

# Generic RS232 Control Commands for iFlex Mixers

iFlex DSP audio mixers provide for third party controllers using the RS232 port or dongle. Parameters for the protocol are:

19200 baud, 8 bit, no parity. Full duplex on RX and TX. No hardware handshaking is required

There are four possible command types that can be sent to an iFlex unit via RS232:

1. Switch state changes
2. Volume changes
3. Queries
4. Feature settings.

All commands begin with 'CMD>' and are terminated with a carriage return, line feed or both. The single character hex values are 0x0d for a carriage return and 0x0a for a line feed. Imbedded spaces or commas between tokens are ignored, but may be added for readability. When responses are returned a single space will be placed before new delimiters for aid in parsing and/or readability.

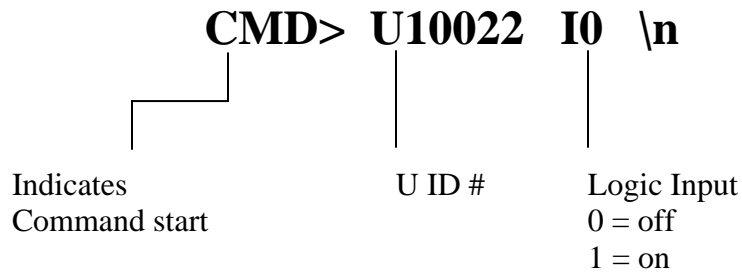
A Sonata/iFlex control has three elements: A switch input, a switch output, and a level input. A control is identified by a unique identifier which is an integer number. This id must precede the control tokens.

**Note** - Switches that are to be controlled by third party controllers must be defined as type "Static" in the Sonata model file. Switches defined as type "Toggle", will not respond to a third party controller commands correctly

**Note** - All of the three control types can be changed in a single command. This means that a command string could include a change to a volume, a switch state, and an output state all at once, or any combination of the three. The order of the I, V, or O statements within the command is not important.

## Switches:

Switch state changes have the following format:    **CMD> U10022 I0\n**



U ID numbers are assigned at the time a control is created in the Sonata software. These U ID numbers are unique to each control in a Sonata layout file, and can be found in the properties dialog box for a given control. These ID numbers are automatically assigned during control creation in the software, but can be manually changed when desired. The U ID numbers are used to match physical controls in the real world with virtual controls created in software.

The state of a logic input is set to 0 for off, and 1 for on.

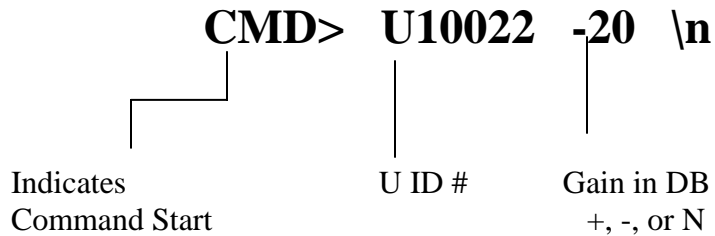
\n is a required terminator as described on page 1.

**Note** - Switches that are to be controlled by third party controllers must be defined as type “Static” in the Sonata model file. Switches defined as type “Toggle”, will not respond to a third party controller command.

## Volume:

Volume change strings have the following format:

CMD> U10022 -20\n



U ID numbers are assigned at the time a control is created in the Sonata software. These U ID numbers are unique to each control in a Sonata layout file, and can be found in the properties dialog box for a given control. These ID numbers are automatically assigned during control creation in the software, but can be manually changed when desired. The U ID numbers are used to match physical controls in the real world with virtual controls created in software.

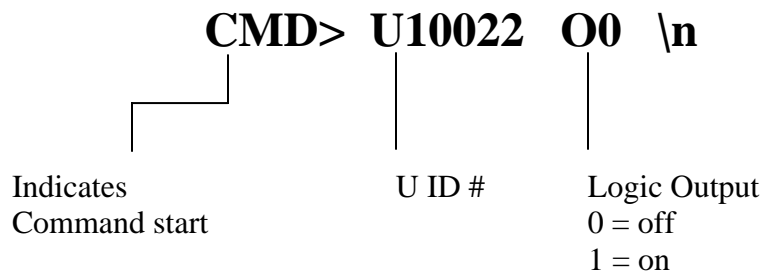
A level change is defined by a +, -, or N character. If + or - is used, the values range from -100 to +20. This would correspond to a range of -100dB to +20dB. If 'N' is used the values range from 0 to 120, and again correspond to a range of -100dB to +20dB. Also see below for use as a selector.

\n is a required terminator as described on page 1.

## Output:

Output change strings have the following format:

CMD> U10022 O1\n



U ID numbers are assigned at the time a control is created in the Sonata software. These U ID numbers are unique to each control in a Sonata layout file, and can be found in the properties dialog box for a given control. These ID numbers are automatically assigned during control creation in the software, but can be manually changed when desired. The U ID numbers are used to match physical controls in the real world with virtual controls created in software.

Output is on for a “1”, and off for a “0”

\n is a required terminator as described on page1.

### Multiple Command Lines:

Multiple commands can be embedded in a single string. An example format for multiple embedded commands is as follows: `CMD> U10022 -10 I1 U10023 I1 U10024 I0\n`

<b>CMD&gt;</b>	<b>U10022</b>	<b>-10</b>	<b>I1</b>	<b>U10023</b>	<b>I1</b>	<b>U10024</b>	<b>I0</b>	<b>\n</b>
Generic	Control	Volume	Switch	New	Switch	New	Switch	Line
Command	ID #	= -10dB	= on	ID #	= on	ID #	= off	Feed

Each U ID number represents a new control that is to be changed. A set of control changes is 'bracketed' by the `CMD>` identifying the string origin, and the terminator indicating the end of the string.

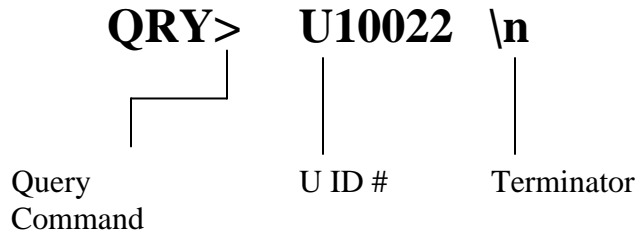
`\n` is a required terminator as described on page1.

Note – As with single commands, a multiple U ID command string can contain any or all of the volume, I, or O statements for a given U ID#, and order is not important.

## Request Data from the iFlex.

Data requests from the third party controller to the iFlex mixer have the following format:

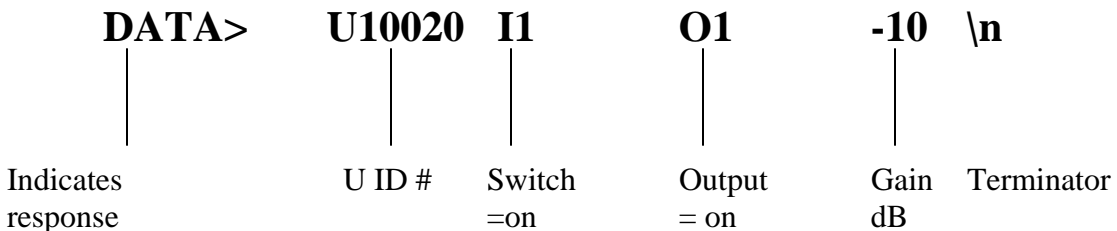
QRY> U10020\n



iFlex controls always contain a switch, an output, and a level control. This means that for each unique U Id# there will be an associated switch, and output, and a level. This does not mean that users will always use the three elements for a given U Id #, and often users will only create one of the elements of a control per U Id #. The iFlex mixer will, however, always return the states of all three elements for a given U Id # request.

The response string that is sent by the iFlex mixer to the third party control is as follows:

DATA> U10020 I1 O1 -10\n



## Multiple Queries

An example of a multiple U Id # query is as follows: `CMD> U10020 U10021\n`

<b>QRY&gt;</b>	<b>U10020</b>	<b>U10021</b>	<b>\n</b>
Query	U ID #	U ID #	Terminator

For the above request the response string will be:

DATA> U10020 I1 O0 +5\n

DATA> U10021 I0 O1 -20\n

## Selector Switches

iFlex mixers support analog or digital selector switches as a part of the programming layout. Analog versions use switches wired with resistors in series to create a voltage divider. Users may specify the number of discrete steps when the control is created in the layout.

Digital versions (such as those that may be implemented by third party controllers) are created using the same method. However, in the digital versions the controller will send a command to the iFlex mixer indicating the volume setting equal to the selector position. If a selector is defined in the layout, the string to update the selector to position 5 would be:

**CMD> U10020 N5\n**

Selectors in Sonata have several powerful programming features that reduce the number of modes, and hence programming time. Users should find selectors a very valuable tool in creating layout designs.

## Monitoring

When controls change state, they can be programmed to automatically report status to the RS232 port. These commands are used to setup this feature for each control in the design.

**MON> ALL \n**                      This will enable all controls to report  
**MON> NONE\n**                      This will disable all controls from reporting.  
**MON> U123 U45 ... \n**              This will enable the controls with matching id's  
**MON> N123 \n**                      The N delimiter will remove a control from reporting.

NOTE: These settings are non-volatile through power up and power down.

**QMON\n**                              Will return a list of controls and monitor status.

The response for this command will be as follows:

**MON> U23 N56 U27 ... \n**

If a control is monitored it will be preceded by U if not by N. All current control id's will be listed.



## System Information

Additional information about the system may be obtained by these commands:

**QIO**\n Requests the id's for each card or IO port in the system.

The response shall be:

```
IO> #1 U22 U34\n  
IO> #2 U23 U35\n  
...  
IO>#24 U567 U0\n  
DONE\n
```

There is one line for each port. If a port is empty or a secondary id is not used **U0** will be the result.

**QSN**\n Request the serial number.

Response shall be: **SN: 5312A123\n** This is the system serial number

**SETTIME> YYYY MM DD HH:MM:SS\n** Will set the internal clock

**QTIME**\n Will return the time as **TIME> YYYY MM DD HH:MM:SS\n**

Errors will be reported as **ERROR> MESSAGE\n** Where the message will inform the user as to cause.