controller number is assigned in the 1-10000 range, then the design with the controller assignment is downloaded. At this point, the RS-232 or Ethernet interfaces, referencing the controller number can control the parameter.

### Conventions used in this document

A dollar sign (\$) preceding a set of alphanumeric characters denotes a hex value. All other number values should be considered decimal values. Example: “\$A0” represents the decimal value of “160”.

Values enclosed in [square brackets] are optional parameters and do not need to be include. If omitted, default values will be used as described for each command.

## General Notes

### Connections

By default, the rear panel RS-232 port is reserved for RS-232 Accessory Controllers. See the section RS-232 Port Configuration for information on how to change this if needed. For the sake of clarity the RS-232 port on the front of the SymNet will be referred to as the “Host PC Control Port” located on the front of the unit, and the “Accessory Remote Control Port” located on the rear of the unit. These names reflect the manner in which these ports are typically used. Note: the DigIO 12x12 DSP and Express units only have one RS-232 port on the rear. Please see the their respective Quick Start Guides for information regarding the configuration of this port.

Connect your RS-232 based accessory remote to the front or rear panel RS-232 connector as determined after reading the RS-232 Port Configuration section. For experimentation purposes, a terminal emulation program such as HyperTerminal included with Windows can be run on a PC instead. Typically this connection requires a “straight through” cable, but a null-modem cable may be required depending on the manufacturer. In general, if the sexes of the two connectors you are trying to connect are the same, a “null modem” cable is required. Set up the controller for baud: 38400, data bits: 8, stop bits: 1, parity: NONE. No handshaking or flow control is used.

### RS-232 Port Configuration

As previously mentioned, SymNet 8x8, 8in, 8out and CobraLink units all have two RS-232 DB-9 connectors: one on the front and one on the rear. Typically, the front panel port is used for host PC control and the rear panel is used for accessory remote controller input. Alternatively, the rear panel jack can be used for host PC control in systems requiring a permanent host PC connection where exposed front panel wiring is not acceptable. But in this situation, accessory remote controllers should not be used. It should also be noted that the BreakIn12 and BreakOut12 SymNet Devices also have a rear panel RS-232 port. This port on these two units is reserved for factory use and should can not be used for host or accessory control. Your accessory controller should be plugged into a master SymNet Device at Ring x, Address 1.

In order to handle this situation and allow the operation of accessory controllers without the host PC, a master SymNet Device can be programmed to assign the host PC connection to either the front or rear panel jacks. This is done using the Set Port command and is fully described on page 18. This setting is done to non-volatile memory, and will be remembered through power cycles. Additional setup commands can also be found on pages 18-19.

Note: As of SymNet Designer 4.0, the parameters of the Accessory Serial Port can be set directly from SymNet Designer. This is done from the Upgrade Firmware dialog found under the Hardware menu of SymNet Designer.

In addition to the Set Port command (see page 18), there are also additional commands, Set Baud, Set Quiet Mode and Set Echo that affect the remote control RS-232 port. The unit's default settings (Baud 38400, Quiet Mode ON, Echo OFF) are typical for most applications, so most users will not need to know about these commands. However, they are also documented for reference on pages 18-19.

## RS-232 Control

The RS-232 protocol is used to simulate the movement of a RS-232/485 controller by referencing the controller number and a new virtual position. As previously mentioned, controller numbers can be 1-10000. Controller positions are a 16-bit unsigned number providing a range of 0-65535.

## Ethernet Control

Starting with SymNet Designer V7.0, SymNet Devices with Ethernet ports support a new remote control protocol. This protocol allows the extension of the existing human-readable RS-232 command language to devices where an RS-232 port is not an available option, and Ethernet is.

The protocol is similar to Telnet in use. However, instead of using TCP as Telnet does, it uses UDP. And, it does not use any of the options or escape sequence found in Telnet.

To use this feature, command strings following the existing RS-232 command language can be sent as the payload of a UDP packet. The following rules should be observed in sending commands:

- Commands should be formatted exactly as defined in the RS-232 control protocol.
- Command strings may or may not include a zero termination character.
- Commands should not be broken up across multiple packets
- Commands should be sent to UDP port number 48630 of the proper SymNet Device's IP address.
- If high reliability communications are required, responses to commands should be analyzed for success.

Responses to commands will exhibit the following behavior:

- Responses to each command issued will be returned in a single packet unless the response is larger than a single packet can hold. Responses will not have any single carriage return-terminated line broken up across packets unless there is no carriage return in the response.
- Response will be returned to the same IP address and source port number that sent the packet.
- Responses will follow the configuration of the port, just as if it were an RS-232 port. For example, echo mode, quiet mode and deaf mode will be kept independently for the RS-232 port and the Ethernet port. And this configuration will be retained across power cycles.
- Responses will not include a zero-termination character.
- All transmissions originating from SymNet Devices will either be responses to commands or pushed data as configured in SymNet Designer.

Each command sent to a SymNet Device contains information in the packet header as to who sent the command, and so, where a response will be sent. This information is saved when a packet is received by a SymNet Device. All responses will go to the last received IP address and port. And this source IP and port are saved in non-volatile memory across power cycles.

Until the first command is received, responses will not know where they are supposed to be sent. This normally not an issue as communication from the SymNet Device is generally a response to a command. However, if the SymNet Device is set up to push control data, it will also be pushed out this UDP port. If no valid packets have ever been received by a SymNet Device, pushed data will not be sent out the Ethernet port.

Pushing of data is not controlled independently for RS-232 and Ethernet ports. It is either enabled or disabled for both.

**Note:** The RS-232 serial port and the Ethernet port are essentially independent. They maintain separate settings for quiet, echo, and deaf modes. Commands sent to one port are not echoed out the other, and responses are sent only to the port that from which the command was received. Hence, the two ports will not necessarily send out the same data. One exception to this is push data, which is sent out both ports in parallel.

## RS-485 Control

RS-485 control is generally done using one or more of the Symetrix ARC (Adaptive Remote Controller) devices. Further discussion of RS-485 and the ARCs can be found on the SymNet web site.

## Parameter Notes

### Faders

Faders in SymNet Designer can be controlled to the limits of their minimum and maximum values shown in the Module View screens. A controller position of zero (0) will cause the minimum fader position to be realized. A controller position of 65535 will cause the maximum fader position to be realized. Increasing positions will move the fader linearly in dB.

Most volume faders in SymNet Designer have a range of -72dB to +12dB. In these cases, the following formula can be used to convert from RS-232/485 controller position to dB:

$$\text{Volume dB} = -72 + 84 * (\text{CONTROLLER POSITION} / 65535)$$

If CONTROLLER POSITION = 0, Volume dB = OFF

Note that some faders have a different range than -72 to +12dB and future versions of SymNet Designer may allow further scaling and limiting the range of RS-232/485 controller values. In this case, the formula will depend upon the actual fader range. The more general formula is shown below:

$$\text{Volume dB} = \text{MINIMUM VALUE} + (\text{MAXIMUM VALUE} - \text{MINIMUM VALUE}) * (\text{CONTROLLER POSITION} / 65535)$$

Where **MINIMUM VALUE** is the fader's lower limit in dB and **MAXIMUM VALUE** is the fader's upper limit in dB.

### Buttons

Buttons in SymNet Designer such as a mute or bypass can be controlled similarly with controller positions by sending the minimum value (0) to turn the switch off (button not pushed) and sending the maximum value (65535) to turn the switch on (button pushed).

### Input and Output Selectors

Controlling selectors can either be done by assigning a controller to the individual buttons, or to the single selector slider. It is recommended that the selector slider be used because it allows reading back the current state whereas the buttons do not. A value of zero (0) will select the first input or output and a value of (65535) will select the last input or output. Other values are selected by sending evenly spaced (linear) values as shown by the formula below:

$$\text{Controller Value} = (\text{INPUT NUMBER} - 1) * 65535 / (\text{NUMBER OF INPUTS} - 1)$$

Note: for an output selector, replace "INPUT" with "OUTPUT" above.

### Meters

Meters in the SymNet Designer dedicated meter modules can be read via RS-232. The read back value will be linear in dB with 65535 representing 0 dBFS and 0 representing -72 dBFS (or less). The formula below can be used to calculate a dB reading from a controller value:

$$\text{Level dBFS} = 72 * (\text{CONTROLLER VALUE} / 65535) - 72$$

If CONTROLLER VALUE = 0, Level dBFS <= -72 dBFS

Input and output meters in some other modules such as Automixers, Compressors, and AGCs can also be read via RS-232. In this case, the read back value will be linear in dB with 65535 representing the maximum value shown on the meter and 0 representing the minimum value shown on the meter (or less). The formula below can be used to calculate a dB reading from a controller value:

$$\text{Level dBFS} = (\text{MAXIMUM VALUE} - \text{MINIMUM VALUE}) * (\text{CONTROLLER VALUE} / 65535) + \text{MINIMUM VALUE}$$

If CONTROLLER VALUE = 0, Level dBFS <= MINIMUM VALUE

Where **MINIMUM VALUE** is the meter’s lower limit in dB and **MAXIMUM VALUE** is the meter’s upper limit in dB.

Control meters in dedicated control meter modules can also be read via RS-232. In this case, the read back value will be linear with 65535 representing 100% and 0 representing 0%. The formula below can be used to calculate a percentage reading from a controller value:

$$\text{Level \%} = 100 * (\text{CONTROLLER VALUE} / 65535)$$

**Note:** Meters are a “read-only” parameter. Attempting to change the meter value via RS-232 will have no effect.

## LEDs

LEDs in various modules can be read via RS-232. The read back value will be either 65535 if the LED is on or 0 if the LED is off.

**Note:** LEDs are a “read-only” parameter. Attempting to change the LED value via RS-232 will have no effect.

## Other Parameters

Many other parameters such as compression ratios, delay times, EQ settings and pans can also be controlled externally. In fact, almost any DSP parameter in SymNet Designer can be controlled. For other parameter types, as in the above examples, sending a value of zero (0) will set the parameter to its minimum value and sending a value of (65535) will set it to its maximum value. Ratios, frequencies, width/Q, and attack/release/hold times all use a logarithmic scale. Pans and delay times use a linear scale. Quantities expressed in dB such as gains, volumes, thresholds and depths are linear in dB.

## Parameters that cannot be controlled externally

The following SymNet Designer features cannot be controlled externally. In some cases, workarounds are suggested.

	FUNCTION	DESCRIPTION
1.	Oscilloscope:	Trigger and Auto-Range controls
2.	Sine Generator:	Sweep button and parameters
3.	Parametric/British EQ:	Flatten All and Octaves/Q buttons
4.	Graphic EQ:	Flat and Bandwidth button
5.	Feedback Fighters:	Response Preset buttons
6.	Matrix:	Mute All and Disconnect All buttons (use individual mute or disconnect buttons instead)
7.	Mixers:	Link buttons (assign same controller to all linked faders instead)
8.	SPL Computers:	Calibrate button
9.	Loudspeaker Managers	Flatten All and Smart™ ID
10.	CobraLink Send/Return:	Meters
11.	Delay:	Default temperature button
12.	Meter Bar:	All meters
13.	Fader Control Input:	Log/linear buttons
14.	String Output:	All parameters
15.	RS-232/485 Input:	Controller number

## Getting Started

### Protocol

The RS-232 Accessory Remote Controller Protocol is a text-based (ASCII string) protocol. Commands are sent with simple character strings with terms separated by spaces and completed with a carriage return character <CR> (ASCII decimal 13 or hex \$0D). There are only a few commands that most applications will need to use, and they are described in the following pages. The general form for commands is:

**<COMMAND> <PARAMETER> <PARAMETER> ... <CR>**

A white space character (space, tab, etc.) must be included between the command and each parameter. Extra white space characters can be sent for readability if desired. In this document a single space will be used. If a command is accepted, the SymNet Device will respond to each command with an acknowledgement string whose syntax varies with each command.

All commands are addressed to a particular SymNet Device and in most cases affect only that unit. The currently addressed unit is chosen using the Set Unit command (see page 10). If addressed to the master unit, some commands affect all connected units. These commands, known as “global” commands, are described individually in the following pages.

### Control Commands

#### (CS) Control RS-232/485

Use this command to move a controller position on the currently addressed unit to a new absolute value. The command must specify the RS-232/485 controller number and the new controller position. The syntax of the command is:

**CS <CONTROLLER NUMBER> <CONTROLLER POSITION><CR>**

Where <CONTROLLER NUMBER> is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer, and <CONTROLLER POSITION> is a 16-bit number in decimal (0-65535).

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that the specified controller number is not controlling anything in the current design or that the specified controller number does not exist on the unit currently being addressed.

#### (CSG) Control RS-232/485 Global

This command is similar to the (CS) Control RS-232/485 command except that it affects all devices, not just the device currently addressed. This can greatly simplify things when controlling a multi-device system. This command should be issued to the master SymNet Device (device address 1). It may be necessary to precede this command with a Set Unit command to ensure that you are talking to the master device. If you design your Site File so that each unit has its own unique range of RS-232/485 controller numbers, you can use this command to issue CSG commands to the master unit only and the other slave units will respond accordingly to commands issued to the controller numbers they host. This eliminates the need to use the Set Unit command in order to talk to the unit that hosts the controller number to be addressed.

The syntax of the command is:

**CSG <CONTROLLER NUMBER> <CONTROLLER POSITION><CR>**

Where **<CONTROLLER NUMBER>** is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer, and **<CONTROLLER POSITION>** is a 16-bit number in decimal (0-65535).

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

## (CC) Change RS-232/485 Controller

Use this command to move a controller to a new relative value. This command will increment or decrement a controller by a specified amount. The command must specify the RS-232/485 controller number, whether it should be incremented or decremented, and the amount to change by. The syntax of the command is:

**CC <CONTROLLER NUMBER> <DEC/INC> <AMOUNT><CR>**

Where **<CONTROLLER NUMBER>** is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer, **<DEC/INC>** is 0 to decrement and 1 to increment, and **<AMOUNT>** is the amount to increment or decrement (a decimal number, 0-65535). If the amount to be decremented or incremented causes the parameter to exceed its minimum or maximum value, the value will be limited to its minimum or maximum value. For example, if you increment a parameter by 10 and its current value is 65530, the new value will be limited to 65535.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that the specified controller number is not controlling anything in the current design.

## (GS) Get RS-232/485

This command will return the controller position (value) associated with a specific controller number. The command must specify the RS-232/485 controller number. The syntax of the command is:

**GS <CONTROLLER NUMBER><CR>**

Where **<CONTROLLER NUMBER>** is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer.

If the command is accepted, the SymNet Device will respond with the string:

**<CONTROLLER POSITION><CR>**

Where controller position is a 16-bit number in decimal (0-65535)

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that the specified controller number is not controlling anything in the current design.

If the value being requested is a button that only has two states, the returned values will be either 0 or 65535, regardless of the actual value sent to the controller. For example, assume controller number 1 controls a mute button. If you send "CS <1> <754>", and then "GS <1>", it will return 0, not 754. More generally, if the parameter you are controlling has granularity coarser than the 16-bit values used, the returned values will be quantized to the granularity of the parameter. Example controls where you might observe this effect are buttons as mentioned above, selector sliders and the 0-2ms delay slider. Note that for selectors, reading back the values of the input buttons does not work correctly.

If the same controller number is controlling multiple parameters in SymNet, the value returned is the first one assigned by the user. Because of the confusion that may arise from this, it is recommended that every parameter controlled be given a unique controller number. If you would like a single user object (e.g. single slider) to control several different parameters (e.g. three different faders), then the combining of the controls should be done in the control system software.

**(GS2) Get RS-232/485 Controller with Controller Number**

This command will return the controller number with controller position (value) associated with it together in the same string. This command is provided at the request of AMX/Crestron programmers to make it easier to interpret and parse returned controller positions. The command must specify the RS-232/485 controller number. The syntax of the command is:

**GS <CONTROLLER NUMBER><CR>**

Where **<CONTROLLER NUMBER>** is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer.

If the command is accepted, the SymNet Device will respond with the string:

**<CONTROLLER NUMBER> <CONTROLLER POSITION><CR>**

Where controller position is a 16-bit number in decimal (0-65535)

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that the specified controller number is not controlling anything in the current design.

**(GSB) Get RS-232/485 Block**

This command will return the controller position (value) of a specific range of consecutive controller numbers. The command must specify the starting RS-232/485 controller number and the number of consecutive controllers to return. The syntax of the command is:

**GSB <CONTROLLER NUMBER> <BLOCK SIZE><CR>**

Where **<CONTROLLER NUMBER>** is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer and **<BLOCK SIZE>** is the number of consecutive controllers. Note that **<BLOCK SIZE>** can be at most 256.

If the command is accepted, the SymNet Device will respond with the string:

**<CONTROLLER POSITION1><CR>**

**<CONTROLLER POSITION2><CR>**

**<CONTROLLER POSITION3><CR>**

...

**"<CONTROLLER POSITIONn><CR>**

Where controller position is a 16-bit number in decimal (0-65535), or -1 if a controller is not controlling anything in the current design. The values will always be five digits, with leading zeros added as necessary (e.g. 7 would be returned as 00007<CR>).

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that the requested block size is larger than 256. For more information and tips on reading back controller numbers, see the GS command.

**Example command sent:**

**GSB 9 3<CR>**

**Example Response:**

**32321<CR>**

**00256<CR>**

**00003<CR>**

**(GSB2) Get RS-232/485 Block with Controller Number**

This command will return the controller number with controller position (value) associated with it for a specific range of consecutive controller numbers. The command is very similar to GSB described above, but the return string may be easier to process in some systems. The command must specify the starting RS-232/485 controller number and the number of consecutive controllers to return. The syntax of the command is:

**GSB2 <CONTROLLER NUMBER> <BLOCK SIZE><CR>**

Where **<CONTROLLER NUMBER>** is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer and **<BLOCK SIZE>** is the number of

consecutive controllers. Note that **<BLOCK SIZE>** can be at most 256.

If the command is accepted, the SymNet Device will respond with the string:

**#<CONTROLLER NUMBER1>=<CONTROLLER POSITION1><CR>**

**#<CONTROLLER NUMBER2>=<CONTROLLER POSITION2><CR>**

**#<CONTROLLER NUMBER3>=<CONTROLLER POSITION3><CR>**

...

**#<CONTROLLER NUMBERn>=<CONTROLLER POSITIONn><CR>**

Where **<CONTROLLER NUMBERn>** is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer and **<CONTROLLER POSITIONn>** is a 16-bit number in decimal (0-65535), or -1 if a controller is not controlling anything in the current design. The values for the controller number and position will always be five digits, with leading zeros added as necessary (e.g. 7 would be returned as 00007).

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that the requested block size is larger than 256. For more information and tips on reading back controller numbers, see the GS command.

Example command sent:

**GSB2 9 3<CR>**

Example Response:

**#00009=32321<CR>**

**#00010=00256<CR>**

**#00011=00003<CR>**

## (SV) Set Volume

This command will set the analog input or output volume of one or all inputs/outputs to the specified value (8x8, 8in and 8out DSP units only). The syntax of the command is:

**SV <I/O NUMBER> <VOLUME><CR>**

Where **<I/O NUMBER>** = 1-8 for inputs (1-8), 101-108 for

the outputs (1-8), 0 for all inputs, 100 for all outputs or 1000 for all inputs and outputs. **<VOLUME>** is a decimal gain value in dB ranging from -96.0 to 31.5. A value of -96.0 is equivalent to mute or OFF.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

Example: "SV 104 -12.5<CR>" sets the volume of output #4 to -12.5 dB.

## (GC) Get Configuration

This command will return the number of the currently running configuration. The syntax of the command is:

**GC<CR>**

If the command is accepted, the SymNet Device will respond with the string:

**<CONFIG NUMBER><CR>**

As of SymNet Designer version 4.0, there is only one configuration, so the **<CONFIG NUMBER>** return value will be either -1, 0 or 1. A return value of -1 indicates that no configuration is currently running (dancing lights). A return value of 0 indicates a configuration downloaded by the host is currently running (from an F5 download). A return value of 1 indicates the configuration stored in flash is currently running (from an F4 download).

If the command is interpreted but fails for any reason or no configuration has been downloaded or triggered, the SymNet Device will respond with the string:

**NAK<CR>**

### (LC) Load Configuration

This command will load and start running the specified configuration on the currently addressed unit. The syntax of the command is:

**LC <CONFIG NUMBER><CR>**

Where <CONFIG NUMBER> = 1.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

**Note:** this command affects the currently addressed unit only. If the intent is to “globally” load a configuration across multiple units, see the (LCG) Load Configuration Global command on the next page.

### (LCG) Load Configuration Global

This command will load and start running the specified configuration on all SymNet Devices. This command should be issued to the master SymNet Device (device address 1). It may be necessary to precede this command with a Set Unit command to ensure that you are talking to the master unit. The syntax of the command is:

**LCG <CONFIG NUMBER><CR>**

Where <CONFIG NUMBER> = 1.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

### (GPR) Get Preset

This command will return the last preset that was loaded. The syntax of the command is:

**GPR<CR>**

If the command is accepted, the SymNet Device will respond with the string:

**<PRESET NUMBER><CR>**

If a configuration is running, the <PRESET NUMBER> return value will be 0-1000. A return value of 0 indicates that no preset has been recalled.

If the command is interpreted but fails for any reason or no configuration has been downloaded or triggered, the SymNet Device will respond with the string:

**NAK<CR>**

### (LP) Load Preset

This command will load the specified preset (1-1000) on the currently addressed unit. The syntax of the command is:

**LP <PRESET NUMBER><CR>**

Where <PRESET NUMBER> = 1-1000.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that no configuration has been downloaded.

**Note:** this command affects the currently addressed unit only. If the intent is to “globally” load a preset across multiple units, see the (LPG) Load Preset Global command on the next page.

## (LPG) Load Preset Global

This command will load the specified preset (1-1000) on all SymNet Devices. This command should be issued to the master SymNet Device (device address 1). It may be necessary to precede this command with a Set Unit command to ensure that you are talking to the master unit. The syntax of the command is:

**LPG <PRESET NUMBER><CR>**

Where <PRESET NUMBER> = 1-1000.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

A typical reason for failure is that no configuration has been downloaded or the command was not issued to a master unit.

## (SU) Set Unit

This command will cause all future commands to be re-directed to the unit at the specified SymLink address. This redirection will persist until power cycling or the next Set Unit command. The Set Unit command can only be sent if you are physically connected to the Accessory Remote Controller Port of a SymLink Master Device (Device Address of 1). NOTE: This is the only command that includes a hex value number. For Express units, the unit ID is calculated assuming the DIP switch indicates the RING NUMBER and the DEVICE ADDRESS is always 1. The syntax of the command is:

**SU <\$DEVICE ID><CR>**

Where <\$DEVICE ID> is a unique identifier for each hardware device in a SymNet system. If the system consists of just a single SymLink ring, then the DEVICE ID is simply equal to the unit's device address as set by the rear panel DIP switches. If the system contains CobraLink units and multiple SymNet rings, then DEVICE ID combines the ring number as set by the CobraLink's rear panel DIP switches and the device address.

Combining the ring number and device address is done according to the formula:

$$\langle \$DEVICE ID \rangle = (\langle RING NUMBER \rangle - 1) * \$100 + \langle DEVICE ADDRESS \rangle$$

If only one ring is present, then the ring number is 1.

Example Formulas	Actual Commands to Send:
Ring 1, Device Address 1, <\$DEVICE ID> = \$001	SU 1<CR>
Ring 2, Device Address 3, <\$DEVICE ID> = \$103	SU 103<CR>
Ring 5, Device Address 7, <\$DEVICE ID> = \$407	SU 407<CR>

(Note that even though a (\$) is used in the above example formulas to indicate a hex value, you do not actually use the (\$) in the actual command sent, just the hex value as a decimal number.)

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

If you do not use the Set Unit command, changes to RS-232/485 controller positions will only be sent to the unit that you are physically connected to (with the exception of the commands designated as "global" which are sent to the master unit and broadcasted on to the other units).

**NOTE:** SU 0 resets the unit state back to "itself" so all commands are processed by the physically connected unit. (This is handy in that you can reset the unit state back to the power-on default without having to know the ID of the unit you are physically connected to.)

## (SUQ) Set Unit Quiet

This command will cause all future commands to be re-directed to the unit at the specified SymLink address. This command is identical to the Set Unit command, except that it does not respond with the ACK or NAK that you otherwise would receive. It is intended to allow a single command string to send a command to a specific unit and receive only a single response.

For example, to send a “GS 1” command to unit #2, the string would be:

**SUQ 2<CR>GS 1<CR>**

The response would be the normal response to a “GS 1” command.

See the Set Unit command description for further details on this command.

### **(FU) Flash Unit**

This command momentarily flashes the front panel LEDs of the unit being addressed. This command can be used as a quick test to verify communications. The syntax of the command is:

**FU<CR>**

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

LEDs on units other than the one to which you are physically connected can be flashed by using the Set Unit command, as previously described.

## **Push Commands**

SymNet Devices can send out unsolicited RS-232 and Ethernet data. Using the Assign Controller dialog in SymNet Designer, individual controls can be set up to automatically send out their values whenever they change. Changes occurring on any unit in a SymNet system will be only be sent out the RS-232 and Ethernet port of the master (Ring #1, Unit #1) unit. This method, referred to as pushing data, can be used instead of or in addition to polling. When using this feature, take care that your system can handle the volume of data you set up and that it can differentiate between responses to commands and unsolicited data. Several new commands described below have been added to control this new push feature. Also, the following questions and answers provide a detailed discussion of this feature, including real-world problems and solutions.

### **When is data pushed?**

For data to be pushed, 1) a controller number must be enabled for RS-232 push in SymNet Designer and 2) the push feature has been globally enabled via an RS-232 command. Then, the controller value will be sent out 1) whenever the control’s underlying parameter changes or 2) when a refresh command is issued via RS-232 or Ethernet. Regardless of if the parameter change is made via the GUI, RS-232, RS-485, a preset, a control signal, or any other method, the data will be pushed. This means for example that if your control system changes a controller value set up for push, you will immediately receive notification of that change.

### **Where (out what port) is the data pushed?**

The data is sent out the accessory RS-232 port and the Ethernet port on the Ring #1, Unit #1 SymNet Device. Data for all units is sent out only on this port, not the ports of any other units regardless of where the controller is located. On most SymNet Devices, the accessory serial port is on the rear panel. On some units, the rear-panel port can be switched between accessory and host via DIP switches. The accessory setting must be used. If the master unit does not contain an Ethernet port (e.g. 8x8 DSP), then of course no data is pushed out Ethernet. If the master unit is an Express Cobra that has both an Ethernet and a CobraNet port, the Ethernet port is used.

## What does the pushed data look like?

The format for unsolicited or “push” data is the same as the GSB2 command. Strings consist of the controller number and its value in the following format:

```
#<CONTROLLER NUMBER>=<CONTROLLER POSITION><CR>
```

Where <CONTROLLER NUMBER> is the decimal RS-232/485 controller number (1-10000) assigned in SymNet Designer and <CONTROLLER POSITION> is a 16-bit number in decimal (0-65535). The values for the controller number and position will always be five digits, with leading zeros added as necessary (e.g. 7 would be returned as 00007).

Up to 64 strings, separated with a <CR> as shown, may be sent out together.

### Example:

```
#00007=12321<CR>
```

```
#00324=00128<CR>
```

```
#10000=65535<CR>
```

Why was this feature added? Can't controllers just be polled by the control system?

The push feature was added by popular request from our customers. You asked for it, you got it! Manual polling can still be used instead of or in addition to unsolicited data, but in situations where there are large numbers of parameters that change infrequently, pushing may make more sense. Use whatever method is appropriate for your situation.

## How often is data pushed?

If there is data to be pushed, it is normally sent out every 100 milliseconds. This is called the push interval. In a system with multiple units, data is sent from each unit with push-enabled controllers in a round-robin fashion, again at 100 ms intervals. This means in a system where multiple units are set up to send push data, there could be several push intervals between when a change occurs and when the data is sent out. Note that units in a system that have no data to push will not add any additional delay. While 100 ms is the default, the update interval can be changed via a Set Push Interval command.

## Can I push meter data?

Yes, meters can be assigned for pushing. Keep in mind that with normal audio signals connected to a meter, the meter value will most likely changing constantly, so you will typically see the meter data being pushed every 100 ms interval. However, a Set Push Threshold command can be used to prevent pushes until the data differs by a specified amount (by default, this amount is 1). Alternatively, we suggest using the LED Display modules for metering rather than standard meter modules. They were designed especially with external control in mind. They allow setting up to five customizable thresholds and sending data only when the signal level crosses one of these thresholds. Set either the “LED's on” or “Level” field to push. Yet another option is to use a level detector, peak detector, and/or threshold detector modules instead of directly pushing the meter data.

## How can I control the amount of data pushed?

There are several methods for controlling pushed data. First, since pushed data is enabled on a per-control basis, your first line of defense is to limit it to only certain controls. Second, pushing can be globally turned on and off using an RS-232 command. Third, pushing can be enabled for just a range of controller numbers. Fourth, the Set Push Threshold command can be used to prevent pushes until the data changes by a specified amount. Fifth, the Set Push Interval command controls how often the data is pushed, useful for meters and other data that changes frequently. Finally, the Push Refresh and Push Clear commands provide additional methods of control.

## I want to refresh everything to make sure my control system is synchronized to the hardware. How can I receive all data even if it hasn't changed?

Use the Push Refresh command. Alternatively, you could use the existing Get Controller commands to manually ask for the controls you are concerned with.

**Sometimes my control system turns off push for an extended period of time. When I turn it on, will I be notified of all changes that occurred while push was turned off?**

Yes, by default, all changes made while push was off will be immediately reported as soon as it is turned on. This applies to both turning push off globally or for individual controllers via the Push Disable command. Take care that your system can handle this the potentially large amount of data that can be generated. It may be helpful to “gradually” turn on the push feature, enabling a small range of controller numbers at once. You can also use the Push Clear command to deal with this scenario. It allows you to effectively ignore all previous unreported changes.

**What is the difference between the Global Push Enable/Disable (PU) command and the Push Enable (PUE) and Push Disable (PUD) commands? Why are there 2 different ways to specify a range of controllers?**

The Global Push Enable/Disable command can be used to completely turn off push, or turn on push for all or a single contiguous range of controller numbers. In contrast, the Push Enable/Disable command allows much finer control. Individual (non-contiguous) controllers can be turned on and off, hence multiple ranges are supported.

The reason both methods are provided is for backwards compatibility. The less flexible “single range” global PU command was added first. Later, the more flexible PUE and PUD commands were added as an enhancement. The older global method was left in so existing programs wouldn’t need to be modified. We recommend that you use either one system or the other exclusively. Don’t combine them. New designs should use the PUE and PUD commands and never use the PU command with a range specified.

**What is the difference between the Push Enable (PUE) and Push Disable (PUD) commands and the “Push Value Out RS-232 port” checkbox in SymNet Designer?**

Both of these methods can be used to enable or disable an individual controller for push. The difference is that changing the setting in SymNet Designer requires re-downloading the site file, whereas the Push Enable and Push Disable commands take effect immediately and are intended to be used for “on the fly” changes. In addition, SymNet Designer checkbox settings are permanent across power cycling, whereas changes made with the Push Enable and Push Disable commands are reset by a power cycle (all controllers default to enabled). Please note that the checkbox acts as an override so it must be enabled for a controller ever to push. You cannot issue a Push Enable command for a parameter that doesn’t have the checkbox checked.

**How does push work at power-up?**

When a SymNet Device is first powered up, push is globally turned on, all controllers are enabled, and all controller numbers are assumed to have changed. This means that right after power-up, you will receive a full dump of all data set to be pushed. This can be prevented by issuing a Push Clear command before enabling push.

**I’m not receiving unsolicited data. Any suggestions for troubleshooting?**

1. First of all, make sure that general communication is working between your control system and the SymNet accessory serial port or Ethernet port. Make sure you can send commands and receive ACK messages.
2. Make sure the parameters you want pushed have the “Push Value Out RS-232 port” box checked in SymNet Designer. You can verify this using the Controller Manager, making sure there is a ‘Yes’ in the Push RS-232 column. Be sure you have performed a full-site download (F4) after making any changes.
3. With a serial connection, make sure you are connected to the correct RS-232 port on the Ring #1, Unit #1 SymNet Device. Data for all units is sent out only on this port, not the ports of

any other units regardless of where the controller is located. For Ethernet, make sure the Ethernet port is connected to the same network as the control system. Verify the connection LED on the Ethernet jack and/or switch is lit.

4. Make sure the push feature has been enabled using the Push Enable/Disable command. Push is enabled by default on power-up, but may be turned off via RS-232 or Ethernet. Power cycling the unit is a quick way to verify this.
5. Make sure the parameter to be pushed is changing. Change the parameter via the GUI, an RS-232 command, or other method. You can also use the Push Refresh command to force the data to be sent. If you have changed the push threshold, make sure the parameter is changing by an amount larger than the threshold.
6. For Ethernet, the SymNet Device needs to know the proper IP address to send the data. Make sure at least one command has been sent from the control system to the SymNet Device. If the control system ever changes IP addresses, another command must be sent to establish the new address.

## What are the limitations of this feature?

A given unit may have up to 1024 controllers assigned to be pushed. If multiple parameters change at the same time, up to 64 controller numbers will be sent out during each push interval (default 100 ms) until all have been sent out. This maximum rate may be further limited by your RS-232 baud rate. If a large amount of data is being pushed, we recommend using the highest baud rate your system can support. A baud rate of 115200 is sufficient to prevent further limiting of the throughput.

There is no system-wide limitation to the number of controllers beyond the limit of 1024 per unit. But keep in mind that as more units must push data, the overall update rate will slow.

## Commands related to push:

### (PU) Global Push Enable/Disable

This command enables or disables the push feature. When enabling, a range of controllers can be specified to allow pushing only certain values. Disabling is always global and prevents any unsolicited data from being pushed. The syntax of the command is:

**PU <ON/OFF> [<LOW> [<HIGH>]]<CR>**

Where <ON/OFF> is 0=OFF and 1=ON, <LOW> is the optional lowest controller number to push (only valid when enabling) and <HIGH> is the optional highest controller number to push (only valid when enabling). <LOW> and <HIGH> are both decimal RS-232/485 controller numbers (1-10000) assigned in SymNet Designer. If no controller numbers are specified, the entire range of 1-10000 will be enabled for push. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be pushed. If two controller numbers are specified, the range formed by those values (including the values themselves) will be enabled for push. <LOW> must be less than or equal to <HIGH>. When enabling, the range specified overrides any previous ranges, i.e. it replaces the range, rather than adding to it.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

At power-on, push is always enabled. Remember that individual controller numbers must have the 'Push Value Out RS-232 port' box checked in SymNet Designer. Data is pushed whenever a change in that controller occurs or if forced to refresh using the Push Refresh command.

This command affects all devices in a system and should only be issued to the Ring #1, Device #1 master device.

**Note:** Global Push Enable with a range specified, e.g. "PU 1 100 200<CR>" is not recommended for new designs.

Instead, we recommend always globally enabling the entire range using “PU 1<CR>” and using the Push Enable command for individual control.

#### **(PUE) Push Enable**

This command enables the push feature for an individual controller or range of controllers. The syntax of the command is:

**PUE [<LOW> [<HIGH>]]<CR>**

Where <LOW> is the optional lowest controller number to push and <HIGH> is the optional highest controller number to push. <LOW> and <HIGH> are both decimal RS-232/485 controller numbers (1-10000) assigned in SymNet Designer. If no controller numbers are specified, the entire range of 1-10000 will be enabled for push. If only one controller number is specified, only that controller number is enabled. If two controller numbers are specified, the range formed by those values (including the values themselves) will be enabled for push. <LOW> must be less than or equal to <HIGH>. Multiple PUE commands can be used to enable non-contiguous controller numbers since changes are additive.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

At power-on, push is enabled for all controllers in SymNet systems. Remember that individual controller numbers must have the ‘Push Value Out RS-232 port’ box checked in SymNet Designer. Data is pushed whenever a change in that controller occurs or if forced to refresh using the Push Refresh command. Changes that happen while a control is disabled will be pushed immediately upon enabling that control. The Push Disable command is the inverse of this command and provides a way to turn off controllers for push.

This command affects all devices in a system and should only be issued to the Ring #1, Device #1 master device.

#### **(PUD) Push Disable**

This command enables the push feature for an individual controller or range of controllers. The syntax of the command is:

**PUD [<LOW> [<HIGH>]]<CR>**

Where <LOW> is the optional lowest controller number to stop pushing and <HIGH> is the optional highest controller number to stop pushing. <LOW> and <HIGH> are both decimal RS-232/485 controller numbers (1-10000) assigned in SymNet Designer. If no controller numbers are specified, the entire range of 1-10000 will be disabled for push. If only one controller number is specified, only that controller number is disabled. If two controller numbers are specified, the range formed by those values (including the values themselves) will be disabled for push. <LOW> must be less than or equal to <HIGH>. Multiple PUD commands can be used to disable non-contiguous controller numbers since changes are subtractive.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

At power-on, push is enabled for all controllers in SymNet systems. Remember that individual controller numbers must have the ‘Push Value Out RS-232 port’ box checked in SymNet Designer. The Push Enable command is the inverse of this command and provides a way to turn on controllers for push.

This command affects all devices in a system and should only be issued to the Ring #1, Device #1 master device.

## (GPU) Get Push-enabled Controllers

This command returns a list of all controllers currently enabled for push on the addressed device. A range may optionally be specified to limit the display to controllers enabled for push within that range. The syntax of the command is:

**GPU [<LOW> [<HIGH>]]<CR>**

Where <LOW> is the optional lowest controller number to enquire about and <HIGH> is the optional highest controller number to enquire about. <LOW> and <HIGH> are both decimal RS-232/485 controller numbers (1-10000) assigned in SymNet Designer. If no controller numbers are specified, the entire range of 1-10000 will be inquired about. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be enquired about. If two controller numbers are specified, the range formed by those values (including the values themselves) will be enquired about. <LOW> must be less than or equal to <HIGH>.

If the command is accepted, the device will respond with a list of enabled controller numbers separated by <CR>. If no controllers are enabled, it returns the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

Special case: Entering **GPU 0<CR>** will return a list settings related to push. It begins with **Global=<0/1>** to show if push is globally enabled (1) or disabled (0). This is followed by five 5-digit values showing the settings of 1) the global lower limit, 2) the global upper limit, 3) the threshold for parameters, 4) the threshold for meters, and 5) the push interval in milliseconds. The default printout would look like this:

**Global=1<CR>**

**00001 10000 00001 00001 00103<CR>**

## (PUR) Push Refresh

This command causes data to be pushed immediately even if it hasn't changed (assuming push is enabled). This may be useful when trying to synchronize a control system to SymNet. A range of controllers can be specified to refresh only certain values. The syntax of the command is:

**PUR [<LOW> [<HIGH>]]<CR>**

Where <LOW> is the optional lowest controller number to refresh and <HIGH> is the optional highest controller number to refresh. <LOW> and <HIGH> are both decimal RS-232/485 controller numbers (1-10000) assigned in SymNet Designer. If no controller numbers are specified, the entire range of 1-10000 will be refreshed. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be refreshed. If two controller numbers are specified, the range formed by those values (including the values themselves) will be refreshed. <LOW> must be less than or equal to <HIGH>.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

At power-on, all controller values are assumed to have changed, so it acts as if a full refresh was performed. Remember that individual controller numbers must have the 'Push Value Out RS-232 port' box checked for push in SymNet Designer. In addition, push must be enabled for the range of controllers you are refreshing (see Push Enable/Disable). Controller numbers that don't meet these two criteria will not be affected by the Push Refresh command. In other words, if a controller is not enabled for push, refreshing it won't cause the value to be pushed even if that controller is later enabled. The controller must be enabled for push at the time the Push Refresh command is issued. This command affects all devices in a system and should only be issued to the Ring #1, Device #1 master device.

### (PUC) Push Clear

This command causes previous changes in data to be ignored and not pushed. It may be desirable to issue this command when first enabling push to prevent being swamped by the flood incoming data. A range of controllers can be specified to clear only certain values. The syntax of the command is:

**PUC [<LOW> [<HIGH>]]<CR>**

Where <LOW> is the optional lowest controller number to clear and <HIGH> is the optional highest controller number to clear. <LOW> and <HIGH> are both decimal RS-232/485 controller numbers (1-10000) assigned in SymNet Designer. If no controller numbers are specified, the entire range of 1-10000 will be cleared. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be cleared. If two controller numbers are specified, the range formed by those values (including the values themselves) will be cleared. <LOW> must be less than or equal to <HIGH>.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

It may be useful to issue this command if push has been disabled for a long time and then is about to be re-enabled. Otherwise, you will immediately receive notification for all changes that occurred during the disabled time. This command affects all devices in a system and should only be issued to the Ring #1, Device #1 master device.

### (PUI) Set Push Interval

This command changes the minimum length of time between consecutive pushes of data. (See “How often is data pushed?” above for more information.) At power-up, this value defaults to 100 milliseconds. The syntax of the command is:

**PUI <MILLISECONDS><CR>**

where <MILLISECONDS> is the push interval in milliseconds, between 20 ms and 30,000 ms (30 seconds).

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

While setting a short interval can speed up the push response, it may have a negative impact on overall system performance. The shorter the interval, the more time will be spent looking for push data. This can slow down responses to other RS-232 commands and SymNet Designer. In extreme cases, it may cause SymNet Designer to go off-line, especially with multi-ring systems. Therefore, we recommend using the longest interval that is practical, especially if data is being pushed while SymNet Designer is on-line and reading meters, changing parameters, etc. The default value of 100 milliseconds usually provides a good compromise between prompt reports of changing data and overall system performance.

Note that in cases where many controllers are changing rapidly, the serial baud rate may ultimately limit the update rate. Using the highest available baud rate is recommended.

### (PUT) Set Push Threshold

This command changes the push threshold value. Recall that data is only pushed when it changes. The threshold is the amount a value must change from the previous push before it is pushed again. For example, if a controller value was 10,000 and the threshold was 1,000, the data would not be pushed again until the value rose to at least 11,000 or fell to 9,000 or less.

SymNet actually maintains two different thresholds: one for parameter data such as faders and buttons, and another for meters (including LEDs). These two thresholds can be set to the same value or be different. It may be desirable to use a fairly large threshold for meters to avoid constant pushing of values. The power-on default for both of these values is 1.

The syntax of the command is:

**PUT [<PARAMETER THRESH>] [<METER THRESH>]<CR>**

Where **<PARAMETER THRESH>** is the optional threshold for parameters other than meters (e.g. faders and buttons) and **<METER THRESH>** is the optional threshold for meters. Both values must be between 0 and 65535. If neither threshold is specified, both thresholds are set to the default of 1. If only one threshold is specified, that value is used for both the parameter and meter thresholds.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

Technical Note: The threshold is a “greater than or equal to” type parameter, meaning it must be exceeded or met to trigger a push. For example: if the threshold is 1 and the last value pushed was 10,000, then a new value of 10,001 or 9,999 would cause a push to occur. Note that it is possible to set the threshold to zero. In this case, the value will be pushed if there is any change at all to the underlying DSP variable - even if the change is so small that the pushed controller value is identical (which may happen due to the limited resolution of the 16-bit controller value scheme). In fact, in versions of SymNet Designer before 7.0 a threshold of zero was always used, so it is provided for backwards compatibility.

## Setup Commands

**Note:** If you ever find yourself in a situation where you are not sure of the accessory controller port settings, you can use SymNet Designer to return the device to default values. Simply open the Hardware->Upgrade Firmware dialog and click on the Erase Memory button. You can then select to erase the Accessory Serial Port Settings. Doing this will set the baud rate, port number, echo mode and quiet mode settings to the factory default values.

### (SP) Set Port

The Set Port command is used to force the port that will be used for host PC communications (CobraLink, SymNet 8x8, 8in and 8out DSP devices only). The port not used for the host is available for use by accessory controllers. But, in general, when both the host PC and accessory controllers are to be used, the host should be assigned to the front panel port. See the section on RS-232 Port Configuration (page 5) for a full description of the issues involved with this feature. The syntax of the command is:

**SP <PORT NUMBER><CR>**

Where **<PORT NUMBER>** is 0, 1 or 2.

The meanings of these values are as follows:

- 0 - Allow the port sensing algorithm to be used on power up to determine which port will be the host port. If a host PC connection is present at the front port on power up, then the front port will be used for Host PC Control. If a host PC connection is present at the rear port on power up, then the rear port will be used for Host PC Control. This setting requires that the hardware handshaking lines (RTS and CTS) be connected on your serial port cable.
- 1 - (DEFAULT) Force the front panel port to be used only for host PC communications. The rear port will be available for accessory controllers.
- 2 - Force the rear panel port to be used only for host PC communications. The front port will be available for accessory controllers.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

The port state programmed will be saved in non-volatile memory. It does not need to be continually set. It will not hurt the device to be repeatedly set with the same value as it is only written if a different value is set. The new state will not be used until the next time power is cycled. Keep in mind that once programmed, to change the program, you will have to connect to the port currently programmed for the accessory controllers in order for commands to be accepted. If you do not get a response to a command, you should try the other port in case the port has been previously forced with this command.

#### **(SB) Set Baud**

The Set Baud command controls the accessory controller port's baud rate. This is normally the rear RS-232 port.

The syntax of the command is:

**SB <BAUD><CR>**

Where <BAUD> is 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 or 230400.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

The baud rate will be saved in non-volatile memory. It does not need to be continually set. It will not hurt the device to be repeatedly set with the same value as it is only written if a different value is set. The new state will not be used until the next time power is cycled. Note: New devices default to a baud rate of 38400.

**NOTE:** The 230400 setting is only recommended for use on SymNet Express and DigIO devices. On 8x8, 8in, 8out, and CobraLink devices, this baud rate may not be accurate enough for reliable communications.

#### **(SQ) Set Quiet Mode**

The Set Quiet Mode command controls the text output of the control port during responses. When quiet mode is turned on, it restricts the output to just ACK, NAK or simple values. All command descriptions above assume that quiet mode is turned on. Quiet mode ON should be used when using a controller with RS-232.

When quiet mode is set to OFF, lengthy strings intended to be read by humans are sent in response to commands. This mode is useful when using a terminal program for testing or debugging.

The syntax of the command is:

**SQ <ON/OFF><CR>**

Where <ON/OFF> is 0 = OFF, 1 = ON.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

The quiet mode state will be saved in non-volatile memory. It does not need to be continually set. It will not hurt the device to be repeatedly set with the same value as it is only written if a different value is set. Note: New devices default to quiet mode ON.

### (EH) Set Echo Mode

The Set Echo Mode command controls the text output of the control port during commands. When echo mode is turned on, all characters that are received on the RS-232 port are sent or “echoed” back. This mode is useful when using a terminal program for testing or debugging.

When echo mode is turned off, the characters received are not echoed back. All command descriptions above assume that echo mode is turned off. Echo mode OFF should be used when using a controller with RS-232.

The syntax of the command is:

**EH <ON/OFF><CR>**

Where <ON/OFF> is 0 = OFF, 1 = ON.

If the command is accepted, the SymNet Device will respond with the string:

**ACK<CR>**

If the command is interpreted but fails for any reason the SymNet Device will respond with the string:

**NAK<CR>**

The echo mode state will be saved in non-volatile memory. It does not need to be continually set. It will not hurt the device to be repeatedly set with the same value as it is only written if a different value is set. Note: New devices default with echo mode OFF.



 **SymNet™** | Network Audio Solutions

6408 216th St. SW | Mountlake Terrace, WA 98043 | USA | Tel: +1 (425) 778.7728 | Fax: +1 (425) 778.7727

Engineered by **Symetrix™**

**External Control Protocol: SymNet Designer Rev 8.x**

© 2008 Symetrix, Inc. All rights reserved. Printed in the United States of America. The information in this document is subject to change without notice. Symetrix, Inc. shall not be liable for technical or editorial errors or omissions contained herein; nor is it liable for incidental or consequential damages resulting from the furnishing, performance, or use of this material. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Symetrix assumes no responsibility with regard to the performance or use of these products. Under copyright laws, no part of this brochure may be reproduced or transmitted in any form or by any means, electronic or mechanical, without permission in writing from Symetrix, Inc. If, however, your only means of access is electronic, permission to print one copy is hereby granted. The following are either Trademarks or Registered Trademarks of Symetrix, Inc.: Symetrix, SymNet, SymNet Designer, SymLink and CobraLink. Windows is a Registered Trademark of Microsoft, Inc. Other product names mentioned herein may be trademarks and/or registered trademarks of other companies and are property of their respective owners.