

JSD-80 Replacement and Automation Command Guide

The JSD-80 is an older sound processor that interfaces to both digital servers and film sound readers with an RS232 control port. The newer generation of products still maintain the RS232 interface with an enhance command set. The new command format requires tab characters to separate command arguments with the same carriage return termination character (0x0D). So basically the command updates are simple and easy to use with regards to the JSD-80, JSD-60, and JSD-100.

JSD-80:

The following is an example of the JSD-80 commands that are typically used for control by a server over RS232 or Ethernet. Note that the JSD-80 echoes the command and parameters. The parameters are space (0x20) or tab (0x09) delimited. The command line is terminated by a carriage return (0x0d). Responses are tab delimited.

A typical exchange is shown below. These are the responses from the JSD-80.

```
00>button
5
00>button 6
```

```
00>button
6
00>
```

Setting the level is simple the value range is from 0 to 64. Now to retrieve the level the ? parameter is used unlike the button command where no parameter is specified.

```
00>level 63
00>level ?
63
00>level
63
00>
```

Sets main level control to value specified in hex (00 to 64, corresponding to 00 to 100 decimal and 0.0 to 10.0 on the front panel display). Updates front panel display and sends appropriate commands to DSPs. If level is followed by a space and then a question mark, it returns the current level as a hex number (0 to 64 corresponding to 00 to 100 decimal and 0.0 to 10.0 on the front panel display). Similarly, if the level command is provided with no parameter, the current level is returned.

JSD-100:

These commands are used to query the status of the JSD100 or change its operation (adjust level, select input, etc.). The commands are typically sent by an automation system. They may be sent over RS232 or Ethernet. This is a subset of the full command set. The full command set includes commands for adjust equalization, etc. **Note that every command generates some sort of response that must be picked up by the command sender. Failure to pick up these responses can cause buffers in the JSD-100 or the host to fill, possibly halting communications. If the response is not needed, a simple "eat garbage" routine that loops until the receive buffer is empty should be called before sending a command. With RS232 the response need not to be read and is only necessary when there's a need to process the unit setting, like doing a relative fader adjustment. As mention the Ethernet read operation is required when sending a command, some media servicers accomplish this by keeping the connection alive for some defined period (a second or two) and then closing it.**

JSD-80 Replacement and Automation Command Guide

Commands are sent in ASCII. Commands and their parameters are case sensitive. Responses are in lower case. Commands and their parameters are separated by the ASCII tab character (0x09). The last parameter (if any) is followed by a carriage return (0x0d). Note that the command is not echoed. Only the response(s) followed by a carriage return are transmitted by the JSD-100. The JSD-100 executes the command and returns the new state of the system as an ASCII value followed by a carriage return. If no parameters are supplied, the JSD-100 does not change the operation of the system but still returns the "new" state of the system. Note that the carriage return character following the command and response is not shown in the table below. Numeric parameters and responses are signed decimal. In many cases, they have an assumed decimal point position two digits from the right. The assumed decimal point description is provided in the descriptions in the table below. The formatting of commands and responses in the table below may break the command or response in to more than one line. The actual command or response is a single line of text followed by a carriage return.

In the following example a tab character is used between parameters such that the command string programmatically appears as "jsd100.sys.input_mode\t1":

```
jsd100.sys.input_mode 1
```

Note that the input parameter is 1 through 10, with 1 representing COAX 1

The fader value is multiplied by 100 such that a value of 7.1 would be represented as 710:

```
Jsd100.sys.fader 710
```

JSD-60:

These commands are modeled on the jsd100.sys command set with a variation regarding the input selection. On the JSD-100 the first format button is assign 1 and on the JSD-60 it's a 0. The JSD-60 supports common JSD-100 commands such that both products will process the jsd100.sys.input_mode, jsd100.sys.fader, and other JSD-100 commands. With regards to the interface processing on the JSD-60, they are same characteristics as described for the JSD-100.

Note the input mode selection for the JSD-60 is zero based.

```
Jsd60.sys.input_mode 0
```

```
Jsd60.sys.fader 710
```