



DIGILINX[™]

Programming for Third
Party Control of
DigilinX

Title: Programming for Third Party Control of *DigiLinX*

Document Number: 020004

Original Publication Date: May 11, 2006

All rights reserved.

Copyright © 2006 by *NetStreams*.

All brand names, product names, and trademarks are properties of their respective owners.



3600 W. Parmer Lane, Suite 100

Austin, TX 78727

USA

Phone: +1 512.977.9393

Fax: +1 512.977.9398

Toll Free Technical Support 1-866-353-3496

Contents

Chapter 1: Introduction	1-1
Communicating with the DigiLinX System	1-1
Chapter 2: Services	2-1
Service Names	2-1
Service Types	2-1
Audio Renderer	2-2
Audio Source	2-2
Audio/Local Source	2-2
Audio Stream Proxy	2-2
User Interface (UI)	2-2
Intercom	2-2
General Purpose IO (GPIO)	2-2
Root.	2-3
List of Services by Device	2-3
SpeakerLinX Services	2-3
MediaLinX™ Services	2-3
TouchLinX Services	2-3
ControLinX Services	2-3
Configuration	2-3
Chapter 3: ASCII Overview	3-1
StreamNet Addressing	3-1
Subscribing	3-2
ASCII Message Syntax	3-2
@toAddress	3-2
Node	3-3
UniqueID (UID)	3-3
Subnode	3-3
StreamNet Subnode Addresses:	3-4
:fromAddress	3-4
%modifier	3-4
#keyword (ASCII Command)	3-4
arg1, arg2, ..., argN	3-4
/0 (null byte)	3-5
Summary	3-5
ASCII Message Types	3-5
Imperatives	3-5
Interrogatives	3-6
Declaratives	3-6

Menus	3-6
Chapter 4: <i>SpeakerLinX</i> Commands	4-1
<i>SpeakerLinX</i> Audio Renderer Type Commands	4-2
Command: #ACTIVE OFF	4-2
Command: #ACTIVE ON	4-2
Command: #AMP OFF	4-3
Command: #AMP ON	4-3
Levels	4-3
Command: #LEVEL_SET BALANCE, x	4-3
Command: #LEVEL_DN BALANCE	4-4
Command: #LEVEL_UP BALANCE 1	4-4
Command: #LEVEL_SET BASS, x	4-4
Command: #LEVEL_DN BASS	4-4
Command: #LEVEL_UP BASS	4-4
Command: #LEVEL_SET TREB, x	4-4
Command: #LEVEL_DN TREB	4-5
Command: #LEVEL_UP TREB	4-5
Command: #LEVEL_SET VOL, x	4-5
Command: #LEVEL_DN VOL	4-5
Command: #LEVEL_UP VOL	4-5
Command: #LEVEL_SET BAND_X, Y	4-5
Command: #LEVEL_DN BAND_X	4-6
Command: #LEVEL_UP BAND_X	4-6
Command: #MUTE OFF	4-6
Command: #MUTE ON	4-6
Command: #MUTE TOGGLE	4-6
Command: #MENU_LIST m,n,SOURCES	4-6
Responses Received.....	4-7
Command: #MENU_SEL {{path}}	4-8
Command: #SRC_SEL {{source id}}	4-9
Command: #SRC_SEL	4-9
Command: #SRC_SEL NEXT	4-9
Command: #QUERY RENDERER	4-9
Parsing Reports	4-10
Command: #QUERY CURRENT_SOURCE	4-10
<i>SpeakerLinX</i> Media Server Audio Stream Type Commands	4-11
Command: #MENU_LIST m,n,{{path}}	4-11
Responses Received.....	4-11
Summary	4-12
Command: #MENU_LIST m,n,{{path}}	4-12
Responses Received.....	4-13
Command: #MENU_SEL {{path}}	4-14
Command: #NEXT	4-14
Command: #PAUSE	4-14
Command: #PLAY	4-14
Command: #PREV	4-15
Command: #SHUFFLE	4-15
Command: #STOP	4-15
Command: #QUERY SOURCE	4-15

Response Received	4-15
SpeakerLinX Intercom serviceType Commands	4-16
Setup Program Guidelines	4-16
Command: #INTERCOM JOIN, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVR- RD] [,IMMED]	4-16
Command: #INTERCOM JOINMON, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD]	4-17
Command: #INTERCOM MONITOR, roomName, audience	4-18
Command: #INTERCOM LEAVE, [tag]	4-18
Command: #INTERCOM LEAVE, monitor	4-18
Command: #INTERCOM LEVEL_UP VOL	4-18
Command: #INTERCOM LEVEL_DN VOL	4-18
Command: #INTERCOM LEVEL_SET VOL, x	4-19
Command: #INTERCOM PTT, [tag]	4-19
Command: #INTERCOM PTT OFF	4-19
Command: #DND x	4-19
Command: #PRIVACY x	4-19
Sample Intercom Scenarios	4-20
Simple Scenario	4-20
Complete Control Scenario.....	4-20
SpeakerLinX Root Type Commands	4-21
Command: #QUERY SERVICE	4-21
Response Received	4-21
Command: #QUERY SERVICE {{serviceName}}	4-21
Response Received	4-22
Response Received	4-22
Command: #@Room 2#Query Service	4-22
Responses Received.....	4-22
Command: #MENU_LIST m,n,RENDERERS	4-23
Responses Received.....	4-24
Case Sensitivities	4-24
Chapter 5: MediaLinX Commands	5-1
CD Player Commands	5-4
Command: #PREV	5-4
Command: #NEXT	5-4
Command: #PWR ON	5-4
Command: #PWR OFF	5-4
Command: #PLAY	5-4
Command: #PAUSE	5-4
Command: #STOP	5-4
Command: #REV	5-5
Command: #FWD	5-5
Command: #DISC PREV	5-5
Command: #DISC NEXT	5-5
Command: #RANDOM	5-5
Command: #REPEAT TOGGLE	5-5
Command: #KEY x	5-5

Command: #Fx	5-6
Command: #KEY x_PLUS	5-6
Command: #CLEAR	5-6
Command: #ENTER	5-6
TUNER Commands	5-6
Command: #PREV	5-6
Command: #NEXT	5-6
Command: #PWR ON	5-7
Command: #PWR OFF	5-7
Command: #TUNE x	5-7
Command: #SEEK x	5-7
Command: #SCAN x	5-7
Command: #PRESET x	5-7
Command: # BAND NEXT	5-8
Command: #KEY x	5-8
Command: #CLEAR	5-8
Command: #ENTER	5-8
DVD Player Commands	5-8
Command: #PWR ON	5-8
Command: #PWR OFF	5-8
Command: #PLAY	5-8
Command: #PAUSE	5-9
Command: #STOP	5-9
Command: #PREV	5-9
Command: #NEXT	5-9
Command: #REV	5-9
Command: #FWD	5-9
Command: #TOPMENU	5-10
Command: #MENU	5-10
Command: #SELECT	5-10
Command: #SETUP	5-10
Command: #DISPLAY	5-10
Command: #DVDAUDIO	5-10
Command: #SUBTITLE	5-11
Command: #LANG	5-11
Command: #MODE	5-11
Command: #EJECT	5-11
Command: #ANGLE	5-11
Command: #ZOOM	5-11
Command: #RETURN	5-12
Command: #NAV x	5-12
Command: #KEY x	5-12
Command: #Fx	5-12
Command: #KEY x_PLUS	5-12
Command: #CLEAR	5-12
Command: #ENTER	5-13
Source ServiceType Commands	5-13
Command: #SET AUDIO, ANALOG	5-13
Command: #SET AUDIO, GAIN, x	5-13

Command: #SET AUDIO, SPDIF	5-13
Command: #MENU_LIST m,n,{{path}}	5-14
Request a List of Media	5-14
Traversing the Menu List.....	5-15
Listing Source Presets.....	5-17
Command: #MENU_SEL {{path}}	5-18
Selecting a Song from the Menu List	5-18
Selecting a Preset	5-18
Command: #QUERY SOURCE	5-18
Response Received	5-18
Root ServiceType Commands	5-19
Response Received	5-19
Command: #QUERY SERVICE {{serviceName}}	5-19
Command: #RESTART	5-20
Chapter 6: TouchLinX Commands	6-1
UI/GUI Type Commands	6-1
Command: #ACTIVE OFF	6-1
Command: #ACTIVE ON	6-1
Command: #QUERY UI	6-2
Response Received	6-2
Command: #SET BKLT, TIMEOUT, x	6-2
Command: #LEVEL_UP BKLT	6-2
Command: #LEVEL_DN BKLT	6-2
Command: #LEVEL_SET VOL, x	6-2
Intercom ServiceType Commands	6-2
Setup Program Guidelines	6-2
Command: #INTERCOM JOIN, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVR- RD] [,IMMED]	6-3
Command: #INTERCOM JOINMON, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD]	6-4
Command: #INTERCOM MONITOR, roomName, audience	6-4
Command: #INTERCOM LEAVE, [tag]	6-5
Command: #INTERCOM LEAVE, monitor	6-5
Command: #INTERCOM LEVEL_UP VOL	6-5
Command: #INTERCOM LEVEL_DN VOL	6-5
Command: #INTERCOM LEVEL_SET VOL, x	6-5
Command: #INTERCOM PTT, [tag]	6-5
Command: #INTERCOM PTT OFF	6-6
Command: #DND x	6-6
Command: #PRIVACY x	6-6
Sample Intercom Scenarios	6-6
TouchLinX Root ServiceType Commands	6-7
Command: #QUERY SERVICE	6-7
Command: #QUERY SERVICE {{serviceName}}	6-8
Command: #MENU_LIST m,n,INTERCOM	6-9
Command: #MENU_LIST m,n,RENDERERS	6-9
Response Received	6-9

Command: #Restart	6-10
Case Sensitivities	6-10
Chapter 7: <i>ControLinX</i> Commands	7-1
<i>ControLinX</i> GPIO Type Commands	7-1
Command: #TXSERIAL	7-1
<i>ControLinX</i> Root ServiceType Commands	7-2
Command: #QUERY SERVICE	7-2
Response Received	7-2
Command: #QUERY SERVICE {{serviceName}}	7-2
Response Received	7-2
Command: #RESTART	7-3

Index

Introduction

The scope of this manual is to provide details on how to integrate a third party controller with a *NetStreams DigiLinX*[™] IP-Based Multi-Room Audio Control system.

It is assumed that the user understands User Datagram Protocol (UDP) and Unicast and Multicast addressing.

It is also assumed that the user has used the *NetStreams, DigiLinX Dealer Setup* tool to configure the *DigiLinX* system.

The following document explains how to communicate with the *DigiLinX* system and integrate a third party controller with the *DigiLinX* system.

Communicating with the *DigiLinX* System

StreamNet[™], *NetStreams* protocol, uses ASCII messages to communicate with services associated with the devices in the *DigiLinX* system. These messages are either control or status messages.

Services

In the *DigiLinX* system, system/device functions are grouped together into logical constructs called services. A single *DigiLinX* device may host multiple services. All communication with *DigiLinX* is addressed to one or more services hosted by the individual devices.

ASCII messages are not addressed to devices but to virtual services which reside on the devices. This scheme alleviates the need to know the IP addresses of each *DigiLinX* device.

Service Names

Every service must have a unique name which can be used to address ASCII messages. Every service may have a room name associated with it. Services may also be assigned to multiple group names. Service names, room names, and group names may all be used to address ASCII messages. If a room name is used as an address, all services assigned to that room will react. If a group name is used, all services assigned to that group will react. There is a special group name (All) to which all services will react.

Service Types

Each physical *DigiLinX* device may be configured to have some or all services enabled on it.

The service types supported are:

- Audio Renderer,
- Audio Source,
- Audio/Local Source,
- Audio Stream Proxy,
- User Interface (UI),
- Intercom,
- General Purpose IO (GPIO), and

- Root.

Audio Renderer

The audio renderer service is responsible for receiving audio off the network and creating the line or amplified audio output. This service may have various controls such as volume, bass, treble, etc.

Audio Source

The audio source service is responsible for taking in analog or digital audio from a legacy device (such as a CD changer) and transmitting this audio over the network. This service also provides controls (such as IR or RS-232) for control of the legacy device.

Audio/Local Source

The audio local source is responsible for taking in analog audio from a legacy device connected to an Audio Port and playing locally by the audio renderer service on the *SpeakerLinX*[™]. The source is not distributed over the network. IR commands may be received by the a *KeyLinX*[™] and passed through to the Audio Port to control the legacy device.

Audio Stream Proxy

An audio stream proxy service provides control and synchronization for native network audio sources, which do not conform to the *StreamNet* protocol (non-*StreamNet* devices). For example, the stream proxy service will convert the format of the content and control commands of a third party media server to be compatible with the *StreamNet* protocol.

User Interface (UI)

A UI service provides services for user inputs. There are currently two types of UI service: UI/GUI and UI/KEYPAD. A *TouchLinX* hosts a UI/GUI service and a *SpeakerLinX* hosts a UI/KEYPAD service.

Intercom

An intercom service provides one or two way voice communication with other intercom services.

General Purpose IO (GPIO)

The general purpose IO service (GPIO) is responsible for sending IR or RS232 commands to devices connected to a *ControlLinX*.

Root.

Every device has one root service. The root services manage common and miscellaneous housekeeping functions for each device. The root service is **always active**, even when all of the other services on the device have been disabled.

List of Services by Device

***SpeakerLinX* Services**

- Audio Renderer,
- Audio Source,
- Audio/Local Source,
- Audio Stream Proxy,
- User Interface (UI),
- Intercom, and
- Root.

***MediaLinX*[™] Services**

- Audio Source,
- Audio/Local Source,
- Audio Stream Proxy, and
- Root.

***TouchLinX* Services**

- User Interface (UI),
- Intercom, and
- Root.

***ControLinX* Services**

- General Purpose IO (GPIO), and
- Root.

Configuration

Each physical *DigiLinX* device may be configured to have some or all services enabled on it.

For example:

- a *SpeakerLinX* in room X may have:
 - an Audio Renderer service,
 - an Intercom service,
 - an Audio Source, service,
 - Audio/Local Source service,
 - an Audio Stream Proxy service, and
 - a Root service.
- a *MediaLinX* in room Y may have:
 - an Audio Source, service,
 - Audio/Local Source service,
 - an Audio Stream Proxy service, and
 - a Root service.
- A *SpeakerLinX* in Room Z may have:
 - an Audio Renderer service, and a
 - a Root service.

NOTE: The Root service is always active on a device.

ASCII Overview

The basic mechanism for control and status reporting of the services in *StreamNet*[™] is the ASCII protocol. This protocol consists of various messages and their parameters represented by strings of ASCII characters. Messages also have associated routing and administrative information which informs recipients of the destination and origin of the messages or provides various supplemental information.

StreamNet Addressing

StreamNet ASCII messages may be sent, using UDP, to any *StreamNet* device. If addressed correctly, the message will be routed to the desired service(s). Also, the messages may be sent to the *StreamNet* System multicast address. In this case, all *StreamNet* devices receive the message simultaneously, but only the addressed services accept the message and react to it. Messages may be addressed to a service, a group, or a room.

It is not necessary to know the IP addresses of the individual *StreamNet* devices to communicate with them. All messages are addressed to service names associated with the devices during the configuration process.

If the third party software has the ability to send multicast UDP messages, then all ASCII messages may be sent to the default system multicast address and the destination device or group of destination devices will receive it. Responses and status may be received by unicast/multicast UDP. If the third party software does not have multicast capability, the messages may be sent unicast UDP to any *StreamNet* device and it will be forwarded.

NOTE: Using the system multicast address is the preferred method of communicating with the *DigiLinX* system.

If the third party controller is also listening to its own (i.e., different) multicast address for responses, the controller and the *DigiLinX* system do not have to be in the same subnet.

The *DigiLinX* system multicast address **should not** be used for responses. Third party controllers should subscribe to use a separate unicast address or an unused multicast address to receive responses (see *Subscribing* on page 3-2).

Unsolicited status messages may be received by the third party controller.

Subscribing

During configuration, the integrator must subscribe to a *StreamNet* service's status messages by entering the controller's IP address and port to where the status messages are to be sent to get status from each service.

If a subscription exists, the *StreamNet* service sends status messages over UDP protocol to the specified IP address and port number.

A static subscription is configured using the *NetStreams, DigiLinX Dealer Setup Program* and is automatically in effect as soon as the device powers up and remains in effect indefinitely. The user should use the *NetStreams DigiLinX Dealer Setup Program* to specify an IP address and port number to which status reports are to be sent. This IP address is typically the static IP address of the third party controller device. If the device receiving the status messages is capable of receiving multicast UDP messages, then an unused multicast address may be specified for the subscription. Using a multicast address eliminates the need for the controller to have a static IP address and also eliminates any subnet related issues.

ASCII Message Syntax

ASCII messages have the following syntax:

#@toAddress:fromAddress%modifier%modifier#keyword arg1, arg2, ... argN /0

- A # symbol is required at the beginning of all messages.
- ASCII messages must not exceed 1000 characters in length.
- The various fields must be in the order shown above.
- The fields after the first # are the administrative fields, addresses and other control functions. The fields after the second # are the actual command.

@toAddress

Specifies the name of the service, group, or room to which the message is sent. The @ symbol precedes all toAddress fields. The address field may consist of two parts: the node address and the subnode address. The node address specifies the service, group, or room the message is addressed to. The subnode address can be used to address a subordinate module within the destination service or to invoke other special routing functions. The subnode address follows the node address and is separated by a tilde (~).

Example:

@nodeAddress~subnodeAddress

NOTE: When no service address is provided, the address defaults to the root service of that device.

The device reacts only if the command is a valid root service command.

Messages without a to address will be ignored if they are sent multicast or broadcast.

Node

A node address is part of the ASCII command address. It can be part of the TO or FROM address. A node address is either a service name, a service unique ID, a room name, or a group name.

UniqueID (UID)

The UID of a service is an unchangeable identifier for the service. It is based upon the device's serial number. The UID will be used as the service name if none has been specified during configuration. It will also be used as the service's room name if none has been specified. Normally, there is no reason for an integrator to use the UID. All services should be assigned names and all of the user's commands should be addressed to those service names (or room/group names).

The structure of the UID is serialNumber_index where **serialNumber** is the serial number of the device and **index** is an integer that identifies the specific service on the device.

Example:

#@MLA10105151001161008019_4



Serial Number



Index

Subnode

A subnode address is part of the ASCII command address. It can be part of the TO or FROM address. A subnode address is a special address modifier which instructs the node where to forward the command once it is received. Typically, the subnode address indicates that the service should forward the command to a non-*StreamNet* device.

Examples:

- ~IRMOD: forwards the command to the device which is providing the IR I/O functions to the *StreamNet* service.

- ~UDP192.168.5.3_5000: forwards the command using UDP to the specified IP address and port 5000. The subnode address may refer to a *StreamNet* device with an address unknown to the sender.

***StreamNet* Subnode Addresses:**

- ~CURSRC Route to renderers current audio source
- ~IRMOD Route to IR module on *MediaLinX*
- ~SERIAL_X Route to serial port number X
- ~KEYPAD Route to *KeyLinX* module
- ~ROOT Route to root service on device
- ~STATUS Not used to route to, but rather to indicate a status message
- ~SUBSCRIBER Route to subscribers

:fromAddress

Indicates which entity sent the message. This is used to address any response that may be generated by the message. If a response is expected, then the *fromAddress* should specify the address the response should be sent to. The **:** symbol must precede all *fromAddress* fields. The *fromAddress* may have a subnode address.

For third party controllers using UDP, it is recommended that the *fromAddress* resemble the following format: ~UDP192.168.5.3_5000. This includes the IP address and port to which any response should be sent.

%modifier

Optional field. May be used for various context specific purposes. If the recipient does not support the particular modifier, the field may be ignored. There **may** be any number of *%modifier* fields in a message. The **%** symbol precedes all modifier fields.

NOTE: This field is reserved for future use in *StreamNet*

#keyword (ASCII Command)

Required field. This is the keyword that specifies the action to be executed. The **#** symbol precedes all keyword fields (no intervening spaces). The keyword is one word and contains no spaces, commas, or other special characters except the underscore (**_**).

arg1, arg2, ..., argN

Optional field (some keywords may require arguments). The keyword may be followed by arguments. There are one or more spaces between the keyword and the first argument and a comma between each additional argument. Arguments are separated by commas and any extra white space should be ignored.

Arguments have the following formats:

- Meaningful strings – these are labels which represent some value, case, or function. Examples of meaningful strings are ON and OFF. Meaningful strings are not enclosed in quotation marks and must begin with an alphabetic character – not a numeral.
- Character literals – character strings do not need any special delimiters if they contain no blanks, spaces, commas, or other special characters. (A special character is other than alpha/numeric)

NOTE: If an argument contains any special characters or spaces, it must be enclosed between double brackets {{.....}}. When in doubt, use the double brackets.

- Numeric scalars – must be a decimal number.
- XML or other special formats. These arguments are enclosed in double brackets {{...}} and can take any form depending on context.

/0 (null byte)

All ASCII messages must end with a 0x00 byte (null).

Summary

The third party controller sends an ASCII message UDP to the IP address of any *DigiLinX* device.

In the ASCII message, you should have the:

- #@toAddress - name of the service you are addressing.
- :fromAddress - your own address with a port number to receive responses.
- The :fromAddress - your IP address/port and all responses will be sent to this address.
- #ASCII command with appropriate arguments required by that command.

ASCII Message Types

There are three types of ASCII messages - imperative, interrogative, and declarative.

Imperatives

Some messages are commands which initiate an action or directly cause a change of state. For example, commands that change volume level, select an audio source, or send an IR code.

Example:

```
#@CD PLAYER:~UDP192.168.5.3_5000#PLAY
```

The message will be routed to the *StreamNet* Service named CD PLAYER. This service will transmit an IR code for PLAY to the actual CD player being controlled.

NOTE: This command could be received by any *StreamNet* device in the system and is routed to the correct device.

If this had been a message which required a response, the response would be sent UDP to the IP address 192.168.5.3 port 5000.

Interrogatives

Some messages are used to request information. If an interrogative message is sent, a response is expected.

NOTE: An imperative may cause a change of state which then generates a report of the change. This is different from the question/answer of the interrogative.

Interrogatives are generally not required in normal integration applications.

Declaratives

Some messages only report information. They may be unsolicited status reports (periodic or caused by a change of state) or they may be a response to an interrogative message. The most common *StreamNet* declarative is the #REPORT message.

Menus

StreamNet has a special set of messages which make use of all three types, interrogative, declarative, and imperative. These messages are the menu messages: #MENU_LIST, #MENU_RESP, and #MENU_SEL. These three messages combine to provide a general means to present menus and allow menu selections without knowing the content of the menu in advance.

The controller sends a #MENU_LIST message to a *StreamNet* service to request the menu. The *StreamNet* service responds with a series of #MENU_RESP messages (one for each item in the menu). After a selection has been made, the controller sends a #MENU_SEL message indicating the user's selection.

Menus are organized in a branching tree structure. In a #MENU_LIST, there will be a certain number of items at the top level of the menu. Selecting one of these items may take the user to another menu. In turn, selecting one of these items from the second menu could return yet another menu. Selecting a terminal item (one without a subordinate menu) will almost always cause some action to take place. Selecting a non-terminal item may or may not cause an action.

SpeakerLinX Commands

A *SpeakerLinX* model SL220 or SL250, will respond only to commands as outlined in Table 4-1.

Commands must be issued to the appropriate serviceType specifying the serviceName in the :toAddress field.

Table 4-1
List of *SpeakerLinX* Commands

ASCII STRING	COMMENTS	SERVICE TYPE	MODEL(S)
AUDIO RENDERER TYPE COMMANDS			
#ACTIVE OFF	Enter the INACTIVE state	Renderer	SL220, SL250
#ACTIVE ON	Enter the ACTIVE state	Renderer	SL220, SL250
#AMP OFF	Turn OFF the amplifier	Renderer	SL220, SL250
#AMP ON	Turn ON the amplifier	Renderer	SL220, SL250
#LEVEL_SET BALANCE, x	Set the balance level.	Renderer	SL220, SL250
#LEVEL_DN BALANCE	Shift balance to the Left	Renderer	SL220, SL250
#LEVEL_UP BALANCE	Shift balance to the Right	Renderer	SL220, SL250
#LEVEL_SET BASS	Set the bass level.	Renderer	SL220
#LEVEL_DN BASS	Turn down bass level	Renderer	SL220
#LEVEL_UP BASS	Turn up bass level	Renderer	SL220
#LEVEL_SET TREB	Set the treble level.	Renderer	SL220
#LEVEL_DN TREB	Turn down treble level	Renderer	SL220
#LEVEL_UP TREB	Turn up treble level	Renderer	SL220
#LEVEL_SET VOL	Set the volume level.	Renderer	SL220, SL250
#LEVEL_DN VOL	Turn down volume level	Renderer	SL220, SL250
#LEVEL_UP VOL	Turn up volume level	Renderer	SL220, SL250
#LEVEL_SET BAND_X	Set band level to a specific value.	Renderer	SL250
#LEVEL_DN BAND_X	Decrease band level to a specific value.	Renderer	SL250
#LEVEL_UP BAND_X	Increase band level to a specific value.	Renderer	SL250
#MUTE OFF	Set MUTE=OFF	Renderer	SL220, SL250
#MUTE ON	Set MUTE=ON	Renderer	SL220, SL250
#MUTE TOGGLE	Toggle the Mute setting.	Renderer	SL220, SL250
#MENU_LIST m,n,SOURCES	List of sources available to the renderer service.	Renderer	SL220, SL250

Table 4-1
List of *SpeakerLinX* Commands

#MENU_SEL {{path}}	Select a specific source to be rendered.	Renderer	SL220, SL250
#SRC_SEL	Command renderer service to stop listening to any source.	Renderer	SL220, SL250
#SRC_SEL {{source id}}	Select a specific source to be rendered.	Renderer	SL220, SL250
#SRC_SEL NEXT	Choose next available source to be rendered	Renderer	SL220, SL250
#QUERY RENDERER	Query settings for current source being rendered.	Renderer	SL220, SL250
#QUERY CURRENT_SOURCE	Query which source is being rendered	Renderer	SL220, SL250

The following pages detail the functions of each *StreamNet* ASCII command.

The format of the text is as follows:

- A brief description of what the command does.
- Command name and format.
- Example ASCII message syntax for the above command.
- Notes specific to the command and explanation of responses generated.

***SpeakerLinX* Audio Renderer Type Commands**

Command: #ACTIVE OFF

Set the audio renderer service to OFF (InActive).

Example: #@Room 1 Player:~UDP10.15.0.11_5000#ACTIVE OFF

Command a renderer service to enter the OFF state.

The audio renderer service; Room 1 Player, is set to the OFF state.

NOTE: In this state, the renderer service will not listen to the audio source. It will also stop all audio output. In effect, if there was an audio source being rendered, the rendering will stop (that is, no music will be heard).

Command: #ACTIVE ON

Set the audio renderer service to on (Active).

Example: #@Room 1 Player:~UDP10.15.0.11_5000#ACTIVE ON

The audio renderer service; Room 1 player, is set to the ON state.

NOTE: When the renderer service enters the on state, it will maintain the last settings for the audio source, volume, treble, bass and balance.

Command: #AMP OFF

Turn amplifier OFF.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#AMP OFF

The amplifier associated with the audio renderer service; Room 1 player, is set to the OFF state.

NOTE: Although the amplifier is turned off, if the renderer service is active, it will still be listening to the audio source. Commands can still be issued to the renderer service to change the audio sources and settings.

Command: #AMP ON

Command a renderer service to turn the amplifier on.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#AMP ON

The amplifier associated with the audio renderer service; Room 1 player, is set to the ON state.

NOTE: #AMP OFF and #AMP ON have no effect on the renderer settings.

Levels

To control, report, and display analog (i.e., continuous) quantities such as volume, balance, brightness, and so on, the level construct was established. A level is a variable which may have any value between 0% and 100%, inclusive. This percentage is mapped to the actual analog value based on context and this mapping need not be linear.

Example:

A volume of 0% might be an attenuation of infinity, 50% is -25dB, 75% is -15dB, and 100% is 0dB.

For audio balance 0% is all left channel, 50% is equal for both channels, and 100% is all right channel.

Command: #LEVEL_SET BALANCE, x

Command a renderer service to set the balance level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_SET BALANCE, x

The audio renderer service; Room 1 player, sets the balance level based on the value x.

- x is a number between 0 – 100.
- x = 50 will set the balance to equal for both channels.

- $x < 50$ will shift the balance toward the left channel (by decreasing the right channel.)
- $x > 50$ will shift the balance toward the right channel (by decreasing the left channel).

Command: #LEVEL_DN BALANCE

Command a renderer service to shift the balance level to the left channel.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_DN BALANCE

The audio renderer service; Room 1 player, shifts the balance to the left channel (by decreasing the right channel).

Command: #LEVEL_UP BALANCE

Command a renderer service to shift the balance level to the right channel.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_UP BALANCE

The audio renderer service; Room 1 player, shifts the balance to the right channel (by decreasing the left channel.)

Command: #LEVEL_SET BASS, x

Command a renderer service to set the bass level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_SET BASS, x

The audio renderer service; Room 1 player, sets the bass level based on the value x.

- x is a number between 0 – 100.
- $x = 50$ will set the bass to nominal.
- $x < 50$ will decrease the bass from nominal.
- $x > 50$ will increase the bass from nominal.

Command: #LEVEL_DN BASS

Command a renderer service to decrease the bass level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_DN BASS

The audio renderer service; Room 1 player, decreases the bass level.

Command: #LEVEL_UP BASS

Command a renderer service to increase the bass level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_UP BASS

The audio renderer service; Room 1 player, increases the bass level.

Command: #LEVEL_SET TREB, x

Command a renderer service to set the treble level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_SET TREB, x

The audio renderer service; Room 1 player, sets the treble level based on the value x.

- x is a number between 0 – 100.
- x = 50 will set the treble to nominal.
- x < 50 will decrease the treble level from nominal.
- x > 50 will increase the treble level from nominal.

Command: #LEVEL_DN TREB

Command a renderer service to decrease the treble level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_DN TREB

The audio renderer service; Room 1 player, decreases the treble level.

Command: #LEVEL_UP TREB

Command a renderer service to increase the treble level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_UP TREB

The audio renderer service; Room 1 player, increases the treble level.

Command: #LEVEL_SET VOL, x

Command a renderer service to set the volume level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_SET VOL, x

The audio renderer service; Room 1 player, sets the volume level based on the value x.

- x is a number between 0 – 100.
- x = 0 will set the volume to the minimum level.
- x =100 will set the volume to the maximum level.

Command: #LEVEL_DN VOL

Command a renderer service to decrease the volume level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_DN VOL

The audio renderer service; Room 1 player, decreases the volume level.

Command: #LEVEL_UP VOL

Command a renderer service to increase the volume level.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_UP VOL

The audio renderer service; Room 1 player, increases the volume level.

Command: #LEVEL_SET BAND_X, Y

Set band level to a specific value.

- x is the band indicator (1 through 5).
- y is a number between 0 – 100.

- ❑ $y = 50$ will set the BAND_1 to nominal.
- ❑ $y < 50$ will decrease the BAND_1 level from nominal.
- ❑ $y > 50$ will increase the BAND_1 level from nominal.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_SET BAND_1, y
Command a Renderer Service to set the BAND_1 level.
The audio renderer service; Room 1 Player, sets the BAND_1 level based on the value of y.

Command: #LEVEL_DN BAND_X

Decrease band level.

- ❑ x is the band indicator (1 through 5).

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_DN BAND_1
Command a Renderer Service to decrease the band level.
The audio renderer service; Room 1 Player, decreases the band level.

Command: #LEVEL_UP BAND_X

Increase band level.

- ❑ x is the band indicator (1 through 5).

Example: #@Room 1 Player:~UDP10.15.0.11_5000#LEVEL_UP BAND_1
Command a Renderer Service to increase the BAND_1 level.
The audio renderer service; Room 1 Player, increases the BAND_1 level.

Command: #MUTE OFF

Command a renderer service to set mute to the off state.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#MUTE OFF

Command: #MUTE ON

Command a renderer service to set mute to the on state.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#MUTE ON

Command: #MUTE TOGGLE

Command a renderer service to toggle the state of mute.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#MUTE TOGGLE

Command: #MENU_LIST m,n,SOURCES

Get a list of audio sources available to the renderer service.

Example: #@Room 2 Player:~UDP10.15.0.11_5000#MENU_LIST
1,6,SOURCES

NOTE: m,n specifies the range of items to be listed from the list of sources. m = item number of the first item on the menu list to be listed, through n = the last item in the range.

Example: m=1 and n=6, will show the first through the sixth item, or m=3 and n=5 will show the third through the fifth items in the menu list.

NOTE: The command is requesting a list of six audio sources associated with this Service. The response can come in a series of messages depending on the numbers of audio sources associated with Room 2 player.

Responses Received

First Response

```
#@TL3800538001161007532_1~UDP10.15.0.11_2141:Room 2
Player#MENU_RESP {{<source id="myXM" children="0" itemnum="1"
idpath="sources" disppath="sources" display="myXM" ip="10.15.96.149"
type="audio/source" />}}
```

- ❑ Source id= "myXM"
- ❑ "myXM"=Service name of the audio sources. Any commands issued to this audio sources must be addressed to its Service name myXM.
- ❑ Children="0" – indicates a terminal node (no more lists under this audio source).
- ❑ Itemnum="1" – indicates the item number on this menu. 1 indicates this is the first item on the menu. If this is a positive number, then there are more audio sources in the list. A negative number indicates the end of the list.
- ❑ IDPath= "sources" – If there are further list(s) under this MENU, then the IDPath concatenated with the Source ID defines the path for the next item on the menu.
- ❑ Dispath= "sources" – This string can be used to display the menu path, if desired.
- ❑ Display= "myXM" – This string can be used to display the first menu item. (based on itemnum).
- ❑ Type = "audio/source" – only audio/source serviceType commands may be issued to this service.

The remaining responses show four more audio sources and can be parsed similarly.

Second Response

```
#@TL3800538001161007532_1~UDP10.15.0.11_2141:Room 2
Player#MENU_RESP {{<source id="Stream 3" children="0" itemnum="2"
idpath="sources" disppath="sources" display="Stream 3" ip="10.15.96.149"
type="audio/source" />}}
```

Third Response

```
#@TL3800538001161007532_1~UDP10.15.0.11_2141:Room 2
Player#MENU_RESP {{<source id="Local Source 2" children="0" itemnum="3"
idpath="sources" disppath="sources" display="Local Source 2" ip="10.15.32.147"
type="audio/localsource" />}}
```

Fourth Response

```
#@TL3800538001161007532_1~UDP10.15.0.11_2141:Room 2
Player#MENU_RESP {{<source id="Stream 1" children="0" itemnum="4"
idpath="sources" disppath="sources" display="Stream 1" ip="10.15.32.147"
type="audio/source" />}}
```

Fifth Response

```
#@TL3800538001161007532_1~UDP10.15.0.11_2141:Room 2
Player#MENU_RESP {{<source id="Stream 2" children="0" itemnum="5"
idpath="sources" disppath="sources" display="Stream 2" ip="10.15.1.114"
type="audio/source" />}}
```

Sixth Response

```
#@TL3800538001161007532_1~UDP10.15.0.11_2141:Room 2
Player#MENU_RESP {{<sources idpath="sources" itemnum="-1" />}}
```

NOTE: itemnum="-1". The "--" sign indicates that there are no more audio sources.

If itemnum = a positive number, then there are more audio sources assigned to this Service and the command needs to be re-issued until itemnum="-x" is received.

Example: #@Room 2 Player:~UDP10.15.0.11_5000#MENU_LIST 7,9,SOURCES – will get a response indicating the seventh through ninth item from the MENU_LIST.

NOTE: If there are only five items in the menu (as in this case), then a command requesting a list of ten sources, like:

```
#@Room 2 Player:~UDP10.15.0.11_5000#MENU_LIST 1,10,SOURCES
```

will still generate the exact same response as above, with the sixth response showing "itemnum"= "-1".

Command: #MENU_SEL {{path}}

Select an audio sources from a list available to the renderer service.

Example: #@Room 2 Player:~UDP10.15.0.11_5000#MENU_SEL
 {{sources>Stream 2}}

This command selects Stream 2 as the source to be rendered.

NOTE: The menu path for this list is gotten by concatenating idpath and source id separated by a >.

Example: {{sources>Stream 2}} becomes the menu path to select the fifth (itemnum=5) audio source based on the responses received from the MENU_LIST example above.

NOTE: Alternatively, in this specific case, an #SRC_SEL command (see below) may be used.

Command: #SRC_SEL {{source id}}

Select a specific audio source to be played.

Example: #@Room 2 Player:~UDP10.15.0.11_5000#SRC_SEL {{Stream 3}}
 selects Stream 3 to be rendered by service; Room 2 player.

NOTE: The source name is case sensitive and must be bound by the double curly brackets, as in {{Stream 3}}. The double curly brackets are required since the Service name has a space (non alpha-numeric character)

Command: #SRC_SEL

De-selects all audio sources from renderer.

Example: #@Room 2 Player:~UDP10.15.0.11_5000#SRC_SEL " "

NOTE: If no audio source is specified between the " ", the current audio sources is de-selected resulting in no audio sources being rendered by the service; Room 2 player.

Command: #SRC_SEL NEXT

Selects the next audio source to be played.

Example: #@Room 2 Player:~UDP10.15.0.11_5000#SRC_SEL next
 Selects the next available audio sources to be rendered by the service; Room 2 player.

Command: #QUERY RENDERER

Query settings for current audio source being rendered.

Example: #@Room 1 Player:~UDP10.15.0.11_5000#query renderer

NOTE: A #QUERY command is interrogative and gets a declarative response #REPORT.

The #QUERY RENDERER gets a report indicating the current settings for the audio source being rendered.

Sample response:

```
#@SL22004280001161000057_4~UDP10.15.0.11_4881:Room 1
Player#REPORT { {<report type="state" vol="29" balance="50" bass="50"
treb="50" loud="0" mute="0" audioSession="All Rooms Multiroom"
audioSessionActive="1" ampOn="1" />}}
```

Parsing Reports

NOTE: Even though the format looks like XML, it isn't. The report will always be flat (i.e., no nested <> brackets).

The order of the attributes may change. Some may be added or omitted in any given #REPORT message. Parsing of the message should key off the attribute name. For example, for balance a suggested parsing method would be to scan for "=". The attribute name is to the left of "=" and the value is to the right. The serviceName attribute (i.e., Room 1 player) should be used to qualify the report before accepting the other data attributes (i.e., Room 1 player). Make sure who the message is from before storing or using the reported status.

NOTE: The data may be distributed among multiple #REPORT messages even if the report(s) is the result of a single #QUERY message.

Command: #QUERY CURRENT_SOURCE

Query current audio sources being rendered.

Example: #@Room 2 Player:~UDP10.15.0.11_5000#query current_source

Response Received:

```
#@TL3800538001161007532_1~UDP10.15.0.11_2230:Room 2
Player#REPORT { {<report type="state" currentSource="myXM"
currentSourceIP="10.15.96.149" permId="MLA10105151001161008019_1" /
>}}
```

NOTE: Room 2 player is now rendering an audio sources called "myXM".

SpeakerLinX Media Server Audio Stream Type Commands

ASCII STRING	COMMENTS	SERVICE TYPE	MODEL
#MENU_LIST m,n,{{path}}	List of items in the song menu.	Stream	All
#MENU_SEL {{path}}	Select a specific item from the MENU_LIST	Stream	All
#NEXT	Select the next song	Stream	All
#PAUSE	Pause	Stream	All
#PLAY	Play	Stream	All
#PREV	Select the previous song	Stream	All
#SHUFFLE	Toggle Shuffle mode	Stream	All
#STOP	Stop	Stream	All
#QUERY SOURCE	Show a report on the current song being played	Stream	All

Command: #MENU_LIST m,n,{{path}}

Request a list of media associated with a media server.

Example: #@Stream 1:~UDP10.15.0.11_5000#menu_list 1,10,media

NOTE: This message is addressed to Stream 1. This is a Stream Service type message. The command is requesting menu items 1 to 10 of the media associated with this Service. Media is the default path name for the first menu list.

Responses Received

First Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP
{{<item idpath="media" disppath="media" itemnum="1" id="All Songs"
display="All Songs" children="1078" />}}
```

Example: {{media>All Songs}} becomes the next menu path.

- ❑ Itemnum = "1", indicates this is the first item on this menu list.
- ❑ "All Songs" is the display name for this list.
- ❑ children="1078", indicates that this is not a terminal node (i.e., there is another menu list below this level).

Second Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1#MENU_RESP
{{<item idpath="media" disppath="media" itemnum="2" id="Artists"
display="Artists" children="63" />}}
```

The next level menu path for this list is gotten by concatenating idpath and ID separated by a >.

Example: `{{media>Artists}}` becomes the next menu path.

- ❑ `Itemnum = 2`, indicates this is the second item on this menu list.
- ❑ `"Artists"` is the display name for this list.
- ❑ `children="63"`, indicates that this is not a terminal node. ie: there is another menu list below this level.

Responses 3 through 5

These responses can be similarly parsed.

- ❑ `#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP {{<item idpath="media" disppath="media" itemnum="3" id="Albums" display="Albums" children="116" />}}`
- ❑ `#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP {{<item idpath="media" disppath="media" itemnum="4" id="Genres" display="Genres" children="12" />}}`
- ❑ `#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP {{<item idpath="media" disppath="media" itemnum="5" id="Playlists" display="Playlists" children="3" />}}`

Sixth Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP
{{<item idpath="media" disppath="media" itemnum="-1" />}}
```

`Itemnum = "-1"`. The negative value indicates there are no more items on this list.

NOTE: As long as `children = positive number`, there are more items under that list. However, `children = "0"` is not a reliable indicator of the end of the list. `itemnum = negative` is more reliable to decide if the end of the list has been reached.

Summary

The service Stream 1 menu has five items which are displayed as:

- All Songs
- Artists
- Albums
- Genre
- Playlists

Each of these branches has children (another menu list) under them.

Command: #MENU_LIST m,n,{{path}}

This command allows you to traverse down the menu list.

As explained above, the next level menu path is gotten by concatenating item idpath and id separated by a >.

Example: `{{media>All Songs}}` becomes the next menu path:

```
#@Stream 1:~UDP10.15.0.11_5000#menu_list 1,5,{{media>All Songs}}
```

The command is requesting menu items 1 to 5 of the menu list below the list All Songs, which was the first item on the previous menu list.

Responses Received

First Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP
{{<song idpath="media&gt;All Songs" disppath="media&gt;All Songs"
itemnum="1" id="1481" display="Brown, Crane, Ker, Finesilver" children="0" /
>}}
```

The next level menu path for this list is gotten by concatenating item idpath and id separated by a >. Example: `{{media>All Songs>1481}}` becomes the next menu path for Brown, Crane, Ker, Finesilver

- Itemnum = "1", indicates this is the first item on this menu list.
- The display name = "Brown, Crane, Ker, Finesilver" for this item.
- children="0", indicates that this is a terminal node (i.e., there are no items below this level). Therefore, a further #MENU_LIST command would neither get a response nor a reaction. Instead, as explained below, a #MENU_SEL command must be issued.

Responses 2 through 5

These responses can be similarly parsed.

- #@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP {{<song idpath="media>All Songs" disppath="media>All Songs" itemnum="2" id="1648" display="(Da Le) Yaleo" children="0" />}}
- #@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP {{<song idpath="media>All Songs" disppath="media>All Songs" itemnum="3" id="1606" display="(Hidden Track)" children="0" />}}
- #@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP {{<song idpath="media>All Songs" disppath="media>All Songs" itemnum="4" id="1974" display="11 O'Clock Tick Tock" children="0" />}}
- #@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP {{<song idpath="media>All Songs" disppath="media>All Songs" itemnum="5" id="1605" display="2000 Miles" children="0" />}}
- Itemnum = "5". The positive value indicates there are more items on this list.

- ❑ Children = "0", in each of these items indicates that there are no more items under these items.

Summary

"All Songs" (the first item on the menu) has a list of items 1 to 5 which are displayed as:

- ❑ Brown, Crane, Ker, Finesilver
- ❑ (Da Le) Yaleo
- ❑ (Hidden Track)
- ❑ 11 OClock Tick Tock
- ❑ 2000 Miles

The "All Songs" menu had shown it had 1078 children of which the command `#menu_list 1,5,{{media>All Songs}}` requested a list of items 1-5.

Similarly, if the integrator wanted to see the list under the second item, "Artists", the command syntax would be:

```
#@Stream 1:~UDP10.15.0.11_5000#menu_list 1,5,{{media>Artists}}
```

Command: #MENU_SEL {{path}}

Use this command to select a specific item from the menu list.

When a MENU_LIST item has children="0", then a #MENU_SEL command must be issued to select that item.

Example: `#@Stream 1:~UDP10.15.0.11_5000#menu_sel {{ media>All Songs>1481 }}`

Selects the song, Borwn, Crane, Ker, Finesilver (id=1481) from the All Songs menu to be played.

Command: #NEXT

This command selects the next song to be played.

Example: `#@Stream 3:~UDP10.15.0.11_5000#next`

The next song from the current playlist from the audio stream, Stream 3 will be selected and played.

Command: #PAUSE

This command sends a pause command to the specified audio source.

Example: `#@Stream 3:~UDP10.15.0.11_5000#pause`

The audio stream, Stream 3, will be paused.

Command: #PLAY

This command sends a play command to the specified audio source.

Example: # @Stream 3:~UDP10.15.0.11_5000#play

The audio stream, Stream 3, will be played.

Command: #PREV

This command selects the previous song to be played.

Example: # @Stream 3:~UDP10.15.0.11_5000#prev

The previous song from the current play list from the audio stream, Stream 3 will be played.

Command: #SHUFFLE

This command sends a shuffle command to the specified audio source.

Example: # @Stream 3:~UDP10.15.0.11_5000#shuffle

The stream will be played in Shuffle mode.

NOTE: If the audio sources is already in Shuffle mode, then sending the command again will toggle the mode.

Command: #STOP

This command sends a STOP command to the specified audio sources.

Example: # @ Stream 3:~UDP10.15.0.11_5000#stop

The audio stream, Stream 3, will stop being played.

Command: #QUERY SOURCE

This command gets details on the current media being played.

Example# @Stream 1:~UDP10.15.0.11_5000#QUERY SOURCE

Response Received

```
# @SL22004285001161000172_4~UDP 10.15.0.11_4028:Stream 1#REPORT
{{<report type="source" artwork="http://10.15.100.116/7.0.5/Database/Music/
Covers/s4/s3/Nails-Corpus Christi" display="song/artist/album/genre" song="14
Dreams" album="Corpus Christi" artist="Nails" genre="Rock" time="376"
percent="59" sngPIIndex="79" sngPITotal="11169" source="Stream 1"
elapsed="223744" next="14 Years" active="2" controlState="PAUSE" shuffle="0"
/>}}
```

- ❑ This command gives a report of the current song in "Stream 1"
- ❑ The display attribute gives a suggestion of which items to display and with what priority.
- ❑ In this example the audio source Stream 1 is suggesting that the song title is the most important item to display; followed by the artist's name, followed by the name of the album and the genre.

- The UI device is not bound by the suggestion.

NOTE: #REPORT responses shall in no case exceed 1000 characters, but data may be distributed among multiple #REPORT messages.

SpeakerLinX Intercom serviceType Commands

Setup Program Guidelines

The following apply to the *DigiLinX* Dealer Setup program:

- Only one service per room may have monitorEnable enabled.
- Only one service per room may have defaultMic enabled.
- All intercom services MUST be a member of the NS_IC_ALL group.
- A range of IP addresses must be configured for Intercom (see the *DigiLinX* Dealer Setup for the default set).

ASCII STRING	COMMENTS	SERVICE TYPE	MODEL
#INTERCOM JOIN, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD] [,IMMED]	The intercom service joins an intercom session.	Intercom	All
#INTERCOM JOINMON, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD]	The intercom service joins an intercom monitoring session.	Intercom	All
#INTERCOM MONITOR, roomName, audience	roomName is room to be monitored and audience are the monitoring room(s) for that room	Intercom	All
#INTERCOM LEAVE [, tag]	Terminate the intercom session	Intercom	All
#INTERCOM LEAVE [, monitor]	Terminate intercom Monitor session	Intercom	All
#INTERCOM LEVEL_UP VOL	Increase intercom service volume level	Intercom	All
#INTERCOM LEVEL_DN VOL	Reduce intercom service volume level	Intercom	All
#INTERCOM LEVEL_SET VOL, n	Set intercom service volume level	Intercom	All
#INTERCOM PTT, tag	Activate microphone for 300ms	Intercom	All
#INTERCOM PTT OFF	Disable microphone immediately	Intercom	All
#DND ON/OFF	Set Do Not Disturb state	Intercom	All
#PRIVACY ON/OFF	Set Privacy state	Intercom	All

Command: #INTERCOM JOIN, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD] [,IMMED]

Join an Intercom session.

- ID = the unique name of the intercom session. An attempt to start a second session with the same ID will fail because the Intercom services will ignore a join command with an ID that is already registered. The default intercom session has the ID of intercom.

NOTE: Members may be added to a session by sending out a join message with the same ID and tag but addressed to different services.

- tag = a 32 bit session number that will be used to identify the session in the header of the intercom audio data and to reference it in other commands.
 - listenIpAdr = the IP address to which the device should listen. It may be multicast or unicast.
 - listenPort = the IP port to which the device should listen.
 - TalkIpAdr = the IP address to which the device should address replies. It may be multicast or unicast.
 - talkPort = the IP port to which the device should address replies.
 - OVRRD = a string which if present overrides the do not disturb setting and initiates a session regardless. [OVRRD] does not override configuration flags.
 - IMMED = a string which if present will cause this new session to immediately become the active session.
-

NOTE: For most intercom sessions, it is expected that both the talk and listen addresses are the same and are multicast addresses.

When point to point (unicast) sessions are desired, these addresses would be the unicast addresses of the two participants; although, even for point to point the addresses may be multicast. TalkIpAdr would be the address of the originator and ListenIpAdr would be the second service.

A service could specify a multicast listen address and its unicast address for talk. This would create a situation where every destination would hear the originator but only the originator would hear the responses.

Example: `#@Room 1 Intercom:~UDP10.15.0.11_5000#intercom join, intercom1, 12345678, 239255481, 5005, 239255481, 5005, OVRRD, IMMED`

The Join message adds the specified session intercom1, to the intercom service's list of sessions.

Command: #INTERCOM JOINMON, ID, tag, listenIpAdr, listenPort, talkIpAdr, talkPort [,OVRRD]

Join and monitor an intercom session.

Example: `#@Room 1 Intercom:~UDP10.15.0.11_5000#intercom JOINMON, intercom1, 12345678, 239255481, 5005, 239255481, 5005, OVRRD`

The Joinmon command has exactly the same format and effect as the Join command except the monitoring flag is set for the intercom session.

NOTE: The OVRRD will override the Privacy setting instead of the DND setting.

The intercom service will execute an #INTERCOM JOINMON command even if it does not have monitor enabled.

Command: #INTERCOM MONITOR, roomName, audience

This command sets monitored and monitoring rooms.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom monitor, Room1, Room2

NOTE: roomName is the name of the room which is to be monitored.

audience is a service, room, or group which will monitor the specified room.

Command: #INTERCOM LEAVE, [tag]

Terminate an intercom session.

[tag] is optional

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom leave, 12345678

NOTE: The Intercom session service Room 1 Intercom will be terminated.

Command: #INTERCOM LEAVE, monitor

Terminate a monitor session.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom leave, monitor

The Intercom monitor session Room 1 Intercom will be terminated.

Command: #INTERCOM LEVEL_UP VOL

Increase intercom volume level.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom level_up vol

NOTE: This does not affect the volume settings for the renderer.

Command: #INTERCOM LEVEL_DN VOL

Decrease intercom volume level.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom level_dn vol

NOTE: This does not affect the volume settings for the renderer.

Command: #INTERCOM LEVEL_SET VOL, x

Set intercom volume level.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom level_set vol, 30

NOTE: This does not affect the volume settings for the renderer.

Command: #INTERCOM PTT, [tag]

Activate microphone while talking.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom ptt, 12345678

NOTE: The command must be repeated at least every 300ms to keep microphone on.

Command: #INTERCOM PTT OFF

Disable microphone immediately.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom ptt, off

Command: #DND x

Set Do Not Disturb state.

x = ON or OFF

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#DND on

NOTE: If ON is set, the intercom service will not join any intercom session.

Command: #PRIVACY x

Set Privacy state.

x = ON or OFF

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#Privacy on

NOTE: If ON is set, then the intercom service will not allow itself to be monitored.

Sample Intercom Scenarios

Simple Scenario

On the user interface, there is a single Talk button. Pressing this button enables the user to talk to the entire house.

When the Talk button is pressed, the controller sends out an #INTERCOM JOIN message addressed to the All intercom session. It would also send an #INTERCOM PTT message for the session to the local room and repeat the #INTERCOM PTT message every 300ms for as long as the button is held. When the button is released, the controller should send an #INTERCOM PTT, OFF message for the session to the local room.

Example:

If a user in the bedroom presses the Talk button:

- #@NS_IC_ALL:~UDP10.15.0.11_5000#intercom join, intercom1, 12345678, 239255481, 5005, 239255481, 5005, OVRRD, IMMED
- #@Bedroom#INTERCOM PTT (every 300ms, while PTT button is pressed)
- #@Bedroom#INTERCOM PTT, OFF (when the PTT button is released)

Complete Control Scenario

On the user interface are a block of call buttons. Each button specifies a section of the house or room or group of rooms (these groupings are called sessions). This button is also a push to talk (PTT) button for that session.

When a session button is pressed, the controller sends out a #INTERCOM JOIN message addressed to the session specified. It also sends a #INTERCOM PTT message for the session to the local room and repeats the #INTERCOM PTT message every 300ms for as long as the button is held. When the button is released, the controller should send a #INTERCOM PTT, OFF message for the session to the local room. When a response comes in the user interface could highlight the button corresponding to the session responding.

If a user in the bedroom selects a session button labeled Downstairs, the controller should send the following messages immediately:

- #@Bedroom#INTERCOM JOIN, Downstairs, 12345678, 239255481, 5005, 239255481, 5005, OVRRD, IMMED
- #@Downstairs# INTERCOM JOIN, Downstairs, 12345678, 239255481, 5005, 239255481, 5005, OVRRD, IMMED
- #@Bedroom#INTERCOM PTT, 12345678 (every 300ms)
- #@Bedroom#INTERCOM PTT, OFF (when the button is released)

SpeakerLinX Root Type Commands

ASCII STRING	COMMENTS	SERVICE TYPE	MODEL
#QUERY SERVICE	List configuration info for all services	Root	All
#QUERY SERVICE, {{service name}}	Show configuration info for specific service	Root	All
#MENU_LIST m,n,Intercom	List of Intercoms	Root	All
#MENU_LIST m,n,renderers	List of Renderers	Root	All
#RESTART	Restart the device	Root	All

Command: #QUERY SERVICE

List services configured with a device.

Example: #query service

NOTE: To get a list of services configured with a device, the third party controller needs to send a UDP message to the IP address of a specific device with the #query service command.

Query service is a root service command and must be addressed to a specific device.

The root service on every device is always enabled. Root service commands do not need an @toAddress.

Response Received

```
#@SL22004280001161000057_4~UDP10.15.0.11_5000:SL22004280001161000057_4#REPORT {{<report type="state" serviceName1="Local Source 2"
serviceName2="Stream 1" serviceName3="Room 2 Player"
serviceName4="SL22004280001161000057_4" serviceName5="Room 2 KL" /
>}}
```

The report shows five services:

- ❑ Local Source 2
- ❑ Stream 1
- ❑ Room 2 player
- ❑ SL22004280001161000057_4 (Root service)
- ❑ Room 2 *KeyLinX*

Each service can be further drilled down as shown below.

Command: #QUERY SERVICE {{serviceName}}

List configuration settings for a specific service.

Example: #query service {{Local Source 2}}

Response Received

```
#~UDP10.15.0.11_5000:SL22004280001161000057_4#REPORT {{<report  
type="state" serviceName="Local Source 2" serviceType="audio/localsource"  
IP="10.15.32.147" permId="SL22004280001161000057_1" enabled="1"  
sourceType="NOCTRL" roomName="Room 2" groupName="ALL" />}}
```

NOTE: Response shows the Local Source 2 is of service type = audio/localsource and therefore only audio/localsource type commands will be recognized by this service. Service names are case sensitive when used in the argument field.

NOTE: The serviceName must be bound by the double curly brackets, as in {{Local Source 2}}. The double curly brackets are required since the Service name has spaces (non alpha-numeric characters):

Example: #query service {{room 2 player}}

Service names are case sensitive when used in the argument field. The above command does not get recognized since the actual service name is Room 2 player, so no response will be seen.

Response Received

```
#~UDP10.15.0.11_5000:SL22004280001161000057_4#REPORT {{<report  
type="state" serviceName="Room 2 Player" serviceType="audio/renderer"  
IP="10.15.32.147" permId="SL22004280001161000057_3" enabled="1"  
roomName="Room 2" groupName="ALL" />}}
```

NOTE: Response shows the Room 2 player is of service type = audio/renderer and therefore only renderer commands will be recognized by this service.

Command: #@Room 2#Query Service

List Services configured for all devices in a room.

Example: #@Room 2:~UDP10.15.0.11_5000#query service

Responses Received

Response # 1

```
#~UDP10.15.0.11_5000:TL3800538001161007532_1#REPORT {{<report  
type="state" serviceName1="TL3800538001161007532_1"  
serviceName2="Room 2 TL UI" />}}
```

Response # 2

```
#~UDP10.15.0.11_5000:SL22004280001161000057_4#REPORT {{<report
type="state" serviceName1="Local Source 2" serviceName2="Stream 1"
serviceName3="Room 2 Player" serviceName4="SL22004280001161000057_4"
serviceName5="Room 2 KL" />}}
```

NOTE: #QUERY SERVICE to a room will get a response from the root service of each device in the room indicating the services configured on each device.

NOTE: Response # 1 is received from the *TouchLinX*. It has two services associated with it:

```
serviceName1="TL3800538001161007532_1" is of type "root "
serviceName2="Room 2 TL UI"
```

If a service name is not assigned to the "root" service the service name defaults to the serial number of the device.

Response # 2 is received from the *SpeakerLinX*. It lists five services:

- serviceName1="Local Source 2"
- serviceName2="Stream 1"
- serviceName3="Room 2 player"
- and so on.

Command: #MENU_LIST m,n,RENDERERS

Get a list of renderer services.

Example: #MENU_LIST 1,5,RENDERERS

Menu lists are interrogative and therefore get responses.

This command will get a response for renderer services 1 through 5.

NOTE: m,n specifies the range of items to be listed from the list of renderers. m = item number on the menu list of the first item to be listed, through n = the last item in the range.

Example: m=3 and n=5, would show the third, fourth and fifth item.

NOTE: The command is requesting a list of 1-5 renderers associated with this root service (starting with the first item on the menu). The response can come in a series of messages depending on the number of renderers in the system.

Responses Received

- ❑ #@MLA10105151001161008019_4~UDP10.15.0.11_2174:Room 1 Player#MENU_RESP {{<renderer id="Room 2 Player" children="0" itemnum="1" idpath="renderers" disppath="renderers" itemnum="1" display="Room 2 Player" ip="10.15.32.147" roomName="Room 2" permId="SL22004280001161000057_3" />}}
- ❑ #@MLA10105151001161008019_4~UDP10.15.0.11_2174:Room 1 Player#MENU_RESP {{<renderer id="Room 1 Player" children="0" itemnum="2" idpath="renderers" disppath="renderers" itemnum="2" display="Room 1 Player" ip="10.15.1.114" roomName="Room 1" permId="SL22004285001161000172_3" />}}
- ❑ #@MLA10105151001161008019_4~UDP10.15.0.11_2174:Room 1 Player#MENU_RESP {{<renderer id="Room 1 Player" children="0" itemnum="-2" idpath="renderers" disppath="renderers" itemnum="-2" display="Room 1 Player" ip="10.15.1.114" roomName="Room 1" permId="SL22004285001161000172_3" />}}

NOTE: This response shows there are two renderers in this project with service names Room 1 player and Room 2 player.

- ❑ Room 1 player is associated with roomName="Room 1"
 - ❑ Room 2 player is associated with roomName="Room 2"
-

- ❑ children="0", indicates there are no further lists under this service.
- ❑ In the third response, Itemnum="-2". If this is a positive number, then there are more renderers in the list. A negative number indicates the end of the list.
- ❑ Although the command requested to list five renderers, since there were only two renderer services, the third response contained Itemnum=" -2", indicating the end of the list. If five responses were received with the fifth Itemnum="5" (a positive number), then there are potentially more available renderer services. The command must be re-issued with a larger number (eg: #MENU_LIST 1,10,renderers) until a response is received with Itemnum = a negative number.

Case Sensitivities

When the source or renderer service name is used as the toAddress; it is not case sensitive

Example 1: #@room 2 player:~UDP10.15.0.11_5000#query renderer

Received from Server:

```
#~UDP10.15.0.11_5000:Room 2 Player#REPORT {{<report type="state"
vol="29" balance="50" bass="50" treb="50" loud="0" mute="0"
audioSession="" audioSessionActive="0" ampOn="1" />}}
```

Example 2: #@stream 3#menu_list 1,10,media

Response Received:

```
#~UDP10.15.0.11_5000:Stream 3#MENU_RESP {{<item idpath="media"
disppath="media" itemnum="1" id="97>1" display="All Songs"
children="11169" />}}
```

The service name is case sensitive when used as an argument

Example: #@Room 2:~UDP10.15.0.11_5000#query service {{Room 2 Player}}

Received from Server:

```
#~UDP10.15.0.11_5000:SL22004280001161000057_4#REPORT {{<report
type="state" serviceName="Room 2 Player" serviceType="audio/renderer"
IP="10.15.32.147" permId="SL22004280001161000057_3" enabled="1"
roomName="Room 2" groupName="ALL" />}}
```

Example: #@Room 2:~UDP10.15.0.11_5000#query service {{room 2 player}}

No Response Received.

MediaLinX Commands

The following commands can only be used on a *MediaLinX*. Commands must be issued to the appropriate serviceType specifying the serviceName in the toAddress. On receipt, the *MediaLinX* converts these commands to the appropriate IR equivalent and transmits the IR command to the legacy unit.

NOTE: The IR commands must be learned on the *MediaLinX* connected to the legacy device.

Table 5-1
Commands

ASCII String	Comments	Service Type	Model
CD PLAYER COMMANDS			
#PREV	Select the previous song	Source	All
#NEXT	Select the next song	Source	All
#PWR ON	Turns on power to the source.	Source	All
#PWR OFF	Turns off power to the source.	Source	All
#PLAY	Play	Source	All
#PAUSE	Pause	Source	All
#STOP	Stop	Source	All
#REV	Generic REV command	Source	All
#FWD	Generic FWD command	Source	All
#DISC PREV	Select the previous disc	Source	All
#DISC NEXT	Select the next disc	Source	All
#RANDOM	Select Random mode	Source	All
#REPEAT TOGGLE	Toggle Repeat mode	Source	All
#KEY x	Keys 0 through 9	Source	All
#F1, #F2	Unique commands to the device that are not in the standard command set.	Source	All
#TEN_PLUS	Allows you to enter a two-digit number or jump the playback 10 CDs forward depending on the brand of CD player you have.	Source	All

Table 5-1
Commands

ASCII String	Comments	Service Type	Model
#HUNDRED_PLUS	Allows you to enter a three-digit number or jump the playback 100 CDs forward depending on the brand of CD player you have.	Source	All
#CLEAR	Optional. Depends on how the dealer programs the button. Typically, it's programmed to clear entries made on the keypad.	Source	All
#ENTER	Optional. Depends on how the dealer programs the button. Typically, it's programmed to enter information keyed in on the keypad.	Source	All
TUNER COMMANDS			
#PREV	Select the previous song	Source	All
#NEXT	Select the next song	Source	All
#PWR ON	Turns on power to the source.	Source	All
#PWR OFF	Turns off power to the source.	Source	All
#TUNE UP	Manually dialing up the frequency.	Source	All
#TUNE DN	Manually dialing down the frequency.	Source	All
#SEEK UP	Goes to the next station and stops.	Source	All
#SEEK DN	Goes to the previous station and stops.	Source	All
#SCAN UP	Scan automatically searches up the frequency for the next station and then continues to the next station.	Source	All
#SCAN DN	Scan automatically searches down the frequency for the previous station and then continues to the previous station.	Source	All
#PRESET UP	Goes to the next preset and stops.	Source	All
#PRESET DN	Goes to the previous preset and stops.	Source	All
#PRESET x	Select Preset station - Tuner command	Source	All
#BAND NEXT	Changes from AM to FM.	Source	All
#KEY x	Keys 0 through 9, A, B	Source	All
#CLEAR	Optional. Depends on how the dealer programs the button. Typically, it's programmed to clear entries made on the keypad.	Source	All
#ENTER	Optional. Depends on how the dealer programs the button. Typically, it's programmed to enter information keyed in on the keypad.	Source	All
DVD COMMANDS			
#PWR ON	Turns on power to the source.	Source	All
#PWR OFF	Turns off power to the source.	Source	All
#PLAY	Play	Source	All
#PAUSE	Pause	Source	All
#STOP	Stop	Source	All
#PREV	Select the previous song	Source	All
#NEXT	Select the next song	Source	All

Table 5-1
Commands

ASCII String	Comments	Service Type	Model
#STOP	Stop	Source	All
#REV	Generic REV command.	Source	All
#FWD	FWD	Source	All
#TOPMENU	Top Menu - takes the disc navigation to the top menu stored on the DVD disc.	Source	All
#MENU	Menu - controls the menu functions and allows you to change DVD player settings.	Source	All
#SELECT	Select - selects the highlighted item.	Source	All
#SETUP	Brings up the onscreen setup menu for the DVD player.	Source	All
#DISPLAY	Brings up an onscreen menu of the video settings for the DVD player.	Source	All
#DVDAUDIO	Brings up an onscreen menu of the audio settings for the DVD player.	Source	All
#SUBTITLE	Brings up the subtitle on/off default settings for the DVD player.	Source	All
#LANG	Brings up the language default settings for the DVD player.	Source	All
#MODE	Changes the mode of play from repeat to shuffle (or other presets programmed by the installer).	Source	All
#EJECT	Opens or closes the DVD tray on the DVD player.	Source	All
#ANGLE	If your DVD has different angles of the same scene, you can press this button to view the scene from a different angle.	Source	All
#ZOOM	If your DVD disc allows you to zoom in on a scene, pressing this button will activate the Zoom feature.	Source	All
#RETURN	Returns to the previous menu (or performs as programmed by the installer).	Source	All
#NAV UP	Navigates up.	Source	All
#NAV DN	Navigates down.	Source	All
#NAV LT	Navigates left.	Source	All
#NAV RT	Navigates right.	Source	All
#KEY x	Keys 0 through 9, A, B	Source	All
#F1, #F2	Unique commands to the device that are not in the standard command set.	Source	All
#TEN_PLUS	Allows you to enter a two-digit number or jump the playback 10 DVDs forward depending on the brand of DVD player you have.	Source	All
#HUNDRED_PLUS	Allows you to enter a three-digit number or jump the playback 100 DVDs forward depending on the brand of DVD player you have.	Source	All
#CLEAR	Optional. Depends on how the dealer programs the button. Typically, it's programmed to clear entries made on the keypad.	Source	All
#ENTER	Optional. Depends on how the dealer programs the button. Typically, it's programmed to enter information keyed in on the keypad.	Source	All

CD Player Commands

Command: #PREV

Select the previous song to be played.

Example: #@ CD Player 1:~UDP10.15.0.11_5000#prev

An IR command to select the previous song is sent to the CD player.

Command: #NEXT

Select the next song to be played.

Example: #@CD Player 1:~UDP10.15.0.11_5000#next

An IR command NEXT is sent to the CD player.

Command: #PWR ON

Send a power ON command to the specified audio source.

Example: #@CD Player1:~UDP10.15.0.11_5000#pwr on

An IR command to turn power on is sent to the CD player.

Command: #PWR OFF

Send a power OFF command to the specified audio source.

Example: #@CD Player1:~UDP10.15.0.11_5000#pwr off

An IR command to turn power off is sent to the CD player.

Command: #PLAY

Send a play command to the specified audio source.

Example: #@CD Player 1:~UDP10.15.0.11_5000#play

An IR command of play is sent to the CD player.

Command: #PAUSE

Send a pause command to the specified audio source.

Example: #@ CD Player 1:~UDP10.15.0.11_5000#pause

An IR command of pause is sent to the CD player.

Command: #STOP

Send a stop command to the specified audio source.

Example: #@CD Player1:~UDP10.15.0.11_5000#stop

If the audio source is an audio stream proxy, then that stream is stopped. If the audio source is a component like a CD player, then an IR command for stop is sent to the CD player.

Command: #REV

Send a generic Rev command to the specified audio source.

Example: #@CD Player 1:~UDP10.15.0.11_5000#rev

This is a generic IR command. A CD player interprets this as a Rev command.

Command: #FWD

Send a generic Fwd command to the specified audio source.

Example: #@CD Player 1:~UDP10.15.0.11_5000#fwd

This is a generic IR command. A CD player interprets this as a Fwd command.

Command: #DISC PREV

Select the previous disc on the specified audio source.

Example: #@CD Player 1:~UDP10.15.0.11_5000#disc prev

An IR command to select the previous disc is sent to the CD player.

Command: #DISC NEXT

Select the next disc on the specified audio source.

Example: #@DVD 2:~UDP10.15.0.11_5000#disc next

An IR command to select the next disc is sent to the DVD player.

Command: #RANDOM

Select the random mode on the specified audio source.

Example: #@CD Player 1:~UDP10.15.0.11_5000#random

An IR command to select random mode is sent to the CD player.

Command: #REPEAT TOGGLE

Select the repeat toggle mode on the specified audio source.

Example: #@CD Player 1:~UDP10.15.0.11_5000#repeat toggle

An IR command to set repeat mode to toggle on or off is sent to the CD player.

Command: #KEY x

Set the key on the specified audio source.

Where x = 1-9

Example: #@CD Player 1:~UDP10.15.0.11_5000#key 5
An IR command to select key 5 is sent to the CD player.

Command: #Fx

Set the key on the specified audio source.
Where x = 1 or 2

Example: #@CD Player 1:~UDP10.15.0.11_5000#F1
An IR command to select the F1 key is sent to the CD player.

Command: #KEY x_PLUS

Set the key on the specified audio source.
Where x = ten or hundred

Example: #@CD Player 1:~UDP10.15.0.11_5000#key ten_plus
An IR command to select key TEN_PLUS is sent to the CD player.

Command: #CLEAR

Optional. Depends on how the dealer programs the button. Typically, it's programmed to clear entries made on the keypad.

Example: #@CD Player1 :~UDP10.15.0.11_5000#clear
An IR command to select clear is sent to the CD player.

Command: #ENTER

Optional. Depends on how the dealer programs the button. Typically, it's programmed to enter information keyed in on the keypad.

Example: #@CD Player1 :~UDP10.15.0.11_5000#enter
An IR command to select enter is sent to the CD player.

TUNER Commands

Command: #PREV

Select the previous song to be played.

Example: #@Tuner 1:~UDP10.15.0.11_5000#prev
An IR command to select the previous song is sent to the tuner.

Command: #NEXT

Select the next song to be played.

Example: #@Tuner 1:~UDP10.15.0.11_5000#next

An IR command NEXT is sent to the tuner.

Command: #PWR ON

Send a power ON command to the specified audio source.

Example: #@Tuner 1:~UDP10.15.0.11_5000#pwr on

An IR command to turn power on is sent to the tuner.

Command: #PWR OFF

Send a power OFF command to the specified audio source.

Example: #@Tuner 1:~UDP10.15.0.11_5000#pwr off

An IR command to turn power off is sent to the tuner.

Command: #TUNE x

Select Tune UP or Tune DOWN on the specified audio source.

x = UP or DN

Example: #@Tuner 1:~UDP10.15.0.11_5000#tune up

An IR command to select Tune UP mode is sent to the tuner.

Command: #SEEK x

Select Seek UP or Seek DN on the specified audio source.

x = UP or DN

Example: #@Tuner 1:~UDP10.15.0.11_5000#seek up

An IR command to select SEEK UP mode is sent to the tuner.

Command: #SCAN x

Select Scan UP or Scan DOWN on the specified audio source:

x = UP or DN

Example: #@Tuner 1:~UDP10.15.0.11_5000#scan up

An IR command to select Scan UP mode is sent to the tuner.

Command: #PRESET x

Select a Preset button on the specified audio source.

x = 1-6, UP or DN

Example: #@Tuner 1:~UDP10.15.0.11_5000#preset 5

An IR command to select Preset button # 5 is sent to the tuner.

Command: #BAND NEXT

Select the next band on the specified audio source.

Example: #@Tuner 1:~UDP10.15.0.11_5000#Band NEXT

An IR command to select the NEXT Band is sent to the tuner.

Command: #KEY x

Set the key on the specified audio source.

Where x = 1-9, A,B

Example: #@Tuner 1:~UDP10.15.0.11_5000#key 5

An IR command to select key 5 is sent to the tuner.

Command: #CLEAR

Optional. Depends on how the dealer programs the button. Typically, it's programmed to clear entries made on the keypad.

Example: #@Tuner1 :~UDP10.15.0.11_5000#clear

An IR command to select clear is sent to the tuner.

Command: #ENTER

Optional. Depends on how the dealer programs the button. Typically, it's programmed to enter information keyed in on the keypad.

Example: #@Tuner 1 :~UDP10.15.0.11_5000#enter

An IR command to select enter is sent to the tuner.

DVD Player Commands

Command: #PWR ON

Send a power ON command to the specified video source.

Example: #@DVD Player1:~UDP10.15.0.11_5000#pwr on

An IR command to turn power on is sent to the DVD player.

Command: #PWR OFF

Send a power OFF command to the specified video source.

Example: #@DVD Player1:~UDP10.15.0.11_5000#pwr off

An IR command to turn power off is sent to the DVD player.

Command: #PLAY

Send a play command to the specified video source.

Example: #@DVD Player 1:~UDP10.15.0.11_5000#play

An IR command of play is sent to the DVD player.

Command: #PAUSE

Send a pause command to the specified video source.

Example: #@DVD Player 1:~UDP10.15.0.11_5000#pause

An IR command of pause is sent to the DVD player.

Command: #STOP

Send a stop command to the specified video source.

Example: #@DVD Player1:~UDP10.15.0.11_5000#stop

If the video source is a video stream proxy, then that stream is stopped. If the video source is a component like a DVD player, then an IR command for stop is sent to the DVD player.

Command: #PREV

Select the previous video to be played.

Example: #@DVD Player 1:~UDP10.15.0.11_5000#prev

An IR command to select the previous song is sent to the DVD player.

Command: #NEXT

Select the next video to be played.

Example: #@DVD Player 1:~UDP10.15.0.11_5000#next

An IR command NEXT is sent to the DVD player.

Command: #REV

Send a generic Rev command to the specified video source.

Example: #@DVD 2:~UDP10.15.0.11_5000#rev

This is a generic IR command. A DVD player interprets this as a Rev command.

Command: #FWD

Send a generic Fwd command to the specified video source.

Example: #@DVD 1:~UDP10.15.0.11_5000#fwd

This is a generic IR command. A DVD player interprets this as a Fwd command.

Command: #TOPMENU

Takes the disc navigation to the top menu stored on the DVD disc.

Example: #@DVD 1:~UDP10.15.0.11_5000#topmenu

This is a generic IR command. A DVD player interprets this as a top menu command.

Command: #MENU

Controls the menu functions and allows you to change DVD player settings.

Example: #@DVD 1:~UDP10.15.0.11_5000#menu

This is a generic IR command. A DVD player interprets this as a menu command.

Command: #SELECT

Selects the highlighted item.

Example: #@DVD 1:~UDP10.15.0.11_5000#select

This is a generic IR command. A DVD player interprets this as a select command.

Command: #SETUP

Brings up the onscreen setup menu for the DVD player.

Example: #@DVD 1:~UDP10.15.0.11_5000#setup

This is a generic IR command. A DVD player interprets this as a setup command.

Command: #DISPLAY

Brings up an onscreen menu of the video settings for the DVD player.

Example: #@DVD 1:~UDP10.15.0.11_5000#menu

This is a generic IR command. A DVD player interprets this as a display command.

Command: #DVDAUDIO

Brings up an onscreen menu of the audio settings for the DVD player.

Example: #@DVD 1:~UDP10.15.0.11_5000#dvdaudio

This is a generic IR command. A DVD player interprets this as a DVD audio command.

Command: #SUBTITLE

Brings up the subtitle on/off default settings for the DVD player.

Example: #@DVD 1:~UDP10.15.0.11_5000#subtitle

This is a generic IR command. A DVD player interprets this as a subtitle command.

Command: #LANG

Brings up the language default settings for the DVD player.

Example: #@DVD 1:~UDP10.15.0.11_5000#lang

This is a generic IR command. A DVD player interprets this as a language command.

Command: #MODE

Changes the mode of play from repeat to shuffle (or other presets programmed by the installer).

Example: #@DVD 1:~UDP10.15.0.11_5000#mode

This is a generic IR command. A DVD player interprets this as a mode command.

Command: #EJECT

Opens or closes the DVD tray on the DVD player.

Example: #@DVD 1:~UDP10.15.0.11_5000#eject

This is a generic IR command. A DVD player interprets this as an eject command.

Command: #ANGLE

If your DVD has different angles of the same scene, you can press this button to view the scene from a different angle.

Example: #@DVD 1:~UDP10.15.0.11_5000#angle

This is a generic IR command. A DVD player interprets this as an angle command.

Command: #ZOOM

If your DVD disc allows you to zoom in on a scene, pressing this button will activate the Zoom feature.

Example: #@DVD 1:~UDP10.15.0.11_5000#menu

This is a generic IR command. A DVD player interprets this as a zoom command.

Command: #RETURN

Returns to the previous menu (or performs as programmed by the installer).

Example: #@DVD 1:~UDP10.15.0.11_5000#return

This is a generic IR command. A DVD player interprets this as a return command.

Command: #NAV x

Send navigation commands to the specified video source.

Where x = DN, UP, LT, RT

Example: #@DVD 1:~UDP10.15.0.11_5000#NAV DN

An IR command to select navigation DN is sent to the DVD player.

Command: #KEY x

Set the key on the specified video source.

Where x = 1-9, A-Z, F1-F12

Example: #@DVD 1:~UDP10.15.0.11_5000#key 5

An IR command to select key 5 is sent to the DVD player.

Command: #Fx

Set the key on the specified video source.

Where x = 1 or 2

Example: #@DVD Player 1:~UDP10.15.0.11_5000#F1

An IR command to select the F1 key is sent to the DVD player.

Command: #KEY x_PLUS

Set the key on the specified video source.

Where x = ten or hundred

Example: #@DVD 1:~UDP10.15.0.11_5000#key ten_plus

An IR command to select key TEN_PLUS is sent to the DVD player.

Command: #CLEAR

Optional. Depends on how the dealer programs the button. Typically, it's programmed to clear entries made on the keypad.

Example: #@DVD 1:~UDP10.15.0.11_5000#clear

An IR command to select clear is sent to the DVD player.

Command: #ENTER

Optional. Depends on how the dealer programs the button. Typically, it's programmed to enter information keyed in on the keypad.

Example: #@DVD 1 :~UDP10.15.0.11_5000#enter

An IR command to select enter is sent to the DVD player.

Source ServiceType Commands

Table 5-2

MediaLinX Source Service Type Commands

ASCII String	Comments	Service Type	Model
#SET AUDIO, ANALOG	Set the audio input to analog on the specified audio source.	Source	All
#SET AUDIO, GAIN, x	Set the audio gain on the specified audio source.	Source	All
#SET AUDIO, SPDIF	Set the audio input to S/PDIF on the specified audio source.	Source	All
#MENU_LIST m,n {{path}}	Request a list of media associated with a media server.	Source	All
#MENU_SEL {{path}}	Select a specific item from the MENU_LIST	Source	All
#QUERY SOURCE	Show a report on the current song being played	Source	All

Command: #SET AUDIO, ANALOG

Set the audio input for analog on the specified audio source.

Example: #@Tuner 1:~UDP10.15.0.11_5000#set audio,analog

An ASCII command to set the audio output to analog is sent to the audio source service.

Command: #SET AUDIO, GAIN, x

Set the audio gain on the specified audio source.

x = 0-6

Example: #@Tuner 1:~UDP10.15.0.11_5000#set audio,gain,5

An ASCII command to set the audio gain level is sent to the audio source service.

Command: #SET AUDIO, SPDIF

Set the audio input for S/PDIF on the specified audio source.

Example: #@Tuner 1:~UDP10.15.0.11_5000#set audio,SPDIF

An ASCII command to set audio output to SPDIF is sent to the audio source service.

Command: #MENU_LIST m,n,{{path}}

Request a List of Media

Request a list of media associated with a media server.

Example: #@Stream 1:~UDP10.15.0.11_5000#menu_list 1,10,media

- This message is addressed to Stream 1. This is a stream service type message. The command is requesting menu items one to ten of the media associated with this service.
- media is the default path name for the first menu list.

Responses Received

First Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP
{{<item idpath="media" disppath="media" itemnum="1" id="All Songs"
display="All Songs" children="1078" />}}
```

The next level menu path for this list is gotten by concatenating item idpath and id separated by a >.

Example: {{media>"All Songs"}} becomes the next menu path.

- Itemnum = "1", indicates this is the first item on this menu list.
- "All Songs" is the display name for this list.
- children="1078", indicates that this is not a terminal node (i.e., there is another menu list below this level).

Second Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP
{{<item idpath="media" disppath="media" itemnum="2" id="Artists"
display="Artists" children="63" />}}
```

The next level menu path for this list is gotten by concatenating idpath and id separated by a >. Example: {{media>Artists}} becomes the next menu path.

- Itemnum = "2", indicates this is the second item on this menu list.
- "Artists" is the display name for this list.
- children="63", indicates that this is not a terminal node. ie: there is another menu list below this level.

Third Through Fifth Responses

These responses can be similarly parsed.

- #@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP {{<item idpath="media" disppath="media" itemnum="3" id="Albums" display="Albums" children="116" />}}

- ❑ `#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP {{<item idpath="media" disppath="media" itemnum="4" id="Genres" display="Genres" children="12" />}}`
- ❑ `#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP {{<item idpath="media" disppath="media" itemnum="5" id="Playlists" display="Playlists" children="3" />}}`

Sixth Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4750:Stream 1 #MENU_RESP
{{<item idpath="media" disppath="media" itemnum="-1" />}}
```

Itemnum = "-1". The negative value indicates there are no more items on this list.

As long as children = positive number, there are more items under that list.

However, children = "0" is not a reliable indicator of the end of the list. itemnum = negative is more reliable to decide if the end of the list has been reached.

Summary

The service Stream 1 menu has 5 items which are displayed as:

- ❑ All Songs
- ❑ Artists
- ❑ Albums
- ❑ Genre
- ❑ Playlists

Each of these branches has children (another menu list) under them.

Traversing the Menu List

Traversing down the menu list.

As explained above, the next level menu path is gotten by concatenating item idpath and id separated by a >.

`{{media>All Songs}}` becomes the next menu path.

Example: `#@Stream 1:~UDP10.15.0.11_5000#menu_list 1,5,{{media>All Songs}}`

The command is requesting menu items 1 to 5 of the menu list below the list All Songs, which was the first item on the previous menu list.

Responses Received

First Response

```
#@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP
{{<song idpath="media>All Songs" disppath="media>All Songs"
itemnum="1" id="1481" display="Brown, Crane, Ker, Finesilver" children="0" /
>}}
```

The next level menu path for this list is gotten by concatenating item idpath and id separated by a >.

Example: `{{media>All Songs>1481}}` becomes the next menu path for Brown, Crane, Ker, Finesilver

- ❑ Itemnum = "1", indicates this is the first item on this menu list.
- ❑ The display name = Brown, Crane, Ker, Finesilver for this item.
- ❑ children=0, indicates that this is a terminal node. ie: there are no items below this level. Therefore a further #MENU_LIST command would neither get a response nor a reaction. Instead, as explained below, a #MENU_SEL command must be issued.

Responses Two through Five

These responses can be similarly parsed.

- ❑ `##@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP {{<song idpath="media>All Songs" disppath="media>All Songs" itemnum="2" id="1648" display="(Da Le) Yaleo" children="0" />}}`
- ❑ `#@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP {{<song idpath="media>All Songs" disppath="media>All Songs" itemnum="3" id="1606" display="(Hidden Track)" children="0" />}}`
- ❑ `#@SL22004285001161000172_4~TCP10.15.0.11_4781:Stream 1 #MENU_RESP {{<song idpath="media>All Songs" disppath="media>All Songs" itemnum="4" id="1974" display="11 O'Clock Tick Tock" children="0" />}}`
 - Itemnum = "5". The positive value indicates there are more items on this list.
 - Children = "0", in each of these items indicates that there are no more items under these items.

Summary

All Songs which was the first item on the menu has a list of items 1 to 5 which are displayed as:

- ❑ Brown, Crane, Ker, Finesilver
- ❑ (Da Le) Yaleo
- ❑ (Hidden Track)
- ❑ 11 OClock Tick Tock
- ❑ 2000 Miles

The All Songs menu had shown it had 1078 children of which the command `#menu_list 1,5,{{media>All Songs}}` requested a list of items 1-5.

Similarly, if the integrator wanted to see the list under the second item, Artists, the command syntax would be:

```
#@Stream 1:~UDP10.15.0.11_5000#menu_list 1,5,{{media>Artists}}
```

Listing Source Presets

Example: #@XM Tuner:~UDP10.15.0.11_5000#menu_list 1,10, presets

This message is addressed to the XM Tuner. The command is requesting menu items 1 to 10 of the media associated with this service. **'presets'** is the default path name for the first menu list.

Responses Received

First Response Received

```
#@MLA10105278001161009F16_4~TCP10.15.1.15_2950:XM Tuner
#MENU_RESP {{<preset id="1" children="0" itemnum="1" display="XM
Cafe" command="#Channel 45" stationName="XM Cafe"
stationNumber="45" />}}
```

2nd - 6th Responses Received

```
#@MLA10105278001161009F16_4~TCP10.15.1.15_2950:XM Tuner
#MENU_RESP {{<preset id="2" children="0" itemnum="2" display="Top
Tracks" command="#Channel 46" stationName="Top Tracks"
stationNumber="46" />}}
```

```
#@MLA10105278001161009F16_4~TCP10.15.1.15_2950:XM Tuner
#MENU_RESP {{<preset id="3" children="0" itemnum="3" display="Big
Tracks" command="#Channel 49" stationName="Big Tracks"
stationNumber="49" />}}
```

```
#@MLA10105278001161009F16_4~TCP10.15.1.15_2950:XM Tuner
#MENU_RESP {{<preset id="4" children="0" itemnum="4" display="The
Loft" command="#Channel 50" stationName="The Loft" stationNumber="50"
/>}}
```

```
#@MLA10105278001161009F16_4~TCP10.15.1.15_2950:XM Tuner
#MENU_RESP {{<preset id="5" children="0" itemnum="5" display="The
Verge" command="#Channel 52" stationName="The Verge"
stationNumber="52" />}}
```

```
#@MLA10105278001161009F16_4~TCP10.15.1.15_2950:XM Tuner
#MENU_RESP {{<preset id="6" children="0" itemnum="6" display="Lucy"
command="#Channel 54" stationName="Lucy" stationNumber="54" />}}
```

7th Response Received

```
#@MLA10105278001161009F16_4~TCP10.15.1.15_2950:XM
Tuner#MENU_RESP {{<presets idpath="presets" itemnum="-1" />}}
```

Itemnum = -1. The negative value indicates there are no more items on this list.

Command: #MENU_SEL {{path}}

Select a specific item from the MENU LIST.

When a MENU_LIST item has children=0, then a #MENU_SEL command must be issued to select that item.

Selecting a Song from the Menu List

Example: #@Stream 1:~UDP10.15.0.11_5000#menu_sel {{ media>All Songs>1481}}

Selects the song, Brown, Crane, Ker, Finesilver (id=1481) from the All Songs menu to be played.

Selecting a Preset

Example: #@XM Tuner:~UDP10.15.0.11_5000#menu_sel {{presets>1}}

This command selects the Channel 45 “XM Café” stored in preset id 1.

Command: #QUERY SOURCE

Get details on the current media being played.

Example: #@Stream 1:~UDP10.15.0.11_5000#QUERY SOURCE

Response Received

```
#@SL22004285001161000172_4~UDP 10.15.0.11_4028:Stream 1#REPORT
{{<report type="source" artwork="http://10.15.100.116/7.0.5/Database/Music/
Covers/s4/s3/Nails-Corpus Christi" display="song/artist/album/genre" song="14
Dreams" album="Corpus Christi" artist="Nails" genre="Rock" time="376"
percent="59" sngPIIndex="79" sngPITotal="11169" source="Stream 1"
elapsed="223744" next="14 Years" active="2" controlState="PAUSE" shuffle="0"
/>}}
```

This command gives a report of the current song in Stream 1

The attribute, display, gives a suggestion of which items to display and with what priority.

In this example the audio source Stream 1 is suggesting that the song title is the most important item to display; followed by the artist's name, followed by the name of the album and the genre.

The UI device is not bound by the suggestion.

NOTE: #REPORT responses shall in no case exceed 1000 characters, but data may be distributed among multiple #REPORT messages.

For all types of status/error reports, the attribute display is used for messages intended for a user or operator. User interface devices should make an attempt to display these messages.

Root ServiceType Commands

Table 5-3
MediaLinX Root Service Type Commands

ASCII String	Comments	Service Type	Model
#QUERY SERVICE	List configuration info for all services	Root	All
#QUERY SERVICE,{{service name}}	show configuration info for specific service	Root	All
#RESTART	Restart the device	Root	All

Command: #QUERY SERVICE

List Services configured with a device.

The third party controller needs to send a UDP message to the IP address of a specific device with the following command, to get a list of Services configured with that device.

Query service is a root service command and must be addressed to a specific device.

The root service on every device is always enabled. Root service commands do not need a @toAddress.

Example: #query service

Response Received

```
#@ ~UDP10.15.0.11_5000:MLA10105151001161008019_4#REPORT {{<report
type="state" serviceName1="myXM"
serviceName2="MLA10105151001161008019_2" serviceName3="Stream 3"
serviceName4="MLA10105151001161008019_4"
serviceName5="MLA10105151001161008019_5" />}}
```

During configuration, serviceNames were not assigned to 2,4, and 5, hence *StreamNet* defaults to using the serial number of the device (as in serviceName2=MLA10105151001161008019_2).

Each service can be further drilled down (see the following):

Command: #QUERY SERVICE {{serviceName}}

List configuration settings for a specific service.

Example: #query service {{myXM}}

Response Received

```
#@ ~UDP10.15.0.11_5000:MLA10105151001161008019_4#REPORT {{<report
type="state" serviceName="myXM" serviceType="audio/source"
IP="10.15.96.149" permId="MLA10105151001161008019_1" enabled="1"}}
```

```
sourceType="XM TUNER" roomName="MLA10105151001161008019_1"  
groupName="ALL" />}}
```

Response shows the myXM is of Service Type = Source and therefore only Source type commands are recognized by this service.

Service names are case sensitive when used in the argument field.

During configuration, a roomName was not assigned; therefore *StreamNet* uses the serial number of the device as in "MLA10105151001161008019_1".

Command: #RESTART

Restart the device.

TouchLinX Commands

The following commands can only be used on a *TouchLinX*. Commands must be issued to the appropriate serviceType specifying the serviceName in the toAddress.

UI/GUI Type Commands

Table 6-1
TouchLinX UI/GUI Commands

ASCII String	Comments	Service Type	Model
#ACTIVE OFF	Enter the inactive state	UI/GUI	All
#ACTIVE ON	Enter the active state	UI/GUI	All
#QUERY UI	Get a report on the UI settings	UI/GUI	All
#SET BKLT, TIMEOUT, x	Set the timeout value for the UI backlight	UI/GUI	All
#LEVEL_UP BKLT	Increase backlight level	UI/GUI	All
#LEVEL_DN BKLT	Decrease backlight level	UI/GUI	All
#LEVEL_SET VOL, x	Set feedback sound On/Off	UI/GUI	All

Command: #ACTIVE OFF

Set the UI/GUI Service to OFF (inactive).

Example: #@Room 1 TL UI:~UDP10.15.0.11_5000#ACTIVE OFF

The UI/GUI service; Room 1 TL UI, is set to the off state.

NOTE: In this state, the UI/GUI service will not listen to the audio source. In effect, if there was an audio source being rendered, the rendering will stop (i.e., no music will be heard).

Command: #ACTIVE ON

Set the audio renderer Service to ON (active).

Example: #@Room 1 TL UI:~UDP10.15.0.11_5000#ACTIVE ON

The UI/GUI service; Room 1 TL UI, is set to the ON state.

Command: #QUERY UI

Get a report on the UI settings.

Example: #@Room 1 TL UI:~UDP10.15.0.11_5000#query ui

Response Received

```
#@~UDP10.15.0.11_5000:Room 1 TL UI#REPORT {{<report type="state"
bkltLevel="100" bkltTimeout="300" vol="100" />}}
```

Command: #SET BKLT, TIMEOUT, x

Set the timeout value for the UI backlight.

x = number of minutes

Example: #@Room 1 TL UI:~UDP10.15.0.11_5000#set bklt, timeout, 5

Sets the backlight timeout value = 5 minutes.

Command: #LEVEL_UP BKLT

Increase UI backlight level.

Example: #@Room 1 TL UI:~UDP10.15.0.11_5000#level_up bklt

Command: #LEVEL_DN BKLT

Decrease UI backlight level.

Example: #@Room 1 TL UI:~UDP10.15.0.11_5000#level_dn bklt

Command: #LEVEL_SET VOL, x

Sets feedback sound On or Off.

- ❑ x is a number between 0 or 100.
- ❑ x = 0 will set the volume to the minimum level.
- ❑ x =100 will set the volume to the maximum level.

Example: #@ Room 1 TL UI:~UDP10.15.0.11_5000#LEVEL_SET VOL, 30

The UI/GUI service; Room 1 TL UI, sets the volume level based on the value x.

Intercom ServiceType Commands

Setup Program Guidelines

- Only one service per room may have monitorEnable enabled.
- Only one service per room may have defaultMic enabled.
- All intercom services must be a member of the group NS_IC_ALL.

A range of IP addresses must be configured for Intercom (see the *DigiLinX Dealer Setup* for the default set).

Table 6-2
TouchLinX Commands

ASCII String	Comments	Service Type	Model
#INTERCOM JOIN, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD] [,IMMED]	The intercom service joins an intercom session.	Intercom	All
#INTERCOM JOINMON, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD]	The intercom service joins an intercom monitoring session.	Intercom	All
#INTERCOM MONITOR, roomName, audience	roomName is room to be monitored and audience are the monitoring room(s) for that room	Intercom	All
#INTERCOM LEAVE [, tag]	Terminate the intercom session	Intercom	All
#INTERCOM LEAVE [, monitor]	Terminate intercom Monitor session	Intercom	All
#INTERCOM LEVEL_UP VOL	Increase intercom service volume level	Intercom	All
#INTERCOM LEVEL_DN VOL	Reduce intercom service volume level	Intercom	All
#INTERCOM LEVEL_SET VOL, n	Set intercom service volume level	Intercom	All
#INTERCOM PTT, tag	Activate microphone for 300ms	Intercom	All
#INTERCOM PTT OFF	Disable microphone	Intercom	All
#DND x	Set Do Not Disturb state	Intercom	All
#PRIVACY x	Set Privacy state	Intercom	All

Command: #INTERCOM JOIN, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD] [,IMMED]

Join an intercom session.

- ❑ ID = the unique name of the intercom session. An attempt to start a second session with the same ID will fail because the Intercom services will ignore a JOIN command with an ID that is already registered. The default intercom session has the ID of INTERCOM.

NOTE: Members may be added to a session by sending out a join message with the same ID and tag but addressed to different services.

- ❑ tag = a 32 bit session number that will be used to identify the session in the header of the intercom audio data and to reference it in other commands.
- ❑ listenIpAdr = the IP address to which the device should listen. It may be multicast or unicast.
- ❑ listenPort = the IP port to which the device should listen.
- ❑ TalkIpAdr = the IP address to which the device should address replies. It may be multicast or unicast.
- ❑ talkPort = the IP port to which the device should address replies.

- ❑ OVRRD = a string which if present overrides the do not disturb setting and initiates a session regardless. [OVRRD] does NOT override configuration flags.
- ❑ IMMED = a string which if present will cause this new session to immediately become the active session.

NOTE: For most intercom sessions it is expected that both the talk and listen addresses are the same and are multicast addresses.

When point to point (unicast) sessions are desired, these addresses would be the unicast addresses of the two participants - although even for point to point the addresses MAY be multicast. TalkIpAdr would be the address of the originator and ListenIpAdr would be the second service.

A service could specify a multicast listen address and its unicast address for talk. This would create a situation where every destination would hear the originator but only the originator would hear the responses.

Example: #@Room 1 Intercom:~UDP10.15.0.11_5000#intercom join, intercom1, 12345678, 239255481, 5005, 239255481, 5005, OVRRD, IMMED

The join message adds the specified session intercom1, to the intercom service's list of sessions.

Command: #INTERCOM JOINMON, ID, tag, listenIpAdr, listenPort , talkIpAdr, talkPort [,OVRRD]

Join and monitor an intercom session.

Example: #@Room 1 Intercom:~UDP10.15.0.11_5000#intercom join, intercom1, 12345678, 239255481, 5005, 239255481, 5005, OVRRD

The Joinmon command has exactly the same format and effect as the join command except the monitoring flag is set for the intercom session.

NOTE: The OVRRD will override the Privacy setting instead of the DnD setting.

The intercom service will execute an #INTERCOM JOINMON command even if it does not have monitor enabled.

Command: #INTERCOM MONITOR, roomName, audience

Set monitored and monitoring rooms.

- ❑ roomName is the name of the room which is to be monitored.
- ❑ audience is a service, room, or group which will monitor the specified room.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom monitor, Room1, Room2

Command: #INTERCOM LEAVE, [tag]

Terminate an intercom session

NOTE: [tag] is optional

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom leave, 12345678

The Intercom session service Room 1 Intercom will be terminated.

Command: #INTERCOM LEAVE, monitor

Terminate a monitor session.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom leave, monitor

The Intercom monitor session Room 1 Intercom will be terminated.

Command: #INTERCOM LEVEL_UP VOL

Increase intercom volume level.

NOTE: This does not affect the volume settings for the renderer.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom level_up vol

Command: #INTERCOM LEVEL_DN VOL

Decrease intercom volume level.

NOTE: This does not affect the volume settings for the renderer.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom level_dn vol

Command: #INTERCOM LEVEL_SET VOL, x

Set intercom volume level.

NOTE: This does not affect the volume settings for the renderer.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom level_set vol, 30

Command: #INTERCOM PTT, [tag]

Activate microphone while talking.

NOTE: The command must be repeated at least every 300ms to keep microphone on.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom ptt, 12345678

Command: #INTERCOM PTT OFF

Disable microphone immediately.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#intercom ptt, off

Command: #DND x

Set Do Not Disturb state.

x = ON or OFF

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#DND on

If ON is set, the intercom service will not join any intercom session.

Command: #PRIVACY x

Set privacy state.

x = ON or OFF

NOTE: If ON is set, the intercom service will not allow itself to be monitored.

Example: #@ Room 1 Intercom:~UDP10.15.0.11_5000#Privacy on

Sample Intercom Scenarios

Simple Scenario

On the user interface there is a single Talk button. Pressing this button enables the user to talk to the entire house.

When the Talk button is pressed the controller sends out a #INTERCOM JOIN message addressed to the all intercom session. It would also send a #INTERCOM PTT message for the session to the local room and repeat the #INTERCOM PTT message every 300ms for as long as the button is held. When the button is released the controller should send a #INTERCOM PTT, off message for the session to the local room.

Example:

If a user in the bedroom presses the Talk button:

- #@NS_IC_ALL:~UDP10.15.0.11_5000#intercom join, intercom1, 12345678, 239255481, 5005, 239255481, 5005, OVRRD, IMMED
- #@Bedroom#INTERCOM PTT (every 300ms, while PTT button is pressed)
- #@Bedroom#INTERCOM PTT, OFF (when the PTT button is released)

Complete Control Scenario

On the user interface are a block of Call buttons. Each button specifies a section of the house or room or group of rooms (these groupings will be called sessions). This button is also a PTT button for that session.

When a session button is pressed, the controller sends out an #INTERCOM JOIN message addressed to the session specified. It also sends an #INTERCOM PTT message for the session to the local room and repeats the #INTERCOM PTT message every 300ms for as long as the button is held. When the button is released, the controller sends an #INTERCOM PTT, off message for the session to the local room. When a response comes in, the user interface highlights the button corresponding to the session responding.

If a user in the bedroom had selected a session button labeled “Downstairs,” the controller sends the following messages:

- ❑ #@Bedroom#INTERCOM JOIN, Downstairs, 12345678, 239255481, 5005, 239255481, 5005, OVRD, IMMED
- ❑ #@Downstairs# INTERCOM JOIN, Downstairs, 12345678, 239255481, 5005, 239255481, 5005, OVRD, IMMED
- ❑ #@Bedroom#INTERCOM PTT, 12345678 (every 300ms)
- ❑ #@Bedroom#INTERCOM PTT, OFF (when the button is released)

TouchLinX Root ServiceType Commands

Table 6-3

TouchLinX Root Service Commands

ASCII STRING	COMMENTS	SERVICE TYPE	MODEL
#QUERY SERVICE	List configuration info for all services	Root	All
#QUERY SERVICE,{{service name}}	show config info for specific service	Root	All
#MENU_LIST m,n,Intercom	List of intercom services	Root	All
#MENU_LIST m,n,renderers	List of renderers	Root	All
#RESTART	Restart the device	Root	All

Command: #QUERY SERVICE

List Services configured with a device.

Query service is a root service command and must be addressed to a specific device or service under that device.

The root service on every device is always enabled. Root service commands do not need a @toAddress.

The third party controller needs to send a UDP message to the IP address of a specific device with the following command, to get a list of services configured with that device.

Example: #query service

Response Received

```
#@ ~UDP10.15.0.11_5000:TL3800538001161007532_1#REPORT { {<report
type="state" serviceName1="TL3800538001161007532_1"
serviceName2="Room 1 TL UI" />}}
```

Each service can be further drilled down as described in the following sections.

Command: `#@Room 2:~UDP10.15.0.11_5000#query service`

List services configured for all devices in a room.

Responses Received

Response # 1

```
#@ ~UDP10.15.0.11_5000:TL3800538001161007532_1#REPORT {{<report
type="state" serviceName1="TL3800538001161007532_1"
serviceName2="Room 1 TL UI" />}}
```

NOTE: Response # 1 is received from the *TouchLinX*. It has two services associated with it:

- ❑ `serviceName1="TL3800538001161007532_1"` – is of type root
 - ❑ `serviceName2="Room 2 TL UI"` – is of type "UI"
-

The root service name is always the serial number of the device.

Response # 2

```
#@ ~UDP10.15.0.11_5000:SL22004280001161000057_4#REPORT {{<report
type="state" serviceName1="Local Source 2" serviceName2="Stream 1"
serviceName3="Room 2 Player" serviceName4="SL22004280001161000057_4"
serviceName5="Room 2 KL" />}}
```

Query Service to a room will get a response from the root service of each device in the room displaying the services on each device.

NOTE: Since the `toAddress` is specified, the root services on all devices for Room 2 will respond to this message.

Response # 2 is received from the SpeakerLinX. It lists 5 services:

- ❑ `serviceName1="Local Source 2"`
- ❑ `serviceName2="Stream 1"`
- ❑ `serviceName3="Room 2 Player"`, etc.

Command: #QUERY SERVICE {{serviceName}}

List configuration settings for a specific service.

- ❑ `#@Room 1:~UDP10.15.0.11_5000#query service {{Room 1 TL UI}}`

Response Received:

- ❑ `#@ ~UDP10.15.0.11_5000:TL3800538001161007532_1#REPORT {{<report
type=state serviceName=Room 1 TL UI serviceType=ui/flash IP=10.15.117.50
permId=TL3800538001161007532_2 enabled=1 roomName=Room 1
groupName=ALL />}}`

Response shows the Room 1 TL UI is of Service Type = ui/flash and therefore only UI type commands will be recognized by this service.

NOTE: Service names are case sensitive when used in the argument field.

MUST be bound by the double curly brackets, as in {{Room 1 TL UI}}. The double curly brackets are required since the Service name has spaces (non alpha-numeric characters)

Command: #MENU_LIST m,n,INTERCOM

Details on this will be available in future versions of this document.

Command: #MENU_LIST m,n,RENDERERS

Get a list of renderer services.

m,n specifies the range of items to be listed from the list of renderers. m = item number on the menu list of the first item to be listed, through n = the last item in the range.

Example: #MENU_LIST 1,5,RENDERERS

Menu lists are interrogative and therefore get responses.

This command will get a response for renderer services 1 through 5.

NOTE: m=3 and n=5, would show the third, fourth and fifth item.

The command is requesting a list of 1-5 renderers associated with this root service (starting with the first item on the menu). The response can come in a series of messages depending on the number of renderers in the system.

Response Received

- #@MLA10105151001161008019_4~UDP10.15.0.11_2174:Room 1 Player#MENU_RESP {{<renderer id="Room 1 Player" children="0" itemnum="2" idpath="renderers" disppath="renderers" itemnum="2" display="Room 1 Player" ip="10.15.1.114" roomName="Room 1" permId="SL22004285001161000172_3" />}}
- #@MLA10105151001161008019_4~UDP10.15.0.11_2174:Room 1 Player#MENU_RESP {{<renderer id="Room 1 Player" children="0" itemnum="-2" idpath="renderers" disppath="renderers" itemnum="-2" display="Room 1 Player" ip="10.15.1.114" roomName="Room 1" permId="SL22004285001161000172_3" />}}

The response shows there are two renderers in this project with service names Room 1 Player and Room 2 Player.

- Room 1 Player is associated with roomName=Room 1
- Room 2 Player is associated with roomName=Room 2

NOTE: children=0, indicates there a no further lists under this Service.

In the third response, itemnum=-2. If this is a positive number, then there are more renderers in the list. A negative number indicates the end of the list.

Although the command requested to list five renderers, since there were only two renderer services, the third response contained itemnum= -2, indicating the end of the list. If 5 responses were received with the fifth itemnum=5 (a positive number), then there are potentially more available renderer services. The command must be re-issued with a larger number (eg: #MENU_LIST 1,10,renderers) until a response is received with Itemnum = a negative number.

Command: #Restart

Restarts the device.

Example:

```
#@Room 1 TL UI:~UDP10.15.0.11_5000#restart
```

Case Sensitivities

- When the source or renderer service name is used as the toAddress; it is not case sensitive.

Example:

```
#@room 1 TL UI:~UDP10.15.0.11_5000#level_dn blkt
```

- The service name is case sensitive when used as an argument.

Example:

```
#@Room 2:~UDP10.15.0.11_5000#query service {{Room 2 TL UI}}
```

Received from server:

```
#@ ~UDP10.15.0.11_5000:TL3800538001161007532_1#REPORT {{<report  
type="state" serviceName="Room 1 TL UI" serviceType="ui/flash"  
IP="10.15.117.50" permId="TL3800538001161007532_2" enabled="1"  
roomName="Room 1" groupName="ALL" />}}
```

```
#query service {{Room 2 TL UI}} – No Response Received.
```

ControLinX Commands

The following commands can only be used on a *ControLinX*. Commands must be issued to the appropriate serviceType specifying the serviceName in the toAddress.

ControLinX GPIO Type Commands

Table 7-1
ControLinX GPIO Commands

ASCII STRING	COMMENTS	SERVICE TYPE	MODEL
#TXSERIAL port#, {{string}}	Transmit serial string	GPIO	CL100

Send a command to a GPIO service.

Command: #TXSERIAL

Port# is always set to 1. Non-alpha numeric characters can be included in the string by converting the character to % followed by the two character hexadecimal value. A carriage return would be %0D.

Example:

```
#@Control:~UDP10.15.1.15_5000#TXSERIAL 1, {{BP,1,TOG%0D}}
```

Commands the GPIO service to transmit the string “BP,1,TOG%0D” over the RS232 port.

NOTE: The #TXSERIAL command is ignored If *ControLinX* is in IR mode. In IR mode, the GPIO service will accept all of the IR control commands and send out whatever IR has been learned (refer to *Chapter 5, MediaLinX Commands* on page 5-1)

ControlLinX Root ServiceType Commands

Table 7-2
ControlLinX Root Service Type Commands

ASCII STRING	COMMENTS	SERVICE TYPE	MODEL
#QUERY SERVICE	List configuration info for all services	Root	All
#QUERY SERVICE,{{service name}}	show config info for specific service	Root	All
#RESTART	Restart the device	Root	All

List services configured with a device.

Command: #QUERY SERVICE

Query service is a Root Service command and must be addressed to a specific device or service under that device.

The root service on every device is always enabled. Root service commands do not need an @toAddress.

The third party controller must send a UDP message to the IP address of a specific device with the following command, to get a list of services configured with that device:

```
#query service
```

Response Received

```
##@CL1000534700116100C918_1~TCP10.15.1.15_1396:CL1000534700116100C918_1#REPORT {{<report type="state"
serviceName1="CL1000534700116100C918_1" serviceName2="Control"
serviceName3="CL1000534700116100C918_3" />}}
```

Command: #QUERY SERVICE {{serviceName}}

List configuration settings for a specific service.

Example:

```
##@CL1000534700116100C918_1~TCP10.15.1.15_5000#query service
{{Control}}
```

Response Received

```
##@CL1000534700116100C918_1~TCP10.15.1.15_1396:CL1000534700116100C918_1#REPORT {{<report type="state" serviceName="Control"
serviceType="gpio" IP="10.10.32.87" permId="CL1000534700116100C918_2"
enabled="1" roomName="CL1000534700116100C918_2" groupName="ALL" />}}
```

Response shows the ServiceName Control is of Service Type = GPIO, so only GPIO type commands are recognized by this service.

NOTE: Service names are case sensitive when used in the argument field.

Command: #RESTART

Restarts the device.

Example:

```
#@CL1000534700116100C918_1:~TCP10.15.1.15_5000#restart
```

Index

Symbols

%modifier3-4
/0 (null byte)3-5
@Room 2#query service4-22
@toAddress3-2, 3-5

A

ACTIVE OFF 4-1, 4-2, 6-1
ACTIVE ON 4-1, 4-2, 6-1
Address
 node3-2
 subnode3-2
Albums4-12
All Songs4-12
AMP OFF4-1, 4-3
AMP ON4-1, 4-3
ANGLE5-3, 5-11
Arguments3-4, 3-5
 arg1, arg2, ..., argN3-4
 character literals3-5
 meaningful strings3-5
 numeric scalars3-5
 special formats3-5
Artists4-12

ASCII

 characters3-1
 command3-5
 command address3-3
 message3-5
 message syntax3-2, 4-2
 message types3-5
 overview3-1
 protocol3-1

ASCII Message Types

 declarative3-5, 3-6
 imperative3-5
 interrogative3-5, 3-6

Audio

 local source 2-1, 2-2, 2-3, 2-4
 renderer 2-1, 2-2, 2-3, 2-4

 source 2-1, 2-2, 2-3, 2-4
 stream proxy 2-1, 2-2, 2-3, 2-4
Audio Renderer 2-1, 2-2, 2-3, 2-4

Audio Renderer Service

 next available source4-2
 query
 current source settings4-2
 source being rendered4-2
 sources
 available4-1
 to be rendered4-2
 specific sources4-2
 stop source4-2

Audio Source 2-1, 2-2, 2-3, 2-4

 deselect all4-9
 pausing4-14
 playing4-14
 query settings4-9
 querying current source4-10
 select
 next4-9
 specific4-9
 stopping4-15

Audio Stream Proxy 2-1, 2-2, 2-3, 2-4

Audio/Local Source 2-1, 2-2, 2-3, 2-4

B

Balance

 left4-1
 level4-1, 4-3
 level down4-4
 level up4-4
 right4-1

Band

 level4-1
 level set4-5
 level up4-1, 4-6

BAND NEXT5-2, 5-8

Bass

 level4-1
 level down4-1, 4-4
 level set4-4
 level up4-1, 4-4

Buttons

 PTT4-20
 talk4-20

C

Case Sensitivities4-24

CD Player

 commands 5-1–5-2, 5-4–5-6
 CLEAR 5-2, 5-3, 5-6, 5-8
 DISC NEXT5-1, 5-5
 DISC PREV5-1, 5-5
 ENTER 5-2, 5-3, 5-6, 5-8
 F Keys5-1, 5-6
 FWD5-1, 5-5
 HUNDRED_PLUS5-2
 KEY x5-1, 5-5
 KEY x_PLUS5-6
 NEXT5-1, 5-4
 PAUSE5-1, 5-4
 PLAY5-1, 5-4
 PREV5-1, 5-4
 PWR OFF5-1, 5-4
 PWR ON5-1, 5-4
 RANDOM5-1, 5-5
 REPEAT TOGGLE5-1, 5-5
 REV5-1, 5-5
 STOP5-1, 5-4
 TEN_PLUS5-1
 jumping to another CD5-1, 5-3

CLEAR 5-2, 5-3, 5-6, 5-8, 5-12

ControLinX2-2

 commands7-1
 GPIO7-1
 QUERY SERVICE7-2
 QUERY SERVICE,{{service-
 Name}}7-2
 RESTART7-2
 ServiceType7-2
 TXSERIAL7-1

D

 defaultMic4-16

 DigiLinX Dealer Setup3-2, 4-16

DISC NEXT5-1, 5-5
 DISC PREV5-1, 5-5
 DISPLAY5-3, 5-10
 DND
 setting state4-16, 4-19
 Do Not Disturb - *see DND*
 DVD Player
 commands 5-2-5-3, 5-8
 ANGLE5-3, 5-11
 CLEAR5-3, 5-12
 DISPLAY5-3, 5-10
 DVDAUDIO5-3, 5-10
 EJECT5-3, 5-11
 ENTER5-3, 5-13
 F Keys5-3, 5-12
 FWD5-3, 5-9
 HUNDRED_PLUS5-3
 KEY x5-3, 5-12
 KEY x_PLUS5-12
 LANG5-3, 5-11
 MENU5-3, 5-10
 menu5-3
 MODE5-3, 5-11
 NAV DN5-3
 NAV UP5-3
 NAV x5-12
 NEXT5-2, 5-9
 PAUSE5-2, 5-9
 PLAY5-2, 5-8
 PREV5-2, 5-9
 PWR OFF5-2, 5-8
 PWR ON5-2, 5-8
 RETURN5-3, 5-12
 REV5-3, 5-9
 SELECT5-3, 5-10
 SETUP5-3, 5-10
 STOP5-2, 5-3, 5-9
 SUBTITLE5-11
 SUBTITLES5-3
 TEN_PLUS5-3
 TOPMENU5-3, 5-10
 ZOOM5-3, 5-11
 DVDAUDIO5-3, 5-10

E

EJECT5-3, 5-11
 ENTER5-2, 5-3, 5-6, 5-8, 5-13

F

F Keys5-1, 5-3, 5-6
 fromAddress3-4, 3-5
 FWD5-1, 5-3, 5-5, 5-9

G

General Purpose IO - *see GPIO*
 Genre4-12
 GPIO2-1, 2-2, 2-3
 Group3-1, 3-2
 NS_IC_ALL4-16
 Group Name
 all2-1

H

HUNDRED_PLUS5-2, 5-3

I

Intercom2-1, 2-2, 2-3, 2-4
 activate microphone6-3, 6-5
 decrease volume ..4-16, 4-18, 6-3, 6-5
 disable microphone6-3, 6-6
 ID4-16
 IMMED4-17
 increase volume4-16, 4-18, 6-3, 6-5
 join
 monitor session 4-16, 4-17, 4-18, 6-3, 6-4
 session4-16, 4-20, 6-3, 6-5
 leave
 monitor session4-16, 4-18
 session4-16, 4-18, 6-3
 listenIpAdr4-17
 listenPort4-17
 monitoring4-16, 4-18, 6-3
 OVRRD4-17
 sample scenarios6-6
 ServiceType commands6-2
 guidelines6-2
 set
 DND6-3, 6-6
 monitored and monitoring
 rooms6-4
 privacy6-3, 6-6
 volume4-16, 4-19, 6-3, 6-5
 tag4-17, 4-18
 TalkIpAdr4-17
 talkPort4-17
 terminate
 monitor session6-3, 6-5
 INTERCOM JOIN 4-16, 4-20, 6-3, 6-5
 INTERCOM JOINMON4-16, 4-17, 4-18, 6-3, 6-4
 INTERCOM LEAVE ...4-16, 4-18, 6-3
 INTERCOM LEAVE, monitor4-16, 4-18

INTERCOM LEVEL_DN VOL4-16, 6-3
 INTERCOM LEVEL_SET VOL ..4-16, 4-19
 INTERCOM LEVEL_UP VOL4-16, 4-18
 INTERCOM MONITOR4-16, 6-3
 INTERCOM PTT4-19, 4-20
 INTERCOM PTT OFF 4-16, 4-19, 6-3
 IP
 address3-4, 3-5, 4-16, 4-17, 6-3
 port3-2, 4-17
 static address3-2

K

KEY x5-1, 5-2, 5-3, 5-5, 5-8, 5-12
 KEY x_PLUS5-6, 5-12
 Keyword3-4

L

LANG5-3, 5-11
 LEVEL_DN BALANCE4-1, 4-4
 LEVEL_DN BAND4-1, 4-6
 LEVEL_DN BASS4-1, 4-4
 LEVEL_DN BKLT6-1, 6-2
 LEVEL_DN TREB4-1, 4-5
 LEVEL_DN VOL4-1, 4-5
 LEVEL_SET BALANCE4-1, 4-3
 LEVEL_SET BAND4-1, 4-5
 LEVEL_SET BASS4-1, 4-4
 LEVEL_SET TREB4-1, 4-4
 LEVEL_SET VOL4-1, 4-5, 6-2
 LEVEL_UP BALANCE4-1, 4-4
 LEVEL_UP BAND4-1, 4-6
 LEVEL_UP BASS4-1, 4-4
 LEVEL_UP BKLT6-1, 6-2
 LEVEL_UP TREB4-1, 4-5
 LEVEL_UP VOL4-1, 4-5

M

Media
 listing4-15
MediaLinX
 commands5-1-5-20
 MENU_LIST m,n,{{path}} ...5-13, 5-14
 MENU_SEL {{path}} ..5-13, 5-18
 QUERY SOURCE5-13, 5-18
 SET AUDIO, ANALOG5-13

- SET AUDIO, GAIN5-13
- SET AUDIO, SPDIF5-13
- preset
 - listing source5-17
 - selecting5-18
- request media list5-14
- root serviceType commands ..5-19
- QUERY SERVICE5-19
- QUERY SERVICE,{{service name}}5-19
- RESTART5-19
- selecting a song from a menu
 - list5-18
 - traversing the menu list5-15
- MENU5-3, 5-10
- Menu4-11
 - list audio sources4-6
 - listing of renderers4-21, 4-23
 - requesting a list of media4-11
 - select from available sources ...4-8
 - selecting a specific item4-14
 - traversing4-12, 5-15
- MENU_LIST3-6
- MENU_LIST m,n,{{path}} ..4-11, 4-12, 5-13, 5-14
- MENU_LIST m,n,INTERCOM ...4-21, 6-9
- MENU_LIST m,n,RENDERERS 4-21, 4-23, 6-7, 6-9
- MENU_LIST m,n,SOURCES .4-1, 4-6
- MENU_RESP3-6
- MENU_SEL3-6
- MENU_SEL {{path}} ... 4-2, 4-8, 4-11, 4-14, 5-13, 5-18
- Messages
 - menu3-6
 - unsolicited3-2
- Microphone
 - activate4-16, 4-19
 - disable4-16, 4-19
- MODE5-3, 5-11
- monitorEnable4-16
- Multicast 1-1, 3-2, 3-3, 4-17
 - address3-1
- Mute
 - off4-1, 4-6
 - on4-1, 4-6
 - toggle4-1, 4-6
- MUTE OFF4-1
- MUTE ON4-1, 4-6
- MUTE TOGGLE4-1, 4-6
- N**
- Name
 - group3-3
 - room3-3
- NAV DN5-3
- NAV LT5-3
- NAV RT5-3
- NAV UP5-3
- NAV x5-12
- NEXT ...4-11, 4-14, 5-1, 5-2, 5-4, 5-6, 5-9
- P**
- PAUSE .4-11, 4-14, 5-1, 5-2, 5-4, 5-9
- PLAY 4-14, 5-1, 5-2, 5-4, 5-8
- Playlists4-12
- Preset
 - listing source5-17
 - selecting5-18
- PRESET DN5-2
- PRESET UP5-2
- PRESET x5-2, 5-7
- PREV ...4-11, 4-15, 5-1, 5-2, 5-4, 5-6, 5-9
- Privacy4-19, 6-3
 - set state4-16, 4-19
- PTT4-20
- Push to Talk - *see PTT*
- PWR OFF 5-1, 5-2, 5-4, 5-7, 5-8
- PWR ON 5-1, 5-2, 5-4, 5-7, 5-8
- Q**
- QUERY CURRENT_SOURCE4-2, 4-10
- QUERY RENDERER4-2, 4-9
- QUERY SERVICE 4-21, 5-19, 6-7, 7-2
- QUERY SERVICE, {{serviceName}} . 4-21, 5-19, 6-7, 7-2
- QUERY SOURCE ... 4-15, 5-13, 5-18
- QUERY UI6-1, 6-2
- R**
- RANDOM5-1, 5-5
- REPEAT TOGGLE5-1, 5-5
- Reports
 - parsing4-10
- RESTART ... 4-21, 5-20, 6-7, 7-2, 7-3
- RETURN5-3
- REV 5-1, 5-3, 5-5, 5-9
- Room3-1, 3-2
- Root2-2, 2-3, 2-4
- S**
- Sample Intercom Scenarios
 - complete control4-20
 - simple4-20
- SCAN DN5-2
- SCAN UP5-2
- SCAN x5-7
- SEEK DN5-2
- SEEK UP5-2
- SEEK x5-7
- SELECT5-3, 5-10
- Service Types
 - audio renderer 2-1, 2-2, 2-3, 2-4
 - audio source 2-1, 2-2, 2-3, 2-4
 - audio stream proxy2-1, 2-2, 2-3, 2-4
 - audio/local source2-1, 2-2, 2-3, 2-4
 - GPIO2-1, 2-2, 2-3
 - intercom 2-1, 2-2, 2-3, 2-4
 - root2-2, 2-3, 2-4
 - UI2-1, 2-2, 2-3
- serviceName4-1
- Services2-1, 3-1, 3-2
 - by device2-3
 - list configuration for all4-21
 - names2-1, 3-3
 - show configuration for
 - specific4-21
 - SpeakerLinX*2-3
 - types2-1
 - unique ID3-3
- serviceType4-1
- Sessions4-20
- SET AUDIO, ANALOG5-13
- SET AUDIO, GAIN5-13
- SET AUDIO, SPDIF5-13
- SET BKL, TIMEOUT6-1, 6-2
- SETUP5-3, 5-10
- SHUFFLE4-11, 4-15
- Software
 - third party3-1

Songs	MENU_LIST	Subnode
select	m,n,INTERCOM	~CURSRC
from menu list	MENU_LIST	~IRMOD
next	m,n,RENDERERS	~KEYPAD
previous	4-23	~ROOT
shuffling	QUERY SERVICE {{service-	~SERIAL_X
Source ServiceType	Name}}	~STATUS
commands	RESTART	~SUBSCRIBER
<i>SpeakerLinX</i>	QUERY SERVICE	address
commands	<i>SpeakerLinX</i> Renderer Service	Subscription
intercom serviceType	ACTIVE OFF	static
DND x	ACTIVE ON	SUBTITLE
INTERCOM JOIN	AMP OFF	T
INTERCOM JOINMON	AMP ON	TEN_PLUS
4-17	LEVEL_DN BALANCE	Third Party
INTERCOM LEAVE	LEVEL_DN BAND	controller
4-18	LEVEL_DN BASS	IP address
INTERCOM LEVEL_DN	LEVEL_DN TREB	software
VOL	LEVEL_DN VOL	toAddress
INTERCOM LEVEL_SET	LEVEL_SET BALANCE	TOPMENU
VOL	LEVEL_SET BAND	<i>TouchLinX</i>
INTERCOM LEVEL_UP VOL	LEVEL_SET BASS	commands
.....	LEVEL_SET TREB	ACTIVE OFF
INTERCOM MONITOR	LEVEL_SET VOL	ACTIVE ON
4-18	LEVEL_UP BALANCE	DND
INTERCOM PTT	LEVEL_UP BAND	INTERCOM JOIN
4-19	LEVEL_UP BASS	INTERCOM JOINMON
INTERCOM PTT OFF	LEVEL_UP TREB	INTERCOM LEAVE
4-19	LEVEL_UP VOL	INTERCOM LEVEL_DN
PRIVACY ON/OFF	levels	VOL
PRIVACY x	MENU_LIST m,n,SOURCES	INTERCOM LEVEL_SET
Setup Program	4-6	VOL
Guidelines	MENU_SEL {{path}}	INTERCOM LEVEL_UP
4-16	MUTE OFF	VOL
Media Server Audio Stream	MUTE ON	INTERCOM LEVEL_UP
commands	MUTE TOGGLE	VOL
MENU_LIST m,n,	QUERY CURRENT_SOURCE	INTERCOM MONITOR
{{path}}	INTERCOM PTT
MENU_SEL {{path}}	4-10	INTERCOM PTT OFF
4-14	QUERY RENDERER	LEVEL_DN BKLT
NEXT	4-2, 4-9	LEVEL_SET VOL
PAUSE	service type commands	LEVEL_UP BKLT
PLAY	SRC_SEL	PRIVACY
PREV	SRC_SEL {{source id}}	QUERY UI
QUERY SOURCE	SRC_SEL NEXT	SET BKLT TIMEOUT
4-15	SRC_SEL	UI/GUI
SHUFFLE	4-2, 4-9	root ServiceType commands
STOP	SRC_SEL {{source id}}	MENU_LIST m,n,INTERCOM
SL220	SRC_SEL NEXT	7,
SL250	STOP	MENU_LIST m,n,RENDERERS
<i>SpeakerLinX</i>	<i>StreamNet</i>	6-7,
commands	addressing	QUERY SERVICE
Root Type	ASCII command	QUERY SERVICE,
@Room 2#query	ASCII messages	{{serviceName}}
service.....	devices	RESTART
4-22	subscribing to	

- UI2-2
 - Treble
 - level4-1
 - level down4-1, 4-5
 - level set4-4
 - level up4-1, 4-5
 - TUNE DN5-2
 - TUNE UP5-2
 - TUNE x5-7
 - Tuner
 - commands5-2, 5-6
 - BAND NEXT5-2, 5-8
 - CLEAR5-2, 5-8
 - ENTER5-2
 - KEY x5-2, 5-8
 - NEXT5-2, 5-6
 - PRESET DN5-2
 - PRESET UP5-2
 - PRESET x5-2, 5-7
 - PREV5-2, 5-6
 - PWR OFF5-2, 5-7
 - PWR ON5-2, 5-7
 - SCAN5-7
 - SCAN DN5-2
 - SCAN UP5-2
 - SEEK DN5-2
 - SEEK UP5-2
 - SEEK x5-7
 - TUNE DN5-2
 - TUNE UP5-2
 - TUNE x5-7
 - TXSERIAL7-1
- U**
- UDP 1-1, 3-1, 3-2, 3-5
 - UI 2-1, 2-2, 2-3
 - UID
 - index3-3
 - serialNumber3-3
 - Unicast 1-1, 3-1, 4-17
 - UniqueID - *see UID*
 - User Datagram Protocol - *see UDP*
 - User Interface - *see UI*
- V**
- Volume
 - level4-1
 - level down4-1, 4-5
 - level set4-5
 - level up4-1, 4-5

Z

- ZOOM5-3, 5-11