Wheatstone Mixer
Automation Protocol

Last Updated: July 25, 2017

1 Revision History

In the CL version control system this document is named:
“$/wheatstone/D9/doc/mixer_automation.doc”

Date	Autho r	Description of Change
June 29, 2003	GAJ	• Created
July 1, 2003	GAJ	• Added PHONE LINK parameter. Added INPUT and PHONE master assigns. Added SYS EVENT & TIME parameter.

July 7, 2003	GAJ	• Added INPUT EQ, INPUT DYN, PHONE EQ, PHONE DYN, MASTER EQ & MASTER DYN. Added GROUP master assigns. Added INPUT BMIN. Added INPUT MM# and PHONE MM#.
July 15, 2003	GAJ	• Added MM FADER, AUX FADER & STUDIO FADER.
July 18, 2003	GAJ	• Added DISPLAY TEXT, DISPLAY COLOR.
March 12, 2004	GAJ	• Protocol Rev 1.0 • Updated to cover G5, G4 & G3
April 14, 2004	GAJ	• Removed SYS-EVENT command. Added SYS-TAKEPROG, SYS-SAVEPROG & SYS-DELPROG commands. Updated surface matrix for INPUT-MM# command.
November 30, 2004	GAJ	• Protocol Rev 1.1 • Added minimum protocol revision information to each message definition. • Added input and spare button subscription commands and responses. • Added spare button LED control commands. • Added section on TCP timeouts.
May 10, 2005	GAJ	• Protocol Rev 1.2 • Changed console's code to allow multiple (10) simultaneous TCP connections. • Added <SYS|CLRSUBS:1> and <SYS?INPUTS> commands.
June 12, 2005	GAJ	• Added G6 & SR9 mixers to the compatibility tables.
August 14, 2007	TJI	• Protocol Rev 1.4 • Added INPUT SRC and SRCPRST commands (for D9 family)
August 16, 2007	BKL	• Removed supported surface listings from each message type and created a spreadsheet which contains all surfaces and message parameters.
April 17, 2009	TJI	• Protocol Rev 1.5 • Incorporate D52, D8, D16 and new D51/D52 features • Add INPUT:SRC command to E5/E6
May 30, 2012	JN	• Protocol Rev 1.6 • Added INCHANSUBSCRIBE ON command
June 19, 2013	SJG	• Protocol Rev 1.7 • Added STUDIOSUBSCRIBE FADER command
September 30, 2014	SJG	• Went back and filled in missing info on INPUTSUBSCRIBE:FADER • Consolidated E-Series to one table since they support the same commands
July 25, 2017	SJG	• Clarified information about protocol version and software version.
May 31, 2018	JN	• STUDIO FADER command supports EMX and L8
July 19, 2018	JN	• Updating appendix B with EMX and L8/L12 information

2 Introduction

The mixer automation control protocol for a variety of Wheatstone TV and Radio surfaces is described in this document. These surfaces all use a common automation control protocol over an Ethernet TCP/IP connection. Note: while the protocol is common, the range of commands supported varies widely from surface model to model. Consult the table of supported commands at the end of this document and cross-reference it with the model to know which surface model supports which command. **You cannot assume that because, say an E6 surface supports protocol version X.Y, and that has command Z, that all surfaces running protocol version X.Y support command Z.** The "protocol version" really only applies within a particular surface model. If in doubt, it is always best to simply query the surface with a query command, and if it is not supported it will reply with `<NAK|Unsupported Request>`.

3 TCP Connection

Each mixer supports up to ten simultaneous TCP connections from client PCs (or other network devices). The mixer acts as a TCP server for the connection. The PC acts as a TCP client for the connection. The mixer listens for TCP connections on port 55777. All remote computers will make a TCP connection to the surfaces with this one TCP port number.

3.1 TCP Connection Timeout

The mixer console will timeout and close a TCP connection if it does not have any activity for more than 60 seconds. Therefore, in order to maintain your TCP connection you want to periodically send a heartbeat message of some sort. You could periodically request the version number of the console, which would result in a response being sent to your application.

Starting with version 1.1 of the automation protocol you can send an empty message, if you want to periodically send heartbeats to keep the connection alive without generating a response from the console. An empty message would be of the form:

```
<>
```

You could also send an empty message in earlier versions of the protocol to keep the connection alive, but you will receive a NAK response for each empty message.

Each of the up to 10 connections are independent, each requires its own heartbeat activity.

4 Message Format

All automation control messages are made up of upper case ASCII character strings. The general format of a message is start character (**<**), message target, query or command character (**?** Or **|**), optional parameters, end character (**>**).

4.1 Message Delimiters

Each message is delimited by a less then (**<**) character at the beginning and a greater than (**>**) character at the end. Any characters following the greater than character at the end of one message and preceding the next less than character starting the next message are ignored. Therefore carriage returns and line feeds and zero terminators between messages will be silently ignored.

Example:

```
<SYS?>
<INPUT:4?FADER>
<INPUT:4|ON:1,FADER:192>
```

The mixer will respond to all properly formatted messages that are sent to it with an answer, an acknowledgement, or an error, depending on the message that was sent.

4.2 Message Target

The message target follows the start character. The target identifies the target of the command. A generic target supported by all mixers is the **SYS** target. The **SYS** target may be queried to find the mixer type and revision, it may also be queried to find the version of this protocol that the mixer supports.

Example:

```
<SYS?>
<SYS?MODEL>
<SYS?VERSION>
```

Other targets are **INPUT** channels, **PHONE** channels, **GROUP** channels, **MASTER** channels, etc.
For numbered targets such as **INPUT** channels the message target also contains the channel number.

Example:

```
<INPUT:1?>
<INPUT:2?>
<INPUT:3?>
<INPUT:1|FADER:192>
<INPUT:2|FADER:192>
<INPUT:3|FADER:128>
<PHONE:1|ON:1>
<GROUP:1|ON:1>
<MASTER:1|ON:1>
<DCM:1|ON:1>
```

4.3 Message Type

The message type follows the message target. The message type may be either a query or a command. Query messages are identified by a question mark character (**?**) and command messages are identified by a pipe character (**|**).

4.4 Optional Parameters

Parameters follow the message type character. When the message contains multiple parameters, the parameters are delimited by comma characters. When a parameter has a value associated, the value will be delimited from the parameter name with a colon character.

Parameters are optional for query messages. For example if you send a **SYS** query with no parameters, you will receive a response string with all of the system parameters and their values in the response.

Example:

```
Query:    <SYS?>
Response: <SYS|MODEL:G9,VERSION:0.32,AUTO:1.0,TIME:12.05.00>
```

As an alternative you can specify a system parameter on a **SYS** query and the mixer will respond with just the value of that single parameter.

Example:

```
Query:    <SYS?MODEL>
Response: <SYS|MODEL:G9>
Query:    <SYS?VERSION>
Response: <SYS|VERSION:0.32>
```

Commands must have at least one parameter but may optionally have multiple parameters. For example you could move an input fader to 0dB and turn that input on with two commands each with a single parameter:

Example:

```
Command:  <INPUT:1|FADER:192>
Response: <OK>
Command:  <INPUT:1|ON:1>
Response: <OK>
```

Or you could move an input fader to 0dB and turn that input on with one command with two parameters:

Example:

```
Command:  <INPUT:1|FADER:192,ON:1>
Response: <OK>
```

5 Reply Format

Reply messages from the mixer follow the same format as command messages. The mixer will insert a carriage (Hex 0D) and line feed character (Hex 0A) after every reply. Since these characters are outside the message delimiters, the characters should be ignored by a message parser. The characters are included to make string parsing easier for some computer languages.

As you've seen from the examples in the previous section the mixer will respond to query messages with a string that is formatted in the same manner as a command string. The mixer will respond to commands with an acknowledgement string "**<OK>**".

If a query or command message is not properly formatted with a start and end character the mixer will ignore the message. If the message is properly formatted, but contains instructions that are out of bounds or not supported by the particular type of mixer an error message will be returned.

Error String	Description
`<NAK|Invalid Message Format>`	The message format was invalid.
`<NAK|Unsupported Request>`	An unsupported command was specified.
`<NAK|Invalid Channel>`	The target channel number is invalid.
`<NAK|Invalid Parameter ID>`	The parameter is invalid.
`<NAK|Invalid Parameter Value>`	The parameter value is invalid.
`<NAK|Not All Commands Processed>`	Some of the command parameters may have been processed but some had errors. This error will only occur when multiple parameters are specified within a command.

6 Message Parameter Reference

This section breaks down the available message targets and their parameters. Each message target and parameter has a description of the parameter value and a listing of which mixers support the target and parameter.

6.1 SYS Parameters

System message targets are used to query and set system parameters.

6.1.1 SYS MODEL

Target:	SYS
Parameter:	MODEL
Channel:	N/A
Value:	D51, D52, D7, D8, S4, S2, D9, D10, D12, D16, E4, E5, E6, LX24, G3, G4, G5, G6, G7, G9, SR9, D1, D3
Query:	Yes
Command:	No
Protocol Rev:	1.0

The SYS MODEL parameter may be queried by the automation application on the PC. The reply parameter value will specify the type of mixer surface (e.g. "D51", "D9" or "G9"). If you want to write a generic automation application, do not hard-code a specific model string, or when new surface models are released you will not be able to talk to them without rebuilding your software to support it.

Example:

```
Query:    <SYS?MODEL>
Response: <SYS|MODEL:G9>
```

6.1.2 SYS VERSION

Target:	SYS
Parameter:	VERSION
Channel:	N/A
Value:	[The software version string]
Query:	Yes
Command:	No
Protocol Rev:	1.0

The SYS VERSION parameter may be queried by the automation application on the PC. The reply parameter value will specify the mixer software revision string. This may be a number like 1.0, or almost any other type of version identifier. It is meant only as a piece of information to be displayed to the user or in logs, etc. There is no guarantee that it will fit to any specific format, just treat it as an opaque string.

Example:

```
Query:    <SYS?VERSION>
Response: <SYS|VERSION:1.0>
```

6.1.3 SYS AUTO

Target:	SYS
Parameter:	AUTO
Channel:	N/A
Value:	Major.Minor
Query:	Yes
Command:	No
Protocol Rev:	1.0

The SYS AUTO parameter may be queried by the automation application on the PC. The reply parameter value will specify the version of the automation controller protocol supported.

Example:

```
Query:    <SYS?AUTO>
Response: <SYS|AUTO:1.0>
```

The version number is a two part number delimited by periods.

The first number is the major revision number. This number will only change when a part of the message formats have changed. If you have written an application according to a major revision that does not match the reported major revision of the surface, you will encounter incompatibility problems in some messages.

The second number is the minor revision number. This number will only change when a message is not backward compatible. The minor revision number will typically increment when a new message target and/or parameter is added to the protocol. If you have written an application according to a minor revision that is greater than the reported minor revision of the surface, you may encounter incompatibility problems in some messages. If you have written an application according to a minor revision that is equal to or less then the minor revision of the surface then you will not encounter any compatibility problems.

NOTE: As stated in the introduction section, *this version is only valid within a particular surface model*. An LX-24 surface running version 1.5 of the automation protocol will not necessarily support the same commands as a D3 running version 1.5. This is inherent in the protocol design, because each surface has its own unique set of buses and features that can be controlled and queried via Automation.

See section 1 “Revision History” to find the version of the protocol defined in this document.

6.1.4 SYS TAKEPROG

Target:	SYS
Parameter:	TAKEPROG
Channel:	N/A
Value:	Event Preset Name
Query:	No
Command:	Yes
Protocol Rev:	1.0

The SYS TAKEPROG parameter may be set by the automation application on the PC. The parameter value is the name of a pre-existing event preset. If an event preset with the specified name is found the event preset will be applied to the mixer. If no matching event preset name is found an error will be returned and the mixer settings will remain unchanged.

On the G4 mixer there is no facility to name event presets. Therefore on the G4 mixer the value parameter specifies the program number (1 – 4).

Example:

```
Command:  <SYS|TAKEPROG:PROG#001>
Response: <OK>
```

6.1.5 SYS SAVEPROG

Target:	SYS
Parameter:	SAVEPROG
Channel:	N/A
Value:	Event Preset Name
Query:	No
Command:	Yes
Protocol Rev:	1.0

The SYS SAVEPROG parameter may be set by the automation application on the PC. The parameter value is the name of a pre-existing event preset. If an event preset with the specified name is found the mixers current settings are stored in the specified event preset. If no matching event preset name is found an error will be returned and the mixer settings will remain unchanged.

On the G4 mixer there is no facility to name event presets. Therefore on the G4 mixer the value parameter specifies the program number (1 – 4).

Example:

```
Command:  <SYS|SAVEPROG:PROG#001>
Response: <OK>
```

6.1.6 SYS DELPROG

Target:	SYS
Parameter:	DELPROG
Channel:	N/A
Value:	Event Preset Name
Query:	No
Command:	Yes
Protocol Rev:	1.0

The SYS DELPROG parameter may be set by the automation application on the PC. The parameter value is the name of a pre-existing event preset. If an event preset with the specified name is found the event preset is deleted. If no matching event preset name is found an error will be returned and the mixer settings will remain unchanged.

Example:

```
Command:  <SYS|SAVEPROG:PROG#001>
Response: <OK>
```

6.1.7 SYS TIME

Target:	SYS
Parameter:	TIME
Channel:	N/A
Value:	HH.MM.SS
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The SYS TIME parameter may be queried or set by the automation application on the PC. The parameter correlates to the mixers clock time. The parameter value is an eight character string. Two characters for the hour (0 – 23), two characters for the minute (0 – 59) and two characters for the second (0 – 59).

Example:

```
Query:    <SYS?TIME>
Response: <SYS|TIME:12.05.00>

Command:  <SYS|TIME:13.05.00>
Response: <OK>
```

6.1.8 SYS INPUTS

Target:	SYS
Parameter:	INPUTS
Channel:	N/A
Value:	Number of available input channels.
Query:	Yes
Command:	No
Protocol Rev:	1.2

The SYS INPUTS parameter may be queried by the automation application on the PC. The reply parameter value will specify the number of input channels on the console.

Example:

```
Query:    <SYS?INPUTS>
Response: <SYS|INPUTS:16>
```

6.1.9 SYS CLRSUBS

Target:	SYS
Parameter:	CLRSUBS
Channel:	N/A

Value:	1
Query:	No
Command:	Yes
Protocol Rev:	1.2

The SYS CLRSUBS parameter may be set to one to by the automation application on the PC to clear ALL subscriptions on the connection.

Example:

```
Query:    <SYS?CLRSUBS:1>
Response: <OK>
```

6.2 DISPLAY Parameters

Display message targets are used to query or set the custom text message section at the bottom of the LCD displays on the D9 & G9 mixers. When an DISPLAY target is specified a channel number must be a part of the message target. The channel number can be from 1 to 12. Channel 1 indicates the right most LCD panel on the mixer, channel 2 indicates the next panel to the left, etc.

6.2.1 DISPLAY TEXT

Target:	DISPLAY:#
Parameter:	TEXT
Channel:	1 – 12
Value:	0 to 63 character text message
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The DISPLAY TEXT parameter may be queried or set by the automation application on the PC. The parameter value indicates the text displayed in the custom text section at the bottom of the LCD panel. The text can contain any ASCII printable characters except the comma (**,**), less than (**<**) and greater than (**>**) characters. If an empty string is sent, then the default custom text will be displayed. Therefore to remove all text from the custom text section you should send a string with a single space character. The default text is an empty string. This text message is not preserved through a power cycle and is not stored with an event preset.

Example:

```
Set the text to “HELLO WORLD !!!”…
Command:  <DISPLAY:1|TEXT:HELLO WORLD !!!>
Response: <OK>

Query:    <DISPLAY:1?TEXT>
Response: <DISPLAY:1|TEXT:HELLO WORLD !!!>

Restore the default text…
Command:  <DISPLAY:1|TEXT:>
Response: <OK>

Erase the text… (Note one space character before greater than character)
Command:  <DISPLAY:1|TEXT: >
Response: <OK>
```

6.2.2 DISPLAY COLOR

Target:	DISPLAY:#
Parameter:	COLOR
Channel:	1 – 12
Value:	0 to 6
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The DISPLAY COLOR parameter may be queried or set by the automation application on the PC. The parameter value indicates the color of the custom text section at the bottom of the specified LCD panel. The color is represented by an integer value as shown in the following table. The default color is 0. This text color is not preserved through a power cycle and is not stored with an event preset.

Color Value	Text Color	Background Color	Example Text
0	WHITE	GRAY BLUE	EXAMPLE TEXT
1	BLACK	RED	EXAMPLE TEXT
2	WHITE	DARK RED	EXAMPLE TEXT
3	BLACK	YELLOW	EXAMPLE TEXT
4	BLACK	GREEN	EXAMPLE TEXT
5	WHITE	DARK GREEN	EXAMPLE TEXT
6	WHITE	BLUE	EXAMPLE TEXT
7	BLACK	LIGHT GRAY	EXAMPLE TEXT

6.3 INPUT Parameters

Input message targets are used to query or set input channel parameters. When an input target is specified a channel number must be a part of the message target. The channel number can be from 1 up to the number of input channels defined on the mixer.

6.3.1 INPUT ON

Target:	INPUT:#
Parameter:	ON
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ON/OFF button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <INPUT:1?ON>
Response: <INPUT:1|ON:1>

Turn the channel off…
Command:  <INPUT:1|ON:0>
Response: <OK>

Turn the channel on…
Command:  <INPUT:1|ON:1>
Response: <OK>
```

6.3.2 INPUT FADER

Target:	INPUT:#
Parameter:	FADER
Channel:	1 - number of input channels
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A1 for a description of the equations to translate the fader position to fader gain.

Example:

```
Query:    <INPUT:1?FADER>
Response: <INPUT:1|FADER:2056>

Set the fader to 0dB…
Command:  <INPUT:1|FADER:192>
Response: <OK>
```

6.3.3 INPUT PFL

Target:	INPUT:#
Parameter:	PFL
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT PFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the PFL button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

Example:

```
Query:    <INPUT:1?PFL>
Response: <INPUT:1|PFL:0>

Take the channel off the PFL bus…
Command:  <INPUT:1|PFL:0>
Response: <OK>

Put the channel on the PFL bus…
Command:  <INPUT:1|PFL:1>
Response: <OK>
```

6.3.4 INPUT AFL

Target:	INPUT:#
Parameter:	AFL
Channel:	1 - number of input channels
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT AFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the AFL button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

Example:

```
Query:    <INPUT:1?AFL>
Response: <INPUT:1|AFL:0>

Take the channel off the AFL bus…
Command:  <INPUT:1|AFL:0>
Response: <OK>

Put the channel on the AFL bus…
Command:  <INPUT:1|AFL:1>
Response: <OK>
```

6.3.5 INPUT CUE

Target:	INPUT:#
Parameter:	CUE
Channel:	1 - number of input channels
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT CUE parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the CUE button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

Example:

```
Query:    <INPUT:1?CUE>
Response: <INPUT:1|CUE:0>

Take the channel off the CUE bus…
Command:  <INPUT:1|CUE:0>
Response: <OK>
```

```
Put the channel on the CUE bus…
Command:  <INPUT:1|CUE:1>
Response: <OK>
```

6.3.6 INPUT 5.1

Target:	INPUT:#
Parameter:	5.1
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT 5.1 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the 5.1 master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.7 INPUT ST1

Target:	INPUT:#
Parameter:	ST1
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT ST1 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ST1 master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.8 INPUT ST2

Target:	INPUT:#
Parameter:	ST2
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT ST2 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ST2 master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.9 INPUT ST3

Target:	INPUT:#
Parameter:	ST3
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT ST3 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ST3 master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.10 INPUT MONO

Target:	INPUT:#
Parameter:	MONO
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0 (D51/D52 only)

The INPUT MONO parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the MONO master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.11 INPUT PGM

Target:	INPUT:#
Parameter:	PGM
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT PGM parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the PGM master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.12 INPUT AUD

Target:	INPUT:#
Parameter:	AUD
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT AUD parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the AUD master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.13 INPUT AUX

Target:	INPUT:#
Parameter:	AUX
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT AUX parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the AUX master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.14 INPUT UTL

Target:	INPUT:#
Parameter:	UTL
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT UTL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the UTL master bus assign button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.3.15 INPUT PGMx

Target:	INPUT:#
Parameter:	PGMA | PGMB | PGMC | PGMD
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.4

The INPUT PGMx parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the PGMx master bus assign button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.3.16 INPUT EQ

Target:	INPUT:#
Parameter:	EQ
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT EQ parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the EQ button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.3.17 INPUT DYN

Target:	INPUT:#
Parameter:	DYN
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT DYN parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the DYN button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.3.18 INPUT CMP

Target:	INPUT:#
Parameter:	CMP
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.2

The INPUT CMP parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the Compressor “IN” button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.3.19 INPUT EXP

Target:	INPUT:#
Parameter:	DYN
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.2

The INPUT EXP parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the Expander or Gate "IN" button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.20 INPUT BMIN

Target:	INPUT:#
Parameter:	BMIN
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT BMIN parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the bus minus IN button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.3.21 INPUT MM#

Target:	INPUT:#
Parameter:	MM#
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The INPUT MM# parameters may be queried or set by the automation application on the PC. The parameter values indicate the state of the mix minus # assign buttons. A value of '0' indicates OFF, a value of '1' indicates ON.

The D9, G9, G5 & G4 mixers each have 8 mix minus buses. Therefore the parameter may be one of "MM1", "MM2", "MM3", "MM4", "MM5", "MM6", "MM7" or "MM8" to access each of the respective eight mix minus bus assigns. The G3 mixer each has 4 mix minus buses. Therefore the parameter may be one of "MM1", "MM2", "MM3" or "MM4" to access each of the respective four mix minus bus assigns.

6.3.22 INPUT SRC

Target:	INPUT:#
Parameter:	SRC
Channel:	1 - number of input channels
Value:	0 – number of signals
Query:	Yes
Command:	Yes
Protocol Rev:	1.4

The INPUT SRC parameter may be queried or set by the automation application on the PC. The parameter value indicates the signal number to take on this channel. The ACI command works by virtually pressing the TAKE button, so all the logic and rules for taking sources apply (locks, etc). Taking source ID 0 disconnects the input channel.

Example:

```
Query:    <INPUT:1?SRC>
Response: <INPUT:1|SRC:1002>

Take signal ID 47 on input 1
Command:  <INPUT:1|SRC:47>
Response: <OK>
```

6.3.23 INPUT SRCPRST

Target:	INPUT:#
Parameter:	SRCPRST
Channel:	1 - number of input channels
Value:	1
Query:	Yes
Command:	Yes
Protocol Rev:	1.4

The INPUT SRCPRST parameter may be queried or set by the automation application on the PC. The parameter value is ignored, but should be preset for syntactic reasons. Sending this command causes the surface to act as if the user pressed the PRESET key on the specified input channel. You cannot set the preset signal using this command, you can only TAKE the already defined preset signal. This command is used on surfaces capable of 1 preset per input channel.

Example:

```
Query:    <INPUT:1?SRCPRST>
Response: <INPUT:1|SRCPRST:1002>

Take the preset signal on input 1
Command:  <INPUT:1|SRCPRST:1>
Response: <OK>
```

6.3.24 INPUT SRCPRA

Target:	INPUT:#
Parameter:	SRCPRA
Channel:	1 - number of input channels
Value:	1
Query:	Yes
Command:	Yes
Protocol Rev:	1.5

The INPUT SRCPRA parameter may be queried or set by the automation application on the PC. The parameter value is ignored, but should be preset for syntactic reasons. Sending this command causes the surface to act as if the user pressed the PRESET A key on the specified input channel. You cannot set the preset signal using this command, you can only TAKE the already defined preset signal. This command is used on surfaces capable of 2 presets per input channel.

Example:

```
Query:    <INPUT:1?SRCPRA>
Response: <INPUT:1|SRCPRA:1002>

Take the “preset A” signal on input 1
Command:  <INPUT:1|SRCPRA:1>
Response: <OK>
```

6.3.25 INPUT SRCPRB

Target:	INPUT:#
Parameter:	SRCPRB
Channel:	1 - number of input channels
Value:	1
Query:	Yes
Command:	Yes
Protocol Rev:	1.5

The INPUT SRCPRB parameter may be queried or set by the automation application on the PC. The parameter value is ignored, but should be preset for syntactic reasons. Sending this command causes the surface to act as if the user pressed the PRESET B key on the specified input channel. You cannot set the preset signal using this command, you can only TAKE the already defined preset signal. This command is used on surfaces capable of 2 presets per input channel.

Example:

```
Query:    <INPUT:1?SRCPRB>
Response: <INPUT:1|SRCPRB:1002>

Take the “preset B” signal on input 1
Command:  <INPUT:1|SRCPRB:1>
Response: <OK>
```

6.4 PHONE Parameters

Phone message targets are used to query or set phone channel parameters. When an phone target is specified a channel number must be a part of the message target. The channel number can be from 1 up to the number of phone channels defined on the mixer.

6.4.1 PHONE ON

Target:	PHONE:#
Parameter:	ON
Channel:	1 - number of phone channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ON/OFF button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

Example:

```
Query:    <PHONE:1?ON>
Response: <PHONE:1|ON:1>

Turn the channel off…
Command:  <PHONE:1|ON:0>
Response: <OK>

Turn the channel on…
Command:  <PHONE:1|ON:1>
Response: <OK>
```

6.4.2 PHONE FADER

Target:	PHONE:#
Parameter:	FADER
Channel:	1 - number of phone channels
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A1 for a description of the equations to translate the fader position to fader gain.

Example:

```
Query:    <PHONE:1?FADER>
Response: <PHONE:1|FADER:256>

Set the fader to 0dB…
Command:  <PHONE:1|FADER:192>
```

```
Response: <OK>
```

6.4.3 PHONE CUE

Target:	PHONE:#
Parameter:	CUE
Channel:	1 - number of phone channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE CUE parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the CUE button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <PHONE:1?CUE>
Response: <PHONE:1|CUE:0>

Take the channel off the CUE bus…
Command:  <PHONE:1|CUE:0>
Response: <OK>

Put the channel on the CUE bus…
Command:  <PHONE:1|CUE:1>
Response: <OK>
```

6.4.4 PHONE LINK

Target:	PHONE:#
Parameter:	LINK
Channel:	1 - number of phone channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE LINK parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the LINK button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <PHONE:1?LINK>
Response: <PHONE:1|LINK:0>

Take the channel off the LINK bus…
Command:  <PHONE:1|LINK:0>
Response: <OK>

Put the channel on the LINK bus…
Command:  <PHONE:1|LINK:1>
Response: <OK>
```

6.4.5 PHONE PGM

Target:	PHONE:#
Parameter:	PGM
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE PGM parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the PGM master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.4.6 PHONE AUD

Target:	PHONE:#
Parameter:	AUD
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE AUD parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the AUD master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.4.7 PHONE AUX

Target:	PHONE:#
Parameter:	AUX
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE AUX parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the AUX master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.4.8 PHONE UTL

Target:	PHONE:#
Parameter:	UTL
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE UTL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the UTL master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.4.9 PHONE EQ

Target:	PHONE:#
Parameter:	EQ
Channel:	1 - number of phone channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE EQ parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the EQ button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.4.10 PHONE DYN

Target:	PHONE:#
Parameter:	DYN
Channel:	1 - number of phone channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE DYN parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the DYN button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.4.11 PHONE MM#

Target:	PHONE:#
Parameter:	MM#
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The PHONE MM# parameters may be queried or set by the automation application on the PC. The parameter values indicate the state of the mix minus # assign buttons. A value of '0' indicates OFF, a value of '1' indicates ON.

The D9 mixer has 8 mix minus buses. Therefore the parameter may be one of "MM1", "MM2", "MM3", "MM4", "MM5", "MM6", "MM7" or "MM8" to access each of the respective eight mix minus bus assigns.

6.5 GROUP Parameters

Group message targets are used to query or set group channel parameters. When a group target is specified a channel number must be a part of the message target. The channel number can be from 1 up to the number of group channels defined on the mixer.

6.5.1 GROUP ON

Target:	GROUP:#
Parameter:	ON
Channel:	1 – 8
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GROUP ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ON/OFF button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <GROUP:1?ON>
Response: <GROUP:1|ON:1>

Turn the channel off…
Command:  <GROUP:1|ON:0>
Response: <OK>

Turn the channel on…
Command:  <GROUP:1|ON:1>
Response: <OK>
```

6.5.2 GROUP FADER

Target:	GROUP:#
Parameter:	FADER
Channel:	1 – 8
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GROUP FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A1 for a description of the equations to translate the fader position to fader gain.

Example:

```
Query:    <GROUP:1?FADER>
Response: <GROUP:1|FADER:256>

Set the fader to 0dB…
Command:  <GROUP:1|FADER:192>
Response: <OK>
```

6.5.3 GROUP PFL

Target:	GROUP:#
Parameter:	PFL
Channel:	1 – 8
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GROUP PFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the PFL button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <GROUP:1?PFL>
Response: <GROUP:1|PFL:0>

Take the channel off the PFL bus…
Command:  <GROUP:1|PFL:0>
Response: <OK>

Put the channel on the PFL bus…
Command:  <GROUP:1|PFL:1>
Response: <OK>
```

6.5.4 GROUP AFL

Target:	GROUP:#
Parameter:	AFL
Channel:	1 – 8
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GROUP AFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the AFL button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <GROUP:1?AFL>
Response: <GROUP:1|AFL:0>

Take the channel off the AFL bus…
Command:  <GROUP:1|AFL:0>
Response: <OK>

Put the channel on the AFL bus…
Command:  <GROUP:1|AFL:1>
Response: <OK>
```

6.5.5 GROUP 5.1

Target:	GROUP:#
Parameter:	5.1
Channel:	1 – 8
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GROUP 5.1 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the 5.1 master bus assign button. A value of '0' indicates OFF, a value of '1' indicates ON.

6.5.6 GROUP ST1

Target:	GROUP:#	D51:	Yes
Parameter:	ST1	D9:	Yes
Channel:	1 – 8	SR9:	No
Value:	0 – 1	G9:	No
Query:	Yes	G6:	No
Command:	Yes	G5:	No
Protocol Rev:	1.0	G4:	No
		G3:	No

The GROUP ST1 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ST1 master bus assign button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.5.7 GROUP ST2

Target:	GROUP:#
Parameter:	ST2
Channel:	1 – 8
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GROUP ST2 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ST2 master bus assign button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.5.8 GROUP ST3

Target:	GROUP:#
Parameter:	ST3
Channel:	1 – 8
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GROUP ST3 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ST3 master bus assign button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.5.9 GROUP MONO

Target:	GROUP:#
Parameter:	MONO
Channel:	1 – 8
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0 (D51/D52 only)

The GROUP MONO parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the MONO master bus assign button. A value of ‘0’ indicates OFF, a value of ‘1’ indicates ON.

6.6 MASTER Parameters

Master message targets are used to query or set master channel parameters. When a master target is specified a channel number must be a part of the message target. The channel number can be from 1 up to the number of master channels defined on the mixer.

6.6.1 MASTER ON

Target:	MASTER:#
Parameter:	ON
Channel:	1 – 5 (D51) 1 – 4 (D9)
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The MASTER ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ON/OFF button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <MASTER:1?ON>
Response: <MASTER:1|ON:1>

Turn the channel off…
Command:  <MASTER:1|ON:0>
Response: <OK>

Turn the channel on…
Command:  <MASTER:1|ON:1>
Response: <OK>
```

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

Channel Number	D51 / D52	D9, D10, D12, D16	D7	D8	Ex	Gx	SR9
1	5.1	5.1	PGM	ST1	PGMA	PGM	5.1
2	+2	ST1	AUD	ST2	PGMB	AUD	PGM
3	ST1	ST2	--	--	PGMC	AUX	AUD
4	ST2	ST3	--	--	PGMD	UTL	UTL
5	MONO	---	---	---	--	---	--

Table 1: Master Channel Numbers

6.6.2 MASTER FADER

Target:	MASTER:#
Parameter:	FADER
Channel:	1 – 5 (D51) 1 – 4 (D9)
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The MASTER FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A1 for a description of the equations to translate the fader position to fader gain.

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

Example:

```
Query:    <MASTER:1?FADER>
Response: <MASTER:1|FADER:256>

Set the fader to 0dB…
Command:  <MASTER:1|FADER:192>
Response: <OK>
```

6.6.3 MASTER PFL

Target:	MASTER:#
Parameter:	PFL
Channel:	1 – 5 (D51) 1 – 4 (D9)
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The MASTER PFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the PFL button. A value of '0' indicates OFF, a value of '1' indicates ON.

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

Example:

```
Query:    <MASTER:1?PFL>
Response: <MASTER:1|PFL:0>

Take the channel off the PFL bus…
Command:  <MASTER:1|PFL:0>
Response: <OK>

Put the channel on the PFL bus…
Command:  <MASTER:1|PFL:1>
Response: <OK>
```

6.6.4 MASTER AFL

Target:	MASTER:#
Parameter:	AFL
Channel:	1 – 5 (D51) 1 – 4 (D9)
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The MASTER AFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the AFL button. A value of '0' indicates OFF, a value of '1' indicates ON.

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

Example:

```
Query:    <MASTER:1?AFL>
Response: <MASTER:1|AFL:0>

Take the channel off the AFL bus…
Command:  <MASTER:1|AFL:0>
Response: <OK>

Put the channel on the AFL bus…
Command:  <MASTER:1|AFL:1>
Response: <OK>
```

6.6.5 MASTER EQ

Target:	MASTER:#
Parameter:	EQ
Channel:	1 – 4
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The MASTER EQ parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the EQ button. A value of '0' indicates OFF, a value of '1' indicates ON.

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

6.6.6 MASTER DYN

Target:	MASTER:#
Parameter:	DYN
Channel:	1 – 4
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The MASTER DYN parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the DYN button. A value of '0' indicates OFF, a value of '1' indicates ON.

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

6.6.7 MASTER CMP

Target:	MASTER:#
Parameter:	CMP
Channel:	1 – 4
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.5

The MASTER CMP parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the Compressor "IN" button. A value of '0' indicates OFF, a value of '1' indicates ON.

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

6.6.8 MASTER EXP

Target:	MASTER:#
Parameter:	EXP
Channel:	1 – 4
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.5

The MASTER EXP parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the Expander or Gate "IN" button. A value of '0' indicates OFF, a value of '1' indicates ON.

The channel indicates the master bus as shown in Table 1: Master Channel Numbers.

6.7 GRMSTR Parameters

GRMSTR message targets are used to query or set grand master channel parameters. When a GRMSTR target is specified a channel number must be a part of the message target. The channel number must be 1.

6.7.1 GRMSTR ON

Target:	GRMSTR:#
Parameter:	ON
Channel:	1
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GRMSTR ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ON/OFF button. A value of '0' indicates OFF, a value of '1' indicates ON.

Example:

```
Query:    <GRMSTR:1?ON>
Response: <GRMSTR:1|ON:1>

Turn the grand master off…
Command:  <GRMSTR:1|ON:0>
Response: <OK>

Turn the grand master on…
Command:  <GRMSTR:1|ON:1>
Response: <OK>
```

6.7.2 GRMSTR FADER

Target:	GRMSTR:#
Parameter:	FADER
Channel:	1
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The GRMSTR FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A2 for a description of the equations to translate the fader position to fader gain.

Example:

```
Query:    <GRMSTR:1?FADER>
Response: <GRMSTR:1|FADER:256>

Set the fader to 0dB…
Command:  <GRMSTR:1|FADER:192>
Response: <OK>
```

6.8 DCM Parameters

DCM message targets are used to query or set DCM channel parameters. When a DCM target is specified a channel number must be a part of the message target. The channel number can be from 1 to 4.

6.8.1 DCM ON

Target:	DCM:#
Parameter:	ON
Channel:	1 – 4
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The DCM ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the ON/OFF (MUTE) button. A value of ‘0’ indicates the channel is OFF (muted), a value of ‘1’ indicates the channel is ON (not muted).

Example:

```
Query:    <DCM:1?ON>
Response: <DCM:1|ON:1>

Turn the DCM off…
Command:  <DCM:1|ON:0>
Response: <OK>

Turn the DCM on…
Command:  <DCM:1|ON:1>
Response: <OK>
```

6.8.2 DCM FADER

Target:	DCM:#
Parameter:	FADER
Channel:	1 – 4
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The DCM FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A2 for a description of the equations to translate the fader position to fader gain.

6.9 MM Parameters

MM message targets are used to query or set mix minus channel parameters. When an MM target is specified a channel number must be a part of the message target.

6.9.1 MM FADER

Target:	MM:#
Parameter:	FADER
Channel:	1 – n (n = # of Mix Minus busses on surface [model-specific]).
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The MM FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A1 for a description of the equations to translate the fader position to fader gain.

6.10 AUX Parameters

AUX message targets are used to query or set aux master channel parameters. When an AUX target is specified a channel number must be a part of the message target.

6.10.1 AUX FADER

Target:	AUX:#
Parameter:	FADER
Channel:	1 – n (n = # of Aux busses on surface [model-specific]).
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The AUX FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A1 for a description of the equations to translate the fader position to fader gain.

6.11 STUDIO Parameters

STUDIO message targets are used to query or set studio monitor channel parameters. When a STUDIO target is specified a channel number must be a part of the message target.

6.11.1 STUDIO FADER

Target:	STUDIO:#
Parameter:	FADER
Channel:	1 – n (see table)
Value:	0 – 256
Query:	Yes
Command:	Yes
Protocol Rev:	1.0

The STUDIO FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the position of the fader. The value can range from 0 to 256. See Appendix A1 for a description of the equations to translate the fader position to fader gain.

The channel indicates the studio as shown in Table 2: Studio Channel Numbers. Note that some mixers in a given family may not have all the studio output buses listed. The channel number assignments do not change within a family.

Channel	D51 / D52	D12, D16	D7, D9, D10, SR9	D8	Ex	Gx	EMX, DMX, L8
1	Studio 1	Studio 1	Studio 1	Studio 1	Studio 1	Studio 1	Studio 1
2	Studio 2	Studio 2	Studio 2	Studio 2	Studio 2	Studio 2	--
3	Studio 3	Studio 3	Headphone	--	Headphone	Headphone	Headphone
4	Studio 4	Control Room 1	Control Room	Control Room	Control Room	Control Room	Control Room
5	Studio 5	Control Room 2	---	---	--	---	
6	Production	---	---	---	--	---	
7	Headphone	---	---	---	--	---	
8	Control Room	---	---	---	--	---	

Table 2: Studio Channel Numbers

6.12 SPARE Parameters

System message targets are used to query and set spare button parameters.

6.12.1 SPARE LED#

Target:	SPARE
Parameter:	LED#
Channel:	N/A
Value:	0-1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The SPARE LED# parameters may be queried or set by the automation application on the PC. The parameter values indicate the state of the spare # button LED. A value of '0' indicates OFF, a value of '1' indicates ON.

The D9, G9 & G5 mixers each have 12 spare buttons, iff they are equipped with the optional surface module. Therefore the parameter may be one of "LED1", "LED2", "LED3" … "LED12" to access each of the respective 12 spare button LEDs. The G3 mixer has 4 spare buttons. Therefore the parameter may be one of "LED1", "LED2", "LED3" or "LED4" to access each of the respective four spare button LEDs.

Example:

```
Query:    <SPARE?>
Response: <SPARE|LED1:0,LED2:0,LED3:0,LED4:0,LED5:0,LED6:0,LED7:0,LED8:0,
           LED9:0,LED10:0,LED11:0,LED12:0>
Command:  <SPARE|LED11:1,LED12:1>
Response: <OK>
Query:    <SPARE?LED11>
Response: <SPARE|LED11:1>
Query:    <SPARE?>
```

```
Response: <SPARE|LED1:0,LED2:0,LED3:0,LED4:0,LED5:0,LED6:0,LED7:0,LED8:0,
           LED9:0,LED10:0,LED11:1,LED12:1>
```

NOTE: The SPARE LED command works with internal flag states within the mixer console. You may experience a situation where your application sends varying the LED state messages, the responses are okay and queries to the console indicate that the flag values are changing… **but** you do not see the LED on the spare button change state. This most likely indicates a problem with the spare button configuration file on the console flash drive. The configuration file (^{eg} G9_OPTS.TXT) must be setup such that the spare button you are trying to access is configured for use by the automation interface. The file may be accessed via FTP to the console, and instructions regarding the option file parameters are contained within. The following example shows the spare button section from a G5 options file. In this options file spare buttons 1 through 4 are not accessible through the automation API, spare buttons 5 through 12 are available to the automation API.

```
// --------------------------------
// SPARE OPTIONS
// --------------------------------
// Syntax: SPARE#:?
// # is the spare button number (1 - 12)
// ? is the mode (default = 2)...
//   0 = None, Button presses are ignored, LED controlled by DIO
//   1 = Toggle, Button state toggles on each press, LED controlled by Surface
//   2 = Momentary, Button is active when held down, LED controlled by Surface
//   3 = Momentary, Button is active when held down, LED controlled by DIO
//   4 = Automation, Button & LED are controlled by automation interface
SPARE1:0
SPARE2:1
SPARE3:2
SPARE4:3
SPARE5:4
SPARE6:4
SPARE7:4
SPARE8:4
SPARE9:4
SPARE10:4
SPARE11:4
SPARE12:4
```

6.13 Event Subscription Messages

Your application can subscribe to certain asynchronous events which may occur do to operator activity at the console surface. This section describes how your application can subscribe to events related to input channels and the spare buttons.

To receive notification of asynchronous events your application must register a subscription to the event with the console. Once your application has subscribed to one or more events, your application will receive asynchronous messages over the TCP socket. The format of these event messages is the same as any other message type. Since they are asynchronous, you may receive event notifications when you don't expect them. For example your application might send a command to the surface and expect to see an <OK> acknowledgement. There is always a slim chance that at that very instance the operator presses a spare button, you may receive the spare button event before you receive the <OK> for a command that you just sent. So please think asynchronously in your message handling when you use subscriptions. You will not ever receive nested messages (i.e. `<SPARE_E<OK>VENT|BTN9:0>` will not occur).

When you first establish a TCP connection to the mixer console there will be no subscriptions. Therefore if your application disconnects and reconnects to the console, you must re-subscribe to any events you want.

Each of the up to 10 connections are independent, each has it's own unique subscription list.

NOTE: Subscription events will only report human operator activity. If a second automation application, or a preset is played back on the surface you will not receive notifications. If you need to know about non-human interaction you must periodically poll the surface or use a different subscription method described in section 6.14.

6.13.1 INPUTSUBSCRIBE

The INPUTSUBSCRIBE message targets allow you to subscribe to events on the input channels related to the ON/OFF buttons, the CUE (PFL) buttons, the master assign buttons and the mix minus assign buttons.

6.13.1.1 INPUTSUBSCRIBE ON

Target:	INPUTSUBSCRIBE:#
Parameter:	ON
Channel:	1 - number of input channels
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to ON/OFF events on a particular input channel. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?ON>
Response: <INPUTSUBSCRIBE:1|ON:0>
Command:  <INPUTSUBSCRIBE:1|ON:1>
Response: <OK>
Event:    <INPUT_EVENT:1|ON:1>    -- User presses the ON button
Event:    <INPUT_EVENT:1|ON:1>    -- User presses the ON button again
Event:    <INPUT_EVENT:1|ON:0>    -- User presses the OFF button
Event:    <INPUT_EVENT:1|ON:0>    -- User presses the OFF button again
```

Note: On the G series consoles, which have separate ON & OFF buttons, you will receive an ON event if the ON button is pressed even if the channel is already ON. Like wise with the OFF button, you will receive OFF events even if the channel is already off. The D9 console have ON buttons which toggle the channel ON/OFF state, so on this console you will not receive repeated messages.

6.13.1.2 INPUTSUBSCRIBE CUE

Target:	INPUTSUBSCRIBE:#
Parameter:	CUE
Channel:	1 - number of input channels
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE CUE parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to CUE events on a particular input channel. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?CUE>
Response: <INPUTSUBSCRIBE:1|CUE:0>
Command:  <INPUTSUBSCRIBE:1|CUE:1>
Response: <OK>
Event:    <INPUT_EVENT:1|CUE:1>   -- User presses the CUE button
Event:    <INPUT_EVENT:1|CUE:0>   -- User presses the CUE button again
```

6.13.1.3 INPUTSUBSCRIBE PFL

Target:	INPUTSUBSCRIBE:#
Parameter:	PFL
Channel:	1 - number of input channels
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE PFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to PFL events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?PFL>
Response: <INPUTSUBSCRIBE:1|PFL:0>
Command:  <INPUTSUBSCRIBE:1|PFL:1>
Response: <OK>
Event:    <INPUT_EVENT:1|PFL:1>   -- User presses the PFL button
Event:    <INPUT_EVENT:1|PFL:0>   -- User presses the PFL button again
```

6.13.1.4 INPUTSUBSCRIBE AFL

Target:	INPUTSUBSCRIBE:#
Parameter:	AFL
Channel:	1 - number of input channels
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE AFL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to AFL events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?PFL>
Response: <INPUTSUBSCRIBE:1|PFL:0>
Command:  <INPUTSUBSCRIBE:1|PFL:1>
Response: <OK>
Event:    <INPUT_EVENT:1|PFL:1>   -- User presses the PFL button
Event:    <INPUT_EVENT:1|PFL:0>   -- User presses the PFL button again
```

6.13.1.5 INPUTSUBSCRIBE 5.1

Target:	INPUTSUBSCRIBE:#
Parameter:	5.1
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE 5.1 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to 5.1 assign events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?5.1>
Response: <INPUTSUBSCRIBE:1|5.1:0>
Command:  <INPUTSUBSCRIBE:1|5.1:1>
Response: <OK>
Event:    <INPUT_EVENT:1|5.1:1>   -- User presses the 5.1 assign button
Event:    <INPUT_EVENT:1|5.1:0>   -- User presses the 5.1 assign again
```

6.13.1.6 INPUTSUBSCRIBE ST1

Target:	INPUTSUBSCRIBE:#
Parameter:	ST1
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE ST1 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to ST1 assign events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?ST1>
Response: <INPUTSUBSCRIBE:1|ST1:0>
Command:  <INPUTSUBSCRIBE:1|ST1:1>
Response: <OK>
Event:    <INPUT_EVENT:1|ST1:1>   -- User presses the ST1 assign button
Event:    <INPUT_EVENT:1|ST1:0>   -- User presses the ST1 assign again
```

6.13.1.7 INPUTSUBSCRIBE ST2

Target:	INPUTSUBSCRIBE:#
Parameter:	ST2
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE ST2 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to ST2 assign events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example: See INPUTSUBSCRIBE ST1

6.13.1.8 INPUTSUBSCRIBE ST3

Target:	INPUTSUBSCRIBE:#
Parameter:	ST3
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE ST3 parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to ST3 assign events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example: See INPUTSUBSCRIBE ST1

6.13.1.9 INPUTSUBSCRIBE PGM

Target:	INPUTSUBSCRIBE:#
Parameter:	PGM
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE PGM parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to PGM assign events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?PGM>
Response: <INPUTSUBSCRIBE:1|PGM:0>
Command:  <INPUTSUBSCRIBE:1|PGM:1>
Response: <OK>
Event:    <INPUT_EVENT:1|PGM:1>   -- User presses the PGM assign button
Event:    <INPUT_EVENT:1|PGM:0>   -- User presses the PGM assign again
```

6.13.1.10 INPUTSUBSCRIBE AUD

Target:	INPUTSUBSCRIBE:#
Parameter:	AUD
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE AUD parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to AUD assign events on a particular input channel. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

Example: See INPUTSUBSCRIBE PGM

6.13.1.11 INPUTSUBSCRIBE AUX

Target:	INPUTSUBSCRIBE:#
Parameter:	AUX
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE AUX parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to AUX master bus assign events on a particular input channel. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

Example: See INPUTSUBSCRIBE PGM

6.13.1.12 INPUTSUBSCRIBE UTL

Target:	INPUTSUBSCRIBE:#
Parameter:	UTL
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE UTL parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to UTL assign events on a particular input channel. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

Example: See INPUTSUBSCRIBE PGM

6.13.1.13 INPUTSUBSCRIBE MM#

Target:	INPUTSUBSCRIBE:#
Parameter:	MM#
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The INPUTSUBSCRIBE MM# parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to MM# assign events on a particular input channel. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

The D9, G9, G5 & G4 mixers each have 8 mix minus buses. Therefore the parameter may be one of “MM1”, “MM2”, “MM3”, “MM4”, “MM5”, “MM6”, “MM7” or “MM8” to access each of the respective eight mix minus bus assigns. The G3 mixer each has 4 mix minus buses. Therefore the parameter may be one of “MM1”, “MM2”, “MM3” or “MM4” to access each of the respective four mix minus bus assigns.

Example:

```
Query:    <INPUTSUBSCRIBE:1?MM1>
Response: <INPUTSUBSCRIBE:1|MM1:0>
Command:  <INPUTSUBSCRIBE:1|MM1:1>
Response: <OK>
Event:    <INPUT_EVENT:1|MM1:1>   -- User presses the MM1 assign button
Event:    <INPUT_EVENT:1|MM1:0>   -- User presses the MM1 assign again
```

6.13.1.14 INPUTSUBSCRIBE FADER

Target:	INPUTSUBSCRIBE:#
Parameter:	Fader #
Channel:	1 - number of input channels
Value:	0 – 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.5

The INPUTSUBSCRIBE FADER# parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to fader level events on a particular input channel. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

Example:

```
Query:    <INPUTSUBSCRIBE:1?FADER>
Response: <INPUTSUBSCRIBE:1|FADER:0>
Command:  <INPUTSUBSCRIBE:1|FADER:1>
Response: <OK>
Event:    <INPUT_EVENT:1|FADER:0>   -- User pulls fader 1 all the way down
Event:    <INPUT_EVENT:1|FADER:255>   -- User pushes fader 1 all the way up
```

6.13.2 SPARESUBSCRIBE

The SPARESUBSCRIBE message targets allow you to subscribe to events related to the spare buttons.
Note: Please see the note in the SPARE LED# section regarding spare button configuration.

6.13.2.1 SPARESUBSCRIBE BTN#

Target:	SPARESUBSCRIBE
Parameter:	BTN#
Channel:	N/A
Value:	0-1
Query:	Yes
Command:	Yes
Protocol Rev:	1.1

The SPARESUBSCRIBE BTN# parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to BTN# events on a particular spare button. A value of ‘0’ indicates no subscription, a value of ‘1’ indicates a subscription.

The D9, G9 & G5 mixers each have 12 spare buttons, iff they are equipped with the optional surface module. Therefore the parameter may be one of “BTN1”, “BTN2”, “BTN3” … “BTN12” to access each of the respective 12 spare button LEDs. The G3 mixer has 4 spare buttons. Therefore the parameter may be one of “BTN1”, “BTN2”, “BTN3” or “BTN4” to access each of the respective four spare button LEDs.

Example:

```
Query:    <SPARESUBSCRIBE?BTN12>
Response: <SPARESUBSCRIBE|BTN12:0>
Command:  <SPARESUBSCRIBE|BTN12:1>
Response: <OK>
Event:    <SPARE_EVENT|BTN12:1>   -- User presses spare button 12
Event:    <SPARE_EVENT|BTN12:0>   -- User releases spare button 12
```

6.14 Event Subscription Messages Extended

Your application can subscribe to certain asynchronous events which may occur do to operator activity at the console surface or activity from an automation application. This section describes how your application can subscribe to events related to input channels.

To receive notification of asynchronous events your application must register a subscription to the event with the console. Once your application has subscribed to one or more events, your application will receive asynchronous messages over the TCP socket. The format of these event messages is the same as any other message type. Since they are asynchronous, you may receive event notifications when you don't expect them. For example your application might send a command to the surface and expect to see an <OK> acknowledgement. There is always a slim chance that at that very instance the state of an input button changes, you may receive the inchan event before you receive the <OK> for a command that you just sent. So please think asynchronously in your message handling when you use subscriptions. You will not ever receive nested messages (i.e. `<INCHAN_E<OK>VENT:4|ON:0>` will not occur).

When you first establish a TCP connection to the mixer console there will be no subscriptions. Therefore if your application disconnects and reconnects to the console, you must re-subscribe to any events you want.

Each of the up to 10 connections are independent, each has it's own unique subscription list.

6.14.1 INCHANSUBSCRIBE ON

Target:	INCHANSUBSCRIBE:#
Parameter:	ON
Channel:	1 - number of input channels
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.6

The INCHANSUBSCRIBE ON parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of the state your subscription to ON/OFF events on a particular input channel. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example:

```
Query:    <INCHANSUBSCRIBE:1?ON>
Response: <INCHANSUBSCRIBE:1|ON:0>
Command:  <INCHANSUBSCRIBE:1|ON:1>
Response: <OK>
Event:    <INCHAN_EVENT:1|ON:1>    -- Button state changed to ON
Event:    <INCHAN_EVENT:1|ON:1>    -- User pressed the ON button
Event:    <INCHAN_EVENT:1|ON:0>    -- Button state changed to OFF
Event:    <INCHAN_EVENT:1|ON:0>    -- User pressed the OFF button
```

Note: On the G series consoles, which have separate ON & OFF buttons, you will receive an ON event if the ON button is pressed (by a human operator) even if the channel is already ON. Likewise with the OFF button, you will receive OFF events even if the channel is already off. The D9 console have ON buttons which toggle the channel ON/OFF state, so on this console you will not receive repeated messages. When an automation application sets the button ON/OFF state only state changes are received.

6.14.2 STUDIOSUBSCRIBE FADER

Target:	STUDIOSUBSCRIBE:#
Parameter:	FADER
Channel:	1 – number of studio buses
Value:	0 - 1
Query:	Yes
Command:	Yes
Protocol Rev:	1.7

The STUDIOSUBSCRIBE FADER parameter may be queried or set by the automation application on the PC. The parameter value indicates the state of your subscription to fader level events on a particular studio bus. A value of '0' indicates no subscription, a value of '1' indicates a subscription.

Example:

```
Query:    <STUDIOSUBSCRIBE:1?FADER>
Response: <STUDIOSUBSCRIBE:1|FADER:0>
Command:  <STUDIOSUBSCRIBE:1|FADER:1>
Response: <OK>
Event:    <STUDIO_EVENT:1|FADER:0>    -- First studio fader turned all the way down
Event:    <STUDIO_EVENT:1|FADER:256>    -- First studio fader turned all the way up
```

6.15 Clip Player Messages

The clip player present in some newer Wheatstone control surfaces can be controlled through ACI. You can query and change the current track and playlist, get the current state and timestamp, as well as stop/play/pause playback.

6.15.1 TRACK

Target:	CLIPPLAYER
Parameter:	TRACK
Value:	Positive or Negative integer on command, string on query
Query:	Yes
Command:	Yes
Protocol Rev:	1.4

The CLIPPLAYER TRACK query can be used to query the current track name as a string. When used as a command, the parameter should be a negative or positive integer (usually -1 or 1), which is the amount by which to go forward or backwards in the track list.

6.15.2 PLAYLIST

Target:	CLIPPLAYER
Parameter:	PLAYLIST
Value:	Positive or Negative integer on command, string on query
Query:	Yes
Command:	Yes
Protocol Rev:	1.4

The CLIPPLAYER PLAYLIST query can be used to query the current playlist name as a string. When used as a command, the parameter should be a negative or positive integer (usually -1 or 1), which is the amount by which to go forward or backwards in the playlist list.

6.15.3 AUTOADVANCE

Target:	CLIPPLAYER
Parameter:	AUTOADVANCE
Value:	1 for on, 0 for off
Query:	Yes
Command:	Yes
Protocol Rev:	1.4

The CLIPPLAYER AUTOADVANCE query can be used to query or change the current state of the Auto Advance feature. A value of 1 means Auto Advance is on, while a value of 0 means it is off.

6.15.4 PLAY

Target:	CLIPPLAYER
Parameter:	PLAY
Value:	N/A
Query:	No
Command:	Yes
Protocol Rev:	1.4

The CLIPPLAYER PLAY command can be used to start playback of the current track.

6.15.5 PAUSE

Target:	CLIPPLAYER
Parameter:	PAUSE
Value:	N/A
Query:	No
Command:	Yes
Protocol Rev:	1.4

The CLIPPLAYER PAUSE command can be used to pause playback of the current track.

6.15.6 CUE

Target:	CLIPPLAYER
Parameter:	CUE
Value:	N/A
Query:	No
Command:	Yes
Protocol Rev:	1.4

The CLIPPLAYER CUE command puts the current track back into cue (stopped at the beginning) if it is playing or paused. If the track is already cued, this has no effect.

6.15.7 STATE

Target:	CLIPPLAYER
Parameter:	STATE
Value:	N/A
Query:	Yes
Command:	No
Protocol Rev:	1.4

The CLIPPLAYER STATE query gets the current state of the clip player. The response is a string indicating the state, and can be one of the following: "STATE:INACTIVE" if inactive (no clip selected), "STATE:CUED" if a clip is selected but stopped at the beginning, "STATE:PLAYING" if a clip is playing, or "STATE:PAUSED" if a clip is paused (stopped somewhere other than the beginning of the track).

6.15.8 TIME

Target:	CLIPPLAYER
Parameter:	TIME
Value:	N/A
Query:	Yes
Command:	No
Protocol Rev:	1.4

The CLIPPLAYER TIME query reports the current timestamp of the clip player as a string. If a clip is not playing or paused, this will always be 00:00. Otherwise, it will be the time in minutes and seconds into the track, in the format mm:ss.

APPENDIX A Fader Curves

This section describes the equations that relate fader position to fader gain in dB.

A1 –60dB to +12dB Faders

Faders that range from –60dB up to +12dB follow the following equations:

Fader Position	Fader Gain In dB
64 – 256	12 – ((256 – N) * 12/64)
1 – 63	12 – (36 + ((64 – N) * 36/64))
0	-99

The curve is linear in dB with a break point at the ¼ point in the scale. The very bottom position in the scale is -99dB. The bottom ¼ of the scale ranges from –60dB to –24dB. The top ¾ of the scale ranges from –24dB to +12dB. This results in the following gains:

Position	Gain	Position	Gain	Position	Gain	Position	Gain	Position	Gain
256	+12dB	192	0dB	128	-12dB	64	-24dB	1	-59dB
240	+9dB	176	-3dB	112	-15dB	48	-33dB	0	-99dB
224	+6dB	160	-6dB	96	-18dB	32	-42dB		
208	+3dB	144	-9dB	80	-21dB	16	-51dB		

A2 –72dB to 0dB Faders

Faders that range from –72dB up to 0dB follow the following equations:

Fader Position	Fader Gain In dB
64 – 256	((256 – N) * 12/64)
1 – 63	(36 + ((64 – N) * 36/64))
0	-99

The curve is linear in dB with a break point at the ¼ point in the scale. The very bottom position in the scale is -99dB. The bottom ¼ of the scale ranges from –72dB to –36dB. The top ¾ of the scale ranges from –36dB to 0dB. This results in the following gains:

Position	Gain	Position	Gain	Position	Gain	Position	Gain	Position	Gain
256	0dB	192	-12dB	128	-24dB	64	-36dB	1	-72dB
240	-3dB	176	-15dB	112	-27dB	48	-45dB	0	-99dB
224	-6dB	160	-18dB	96	-30dB	32	-54dB		
208	-9dB	144	-21dB	80	-33dB	16	-63dB		

APPENDIX B API Cross Reference

This section details which commands are available on which surface models.

Message Parameters	Protocol Rev.	DTV / D5.1	D52	D7	D8/S4/S2	D9	D10	D12	Ex / LX24	G3	G4	G5	G6	G7	G9	SR9	Dim1	D3	EMX	L8/L12
AUX FADER	1.0	X	X			X	X	X							X	X	X	X		
DCM FADER	1.0	X	X			X	X	X									X	X		
DCM ON	1.0							X												
DISPLAY COLOR	1.0	X	X			X	X	X				X			X	X	X	X		
DISPLAY TEXT	1.0	X	X			X	X	X				X			X	X	X	X		
GRMSTR FADER	1.0	X	X																	
GRMSTR ON	1.0	X	X																	
GROUP 5.1	1.0	X	X			X	X	X									X	X		
GROUP AFL	1.0	X				X	X	X												
GROUP FADER	1.0	X	X		X	X	X	X									X	X		
GROUP ON	1.0	X	X		X	X	X	X									X	X		
GROUP PFL	1.0	X	X			X	X										X	X		
GROUP ST1	1.0	X	X			X	X	X									X	X		
GROUP ST2	1.0	X	X			X	X	X									X	X		
GROUP ST3	1.0	X	X			X	X	X									X	X		
INPUT 5.1	1.0	X	X			X	X	X												
INPUT SRND1	1.4																X	X		
INPUT SRND2	1.4																X	X		
INPUT AFL	1.0	X				X	X	X										X		
INPUT AGC	1.2			X	X									X						
INPUT AUD	1.0			X					X	X	X	X	X	X	X	X			X	X
INPUT AUX	1.0			X					X	X	X	X	X	X	X	X			X	X

Message Parameters	Protocol Rev.	DTV / D5.1	D52	D7	D8/S4/S2	D9	D10	D12	Ex / LX24	G3	G4	G5	G6	G7	G9	SR9	Dim1	D3	EMX	L8/L12
INPUT PGMA	1.4								X										X	X
INPUT PGMB	1.4								X										X	X
INPUT PGMC	1.4								X										X	X
INPUT PGMD	1.4								X										X	X
INPUT BMIN	1.0	X	X			X	X	X							X	X	X	X		
INPUT CMP	1.2		X	X	X									X						
INPUT CMPC	1.2			X	X									X						
INPUT CMPM	1.2			X	X									X						
INPUT CMPO	1.2			X	X									X						
INPUT CMPR	1.2			X	X									X						
INPUT CMPT	1.2			X	X									X						
INPUT CUE	1.0			X	X				X	X	X	X	X	X	X	X			X	X
INPUT DPEQ	1.2			X					X					X						
INPUT DYN*	1.0	X				X	X	X				X			X	X	X	X		
INPUT EQ**	1.0	X	X			X	X	X				X			X	X	X	X		
INPUT EXP	1.2		X	X	X				X					X			X	X		
INPUT EXPC	1.2			X	X				X					X						
INPUT EXPD	1.2			X	X				X					X						
INPUT EXPH	1.2			X	X				X					X						
INPUT EXPO	1.2			X	X				X					X						
INPUT EXPR	1.2			X	X				X					X						
INPUT EXPT	1.2			X	X				X					X						
INPUT FADER	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
INPUT HPF	1.2			X	X				X					X						
INPUT HPFF	1.2			X	X				X					X						
INPUT LPF	1.2			X	X				X					X						
INPUT LPFF	1.2			X	X				X					X						
INPUT MONO	1.2	X	X																	

Message Parameters	Protocol Rev.	DTV / D5.1	D52	D7	D8/S4/S2	D9	D10	D12	Ex / LX24	G3	G4	G5	G6	G7	G9	SR9	Dim1	D3	EMX	L8/L12
INPUT MM#	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
INPUT ON	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
INPUT PEQ	1.2			X	X				X					X						
INPUT PEQF#	1.2			X	X				X					X						
INPUT PEQL#	1.2			X	X				X					X						
INPUT PEQW#	1.2			X	X				X					X						
INPUT PFL	1.0	X	X			X	X	X	X								X	X		
INPUT PGM	1.0			X	X				X	X	X	X	X	X	X	X			X	X
INPUT SEQ#	1.2			X	X				X					X						
INPUT SEQF#	1.2			X	X				X					X						
INPUT SEQL#	1.2			X	X				X					X						
INPUT SRC	1.4	X	X			X	X		X			X			X	X	X	X	X	X
INPUT SRCPRST	1.4					X	X					X			X	X	X	X		
INPUT SRCPRA	1.5	X	X																	
INPUT SRCPRB	1.5	X	X																	
INPUT ST1	1.0	X	X			X	X	X									X	X		
INPUT ST2	1.0	X	X			X	X	X									X	X		
INPUT ST3	1.0					X	X	X												
INPUT UTL	1.0								X	X	X	X	X		X	X			X	X
INPUTSUBSCRIBE 5.1	1.1	X	X			X	X	X												
INPUTSUBSCRIBE SRND1	1.4																X	X		
INPUTSUBSCRIBE SRND2	1.4																X	X		
INPUTSUBSCRIBE MONO	1.5	X	X																	
INPUTSUBSCRIBE AFL	1.1					X	X										X	X		
INPUTSUBSCRIBE AUD	1.1			X					X	X	X	X	X	X	X				X	X
INPUTSUBSCRIBE AUX	1.1			X					X	X	X	X	X	X	X				X	X
INPUTSUBSCRIBE CUE	1.1			X	X				X	X	X	X	X	X	X	X			X	X
INPUTSUBSCRIBE MM#	1.1	X	X	X	X			X	X	X	X	X	X	X	X	X			X	

Message Parameters	Protocol Rev.	DTV / D5.1	D52	D7	D8/S4/S2	D9	D10	D12	Ex / LX24	G3	G4	G5	G6	G7	G9	SR9	Dim1	D3	EMX	L8/L12
INPUTSUBSCRIBE ON	1.1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
INPUTSUBSCRIBE PFL	1.1	X	X			X	X	X	X								X	X		
INPUTSUBSCRIBE PGM	1.1			X					X	X	X	X	X	X	X				X	X
INPUTSUBSCRIBE ST1	1.1	X	X			X	X	X									X	X		
INPUTSUBSCRIBE ST2	1.1	X	X			X	X	X									X	X		
INPUTSUBSCRIBE ST3	1.1					X	X	X												
INPUTSUBSCRIBE UTL	1.1			X					X	X	X	X	X	X	X				X	X
INPUTSUBSCRIBE FADER	1.5					X	X		X								X	X	X	X
INCHANSUBSCRIBE ON	1.6								X										X	X
MASTER AFL	1.0	X				X	X													
MASTER CMP	1.5		X																	
MASTER DYN*	1.0	X				X	X	X									X	X		
MASTER EQ**	1.0	X	X			X	X	X									X	X		
MASTER EXP	1.5		X																	
MASTER FADER	1.0	X	X		X	X	X	X									X	X		
MASTER ON	1.0	X	X		X	X	X	X									X	X		
MASTER PFL	1.0	X	X			X	X										X	X		
MM FADER	1.0	X	X			X	X	X									X	X		
PHONE AUD	1.0														X					
PHONE AUX	1.0														X					
PHONE CUE	1.0														X					
PHONE DYN*	1.0														X					
PHONE EQ**	1.0														X					
PHONE FADER	1.0														X					
PHONE LINK	1.0														X					
PHONE MM#	1.0														X					
PHONE ON	1.0														X					
PHONE PGM	1.0														X					

Message Parameters	Protocol Rev.	DTV / D5.1	D52	D7	D8/S4/S2	D9	D10	D12	Ex / LX24	G3	G4	G5	G6	G7	G9	SR9	Dim1	D3	EMX	L8/L12
PHONE UTL	1.0														X					
SPARE LED#	1.1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
SPARESUBSCRIBE BTN#	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
STUDIO FADER	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SYS AUTO	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SYS CLRSUBS	1.2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SYS DELPROG	1.0	X	X	X	X	X	X	X			X			X	X	X	X	X		
SYS INPUTS	1.2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SYS MODEL	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SYS SAVEPROG	1.0	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X		
SYS SYNCPROG	1.2	X	X	X	X			X	X					X						
SYS TAKEPROG	1.0	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X
SYS TIME	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
SYS VERSION	1.0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CLIPPLAYER TRACK	1.4																	X		
CLIPPLAYER PLAYLIST	1.4																	X		
CLIPPLAYER AUTOADVANCE	1.4																	X		
CLIPPLAYER PLAY	1.4																	X		
CLIPPLAYER PAUSE	1.4																	X		
CLIPPLAYER CUE	1.4																	X		
CLIPPLAYER STATE	1.4																	X		
CLIPPLAYER TIME	1.4																	X		