



# RS232 Instrument Drivers

## *Instrument Driver Scripts*

Instrument Driver Scripts are text files used to control data acquisition and retrieval from modules, such as the RS232 serial module or LVA analog module. These driver scripts are saved in the modules and run as directed by DANTE's A and B sample programs.

Driver scripts are usually handled in DANTE Config software, either by selecting from the library of pre-defined scripts or by writing a new script in the DANTE Config editor. Scripts are saved in the configuration file output by DANTE Config and are passed to the modules when this file is sent to DANTE.

## Basic Script Structure

Driver scripts are little more than a list of commands representing a non-branching and non-looping algorithm<sup>1</sup>. Each line of a script represents a single command. When a command is completed the next command is processed. This repeats until the end of the file is reached.

## Script Data Output

All data received from the instrument is recorded to the module's data buffer. Bytes sent to the instrument are not recorded (unless they are echoed back by the instrument). The module also records any script processing errors or timeout conditions in the data buffer.

The DANTE A & B sample programs use the GETDATA command to retrieve the module's data buffer. All data retrieved by DANTE is saved in DANTE's data buffer (see DANTE's READFILE command), which is recorded on the SD memory card.

---

<sup>1</sup> Branching requires conditional statements. Looping requires statements such as GOTO, FOR or WHILE. These statements are not currently supported, but may be added in the future.

## Instrument Driver Options

Select a driver from the library in DANTE Config  
or  
Write a new driver in the DANTE Config editor  
or  
(for advanced users)  
Write a driver in a text editor

## Saving Drivers in Modules

Use DANTE Config to save a configuration file, then use a terminal program to send it to DANTE through the PC serial port.  
or  
(for advanced users)  
Use the OPEN MOD pass-through command and the module's WRITEFILE command to save the driver.



The module's data buffer is limited to 4096 characters<sup>2</sup>. When the data buffer is full any additional characters are dropped.

## RS232 Module Driver Script Commands

### SERIAL ON

Enables the module's serial port at the specified baud rate. The baud rate may be any multiple of 1200 up to 115200.

SERIAL ON 9600

Enables the serial port at 9600 baud (8-bits, no parity, 1 stop bit, no flow control)

### SERIAL OFF

Disables the modules serial port

### 12V ON / OFF / PERM

Enables or disables the module's 12V switched power output.

12V PERM causes the module to leave the output enabled after the script finishes, even if the module goes to sleep. This requires the `SLPOWERPERM=1` setting in DANTE, otherwise the 12V power supply to the modules is disconnected between samples.

12V OFF turns the output off immediately and keeps it off when the module sleeps.

### VB ON / OFF / PERM

Controls the module's VB (5V) switched power output.

VB PERM causes the module to leave the output enabled after the script finishes, even if the module goes to sleep. This requires the `SLPOWERPERM=1` setting in DANTE, otherwise the 5V power supply to the modules is disconnected between samples.

VB OFF turns the output off immediately and keeps it off when the module sleeps.

### BINARY

Sets the module to operate in binary data mode (default is ASCII). In binary mode all data received from the instrument is converted to ASCII hex. Note that data is stored as ASCII hex in the data buffer, meaning number of bytes stored in the sample data buffer is twice the length of the received binary data.

### ASCII

Sets the module to ASCII mode (default is ASCII). In ASCII mode all data sent to and received from the serial port is expected to be plain text. Non-text characters (>0x7F) may be ignored, cause events in the event counters and / or be replaced with other characters ('X', '\$' or '\*').

### PARSER

Specifies a default parser for the data generated by the instrument driver script. This setting may be overridden in DANTE Server or in DANTE Vis software, but cannot be overridden in data parsed by DANTE prior to Iridium transmission.

---

<sup>2</sup> The module data buffer size can be increased, please contact Soundnine if the standard 4096 character buffer is too small for your application.



PARSER SBE39\_A

Sets the parser to "SBE39\_A". When DANTE retrieves the data buffer it will be wrapped in XML tags:

```
<SampleData mid='001' lid='001' pt='SBE39_A'>
Received data appears here
</SampleData>
```

**CLEARBUFFER**

Clears the contents of the module's data buffer. This is intended to remove characters that result from transients on the serial data lines when an instrument is powered. Any data received before a CLEARBUFFER command is lost and not recorded in any log file.

**DELAY**

Waits a specified time in milliseconds, up to a maximum of 15 minutes. (15 min \* 60 seconds/min\*1000 milliseconds per second = 900000 milliseconds).

```
DELAY 100
```

Waits 100 milliseconds

**TIMESTAMP**

Records the current time in the data buffer. The timestamp looks like:

```
TIME=2013-11-05T11:57:00
```

If a module fails to receive the time commands sent automatically by DANTE then this format is possible:

```
TIME=S100
```

This usually means 100 seconds since DANTE started the A or B program.

**SEND**

Transmits characters to the module's serial port. Behavior of this command depends on the ASCII/Binary setting (see ASCII and BINARY commands)

*ASCII Mode (default)*

```
SEND "a string\r"
```

Sends 'a string' followed by carriage return (hex 0D) to the serial port.

The maximum string length is 64 characters.

A few common character sequences are available to send special or non-printing characters to the module serial port:

Sequence	Description	Hex Value
\r	Carriage Return	0D
\n	Line Feed	0A
\\	Front slash	2F
\"	Quote	22
\t	Tab	09

So SEND "test\r\n" will send 'test' followed by CR (hex 0D) and LF(hex 0A)

*Binary Mode*

```
SEND "6120737472696E670D"
```



Sends 'a string' followed by carriage return (hex 0D) to the serial port.  
The maximum string length is 32 binary characters, which is 64 bytes of ASCII hex.

#### WAITFOR

Waits for a string to arrive from the module's serial port, with a maximum delay. If the maximum delay time is reached without receiving the target string then the RS232 module adds 'S9M-TIMEOUT' to the received data buffer and proceeds to the next line of the script. Behavior of this command depends on the ASCII/Binary setting (see ASCII and BINARY commands)

#### *ASCII Mode*

```
WAITFOR "S>" 3000
```

Waits for the string 'S>' for up to 3 seconds.  
The maximum string length is 64 characters.

#### *Binary Mode*

```
WAITFOR "533E" 3000
```

Waits for the string 'S>' for up to 3 seconds.  
The maximum string length is 32 binary characters, which is 64 bytes of ASCII hex.