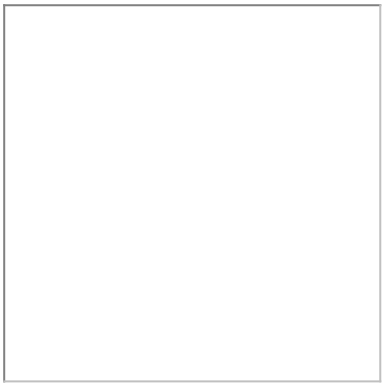

A problem has occured

2020-10-01 10:10:33

Error message:

Insufficient license packs. This license can only be used with 4 CPU cores. 8 CPU cores have been detected by Java. 1 additional license pack(s) are required.

License Information:

License serial no:	4363
Licensee:	Shure Incorporated
Street:	5800 West Touhy Avenue
City:	60714-4608 Niles, IL
Country:	United States
Product:	PDFreactor
Version:	10.0
License Type:	CPU
Amount:	4 CPU(s)
Maintenance Exp. Date:	2019-11-02
Purchase Date:	2017-10-30
Sign Date:	2019-03-06 09:25

SLX-D

Command Strings

Third-party command strings for Shure SLX-D wireless system
Version: 2.0 (2020-J)

REALOBJECTS

PDFreactor®

Evaluation Version

This PDF document was created by an evaluation version of RealObjects PDFreactor 10.0.10722.2. The evaluation version is fully functional, but includes this information page. It must not be used for production purposes. The information page and all other evaluation notices must not be removed from the PDF file.

NOTE: Conversions in evaluation mode might be slower and the results might have a larger file size than in production mode.

Buy PDFreactor

PDFreactor has detected 8 CPU cores, which means you need 2 license packs to use PDFreactor.

To buy a PDFreactor license follow this link:

Buy PDFreactor online

About PDFreactor

RealObjects PDFreactor is a powerful formatting processor for converting HTML and XML documents into PDF. It uses Cascading Style Sheets (CSS) to define page layout and styles. The server-side tool enables a great variety of applications in the fields of ERP, eCommerce and Electronic Publishing.

PDFreactor supports HTML5, CSS3 and JavaScript.

It allows you to dynamically generate PDF documents such as invoices, delivery notes and shipping documents on-the-fly. PDFreactor allows you to easily add server-based PDF generation functionality to your application or service. Since PDFreactor runs on a server, the end-user in general does not need any software other than a PDF viewer.

For more information visit **www.pdfreactor.com**

Table of Contents

SLX-D Command Strings

The SLX-D device is connected via Ethernet to a control system, such as

- AMX, Crestron or Extron
- Symetrix, Biamp, other digital signal processors (DSP)
- Specialized custom programs

Connection: Ethernet (TCP/IP; select “Client” in the AMX/Crestron program)
Port: 2202

Conventions

There are 4 types of strings:

GET	Finds the status of a property. After the AMX/Crestron sends a GET command, the system responds with a REPORT string
SET	Changes the status of a property. After the AMX/Crestron sends a SET command, the system responds with a REPORT string to indicate the new value of the property.
REP	When the system receives a GET or SET command, it replies with a REPORT command to indicate the status of the property. **Important:** With the exception of the metered properties, the device sends a REPORT when a value changes. Thus, it is not necessary to constantly query most device properties.
SAMPLE	Used for metering audio levels.

All messages sent and received are ASCII. Note that the level indicators and gain indicators are also in ASCII

The character "x" in all of the following strings represents the channel and can be ASCII numbers 0 through 4 as in the following table.

0	All channels
1, 2	Individual channels

Device Command Strings

ALL

Description	Discovery of device properties.

Commands	< GET x ALL > < REP ... >
Variables	When **x** is zero, the device responds with REP for all device-specific properties and ALL channel-related properties including all metered properties. When **x** is a channel number, the device responds with REP for all device-specific properties and ALL channel **x**-related properties including all metered properties.
Notes	None.

FLASH

Description	Controls the flash to identify a device or channel.
Commands	< SET FLASH ON > < REP FLASH ON > Device initiates an identify then stops flashing: < REP FLASH OFF > **Note:** When used with no channel index the command initiates a Device Identify. When used with a channel index the command initiates a Channel Identify. < SET x FLASH ON > < REP x FLASH ON >
Variables	When used, **x** is the channel number.
Notes	None.

MODEL

Description	Discovery of the model name of the device.
Commands	< GET MODEL > < REP MODEL {SLXD4yyyyyyyyyyyyyyyyyyyyyyyyyyyy} >
Variables	Where the repeating **y** represents the spaces returned by the device to pad the model name to 32 characters.
Notes	The device always responds with a 32-character model name.

DEVICE_ID

Description	Controls the Device ID.

Commands	< GET DEVICE_ID > < REP DEVICE_ID {SLXD4yyy} > < SET DEVICE_ID {Name1} > < REP DEVICE_ID {Name1yyy} >
Variables	Where the repeating **y** represents the spaces returned by the device to pad the Device ID to 8 characters.
Notes	The device always responds with 8-character ID. SET accepts 1-8 Characters from the set: A-Z,a-z,0-9,!"#$%&'()*+,-./:;<=>?@[\]^_`~ and space.

RF_BAND

Description	Discovery of the RF band.
Commands	< GET RF_BAND > < REP RF_BAND {G55yyyyy} >
Variables	Where the repeating **y** represents the spaces returned by the device to pad the response to 8 characters.
Notes	None.

LOCK_STATUS

Description	Discovery of the transmitter Lock.
Commands	< GET LOCK_STATUS > < REP LOCK_STATUS ALL >
Variables	None.
Notes	Report responses: OFF MENU ALL

FW_VER

Description	Discovery of the firmware version.
Commands	Self test passed:

	< GET FW_VER > < REP FW_VER {2.0.15.2yyyyyyyyyyyyyyyy} > Self test failed: < GET FW_VER > < REP FW_VER {2.0.15.2*yyyyyyyyyyyyyyy} >
Variables	Where the repeating **y** represents the spaces returned by the device to pad the response to 24 characters.
Notes	Package version number reported as Maj.Min.Pack.Build.

Channel Command Strings

CHAN_NAME

Description	Control for the channel name.
Commands	< GET x CHAN_NAME > < REP x CHAN_NAME {yyyyyyyyyyyyyyyyyyyyyyyyyyyyyy} > < SET x CHAN_NAME {Lead Vox} > < REP x CHAN_NAME {Lead Voxyyyyyyyyyyyyyyyyyyyyyyy} >
Variables	Where **x** is the channel number. Where the repeating **y** represents or pads the 31-character string from the set: A-Z,a-z, 0-9,!"#$%&'()*+,-./:;<=>?@[\]^_`~ and space, that is, {12345678901234567890123456789 01}.
Notes	SET only supports 8 characters. The device always responds with a 31-character name.

AUDIO_GAIN

Description	Control for the channel audio gain.
Commands	< GET x AUDIO_GAIN > < REP x AUDIO_GAIN 030 > There is an offset of 18 so the actual value = 30 - 18 = 12 dB. To set to 22 dB:

	< SET x AUDIO_GAIN 40 > < REP x AUDIO_GAIN 040 > To decrement the value down 5 dB: < SET x AUDIO_GAIN DEC 5 > < REP x AUDIO_GAIN 035 > To increment the value up 10 dB: < SET x AUDIO_GAIN INC 10 > < REP x AUDIO_GAIN 045 >
Variables	Where **x** is the channel number.
Notes	Numeric 3 Characters 000 to 060 in increments of 1 The values REPorted and SET are offset by 18 Actual range: -18 to 42 dB in 1 dB steps

AUDIO_OUT_LVL_SWITCH

Description	Discovery of the current Mic/Line switch status.
Commands	< GET x AUDIO_OUT_LVL_SWITCH > < REP x AUDIO_OUT_LVL_SWITCH MIC >
Variables	Where **x** is the channel number.
Notes	"MIC" - Audio output is at microphone level (default) "LINE" - Audio output is at line level

GROUP_CHANNEL

Description	Controls the group channel mappings.
Commands	< GET x GROUP_CHANNEL > < REP x GROUP_CHANNEL {1,1yy} > < SET x GROUP_CHANNEL {6,100} > < REP x FREQUENCY 0652875 > < REP x GROUP_CHANNEL {6,100} >
Variables	Where **x** is the channel number.

	Where the repeating **y** represents the spaces returned by the device to pad the response to 5 characters.
Notes	Device always returns a 5-character string Refer to the Group/Channel mappings corresponding to the RF Band and Transmission Mode of the device You must parse the "," from within the reported value Characters before the "," are the Group ID Characters after the "," are the Channel ID The value: "--,--" is the wildcard indicating no GROUP_CHANNEL value is set You cannot SET to "--,--" **Note:** GROUP_CHANNEL and FREQUENCY are related as described in FREQUENCY.

FREQUENCY

Description	Controls frequency settings.
Commands	< GET x FREQUENCY > < REP x FREQUENCY 0578350 > < SET x FREQUENCY 602125 > < REP x GROUP_CHANNEL {--,--yyyyy} > (Note 1) < REP x FREQUENCY 0602125 >
Variables	Where **x** is the channel number. Where the repeating **y** represents the spaces returned by the device to pad the response to 7 characters.

Notes	Frequency and Group Channel
	
	FREQUENCY - Device always returns a 7-character, numeric string
	GROUP_CHANNEL - Device always returns a 10-character string
	Range and Step per the RF Band
	
	1. GROUP_CHANNEL and FREQUENCY are related:
	    1. Setting FREQUENCY results in the GROUP_CHANNEL value reverting to "--,--" if not already indicating "--,--" in addition to the FREQUENCY value.
	    2. Setting GROUP_CHAN results in the corresponding FREQUENCY value being reported in addition to the GROUP_CHAN value.
	Commands:
	
	Starting from a default condition:
	
	< GET x GROUP_CHANNEL >
	< REP x GROUP_CHANNEL {1,1yyyyyyy} >
	< GET x FREQUENCY >
	< REP x FREQUENCY 0606025 >
	
	SET the FREQUENCY to some new value:

	< SET x FREQUENCY 620000 > < REP x GROUP_CHANNEL {--,--yyyyy} > < REP x FREQUENCY 0620000 > Similarly, when setting GROUP_CHANNEL, the corresponding FREQUENCY is reported: < SET x GROUP_CHANNEL {6,6} > < REP x FREQUENCY 0614650 > < REP x GROUP_CHANNEL {6,6yyyyyyy} >

Metering Command Strings

The majority of properties generate REP (Report) messages when their values change (for example, Frequency, Channel Name, and so on.)

For attributes such as audio meters, RF meters, channel quality meters, and the like, a REP on each change is inefficient and can flood many simple control systems.

The Shure approach is to use metering to periodically sample your channels and devices:

- You can still use GET to discover a value when necessary.
- Combine the metered attributes into a single SAMPLE message per channel.

  for example,

  < SAMPLE chNum ALL audPeak audRms rfRssi >

  returns

  < SAMPLE 1 ALL 102 102 086 >

  where each field is documented and easy to parse.
- Generate periodic SAMPLE messages at the interval set via the METER_RATE.

  Note: *To turn off sampling, use*

  < SET x METER_RATE 00000 >

  where **x** is the channel number.

The following sections detail METER_RATE and SAMPLE followed by the set of metered attributes.

METER_RATE

Description	Controls the meter rate.
Commands	< GET x METER_RATE > < REP x METER_RATE 00000 > < SET x METER_RATE 01000 > < REP x METER_RATE 01000 > **Note:** This results in one SAMPLE every second.

Variables	Where **x** is the channel number.
Notes	Numeric, 5-character fixed output. 00000 - Metering OFF (default) 00100 to 65535 - The interval of each SAMPLE report in milliseconds. For example, 00100 - Sample every 10 millisecond (10 samples per sec) 01000 - Sample every second 05000 - Sample every 5 seconds

SAMPLE

Description	Specifies which of the Metering Commands you want to sample.
Commands	Sample input: **Note:** The sample input that follows is actually 1 line of input spaced to show correlation. < SAMPLE chNum ALL audPeak audRms rfRssi > Sample response: < SAMPLE 1 ALL 102 102 086 >
Variables	Key mapping:

Key	Corresponding command string for value format	Num Char	Notes
audPeak	AUDIO_LEVEL_PEAK	3	
audRms	AUDIO_LEVEL_RMS	3	
rfRssi	RSSI	3	

Notes	None.

AUDIO_LEVEL_PEAK

Description	Discovery of peak audio level.
Commands	< GET x AUDIO_LEVEL_PEAK > < REP x AUDIO_LEVEL_PEAK 102 >
Variables	Where **x** is the channel number

Notes	This is a metered property using SAMPLE; it does not generate a report on a value change. Numeric, 3-character fixed output Units: dBFS The actual value = the reported value - 120 Value range as reported over command strings: 000 - 120 Value range after conversion to the actual value: -120 - 0 dBFS The SLX-D values fall in the range of about -100 to 0 dBFS.

AUDIO_LEVEL_RMS

Description	Discovery of the RMS audio level.
Commands	< GET x AUDIO_LEVEL_RMS > < REP x AUDIO_LEVEL_RMS 102 >
Variables	Where **x** is the channel number.
Notes	This is a metered property using SAMPLE; it does not generate a report on a value change. Format: Numeric, 3-character fixed output Units: dBFS The actual value = the reported value - 120 Value range as reported over command strings: 000 - 120 Value range after conversion to the actual value: -120 - 0 dBFS The SLX-D values fall in the range of about -100 to 0 dBFS.

RSSI

Description	Discovery of the RSSI.
Commands	< GET x RSSI 0 > < REP x RSSI 1 083 > < REP x RSSI 2 064 >
Variables	Where **x** is the channel number.
Notes	This is a metered property using SAMPLE; it does not generate a report on a value change. Numeric, 3-character fixed output per antenna Units: dBm The actual value = the reported value - 120 Value range as reported over command strings: 000 - 120 Value range after conversion to the actual value: -120 - 0 dBm

Side Channel Command Strings

TX_MODEL

Description	Discovery of the transmitter model.
Commands	< GET x TX_MODEL > < REP x TX_MODEL UNKNOWN > Report of a transmitter being received: < REP x TX_MODEL SLXD2 >
Variables	Where **x** is the channel number.
Notes	Report responses: SLXD1 SLXD2 UNKNOWN

TX_BATT_MINS

Description	Discovery of the transmitter battery runtime minutes.
Commands	< GET x TX_BATT_MINS > < REP x TX_BATT_MINS 65535 > Report when data becomes known (example is 2 hours 5 minutes): < REP x TX_BATT_MINS 00125 >
Variables	Where **x** is the channel number.
Notes	Numeric, 5-character fixed output 00000 to 65532 - Number of minutes of runtime 65533 - Battery communication warning 65534 - Battery time calculating 65535 - Unknown, or not applicable

TX_BATT_BARS

Description	Discovery of the transmitter Battery Bars.
Commands	< GET x TX_BATT_BARS > < REP x TX_BATT_BARS 255 >

	Report when data becomes known: < REP x TX_BATT_BARS 004 >
Variables	Where **x** is the channel number.
Notes	Numeric, 3-character fixed output 000 to 005 255 - Unknown