

MIDI Sysex Messages for the Eventide Factor series pedals

This technote assumes familiarity with the MIDI System Exclusive Message format, and the use of hex (hexadecimal) numbers.

The factor pedals accept some Standard Non-Real Time messages as well as proprietary messages using the following forms.

Standard Non-Real Time

0xF0 0x7E <channel number> <sub-ID#1> <sub-ID#2> 0xF7

the 0xF0 and 0xF7 are standard MIDI for start of system exclusive, and end of system exclusive. Note that 0xF0 (for example) is a hexadecimal representation of the decimal value 240, while 0xF7 (sometimes known as EOX) is decimal 247.

<channel number> must be either 0x7F or equal to the unit's configured SYSEX ID (0 in update mode).

<sub-ID#1> <sub-ID#2> give more information about the message type.

Supported Types:

General Information Request: sub-ID#1 = 0x06

Identity Request: sub-ID#2 = 0x01

This message may be sent in either normal or update modes and causes an ***Identity Reply*** message to be sent.

General Information Request: sub-ID#1 = 0x06

Identity Reply: sub-ID#2 = 0x02

This message is the reply to an ***Identity Request***. The format is as follows.

0xF0 0x7E <channel> 0x06 0x02 mm ff ff dd dd ss ss ss ss <text> 0xF7

0xF0 0x7E <channel>	Header
0x06	General Information (sub-ID#1)
0x02	Identity Reply (sub-ID#2)
mm	Manufacturer's ID – Eventide 0x1C
ff ff	Device Family Code (14 bits, LSB first)
dd dd	Device Family Member Code (14 bits, LSB first)
ss ss ss ss	Software Revision Level.
<text>	A long XML format string that gives further information on the unit.

MIDI Sysex Messages for the Eventide Factor series pedals

Eventide proprietary messages

0xF0 <EVENTIDE> <H4000> <id> <message_code> <lots-o-bytes> 0xF7

<EVENTIDE> 0x1C (decimal 28)

<H4000> 0x70 (decimal 112).

<id> device id number. If this is zero, all units will listen to the message. Otherwise it must match the unit's configured SYSEX ID.

<message_code> tells us what message this is. The various messages are described below.

<lots-o-bytes> is the rest of the message. This **data** depends on the type of message. Not all messages have 'lots-o-bytes'.

With many messages, a "byte" is actually two bytes. Since MIDI allows only 7 bits of data, we split an 8 bit byte into two 4 bit *nibbles* and send the nibbles. The most significant nibble is sent first.

Message codes:

SYSEXC_OK 0x00

Data: None. This message is returned by the unit in response to assorted commands. It simply says everything was ok with that last command.

Response: None.

SYSEXC_ERROR 0x0D

Data: This message is returned by the unit in response to assorted commands. The message indicates that an error occurred with the last command. <lots-o-bytes> may contain a ascii text error message (not split into nibbles)

Response: None.

SYSEXC_PROGRAM_DUMP 0x15

Data: A program in binary form, the format being the same as SYSEXC_FILES_DUMP.

Response: None.

SYSEXC_VALUE_PUT 0x2d

Data: Send a new value for a parameter.

Parameter 1 (ascii hex) is the key

Parameter 2 (ascii number) is the new value

Response: A SYSEXC_VALUE_DUMP message.

SYSEXC_VALUE_DUMP 0x2e

Data: A value of a parameter in ascii hex

Response: N/A

MIDI Sysex Messages for the Eventide Factor series pedals

SYSEXC_OBJECTINFO_WANT 0x31

Data: Same as SYSEXC_VALUE_WANT. A ascii key value in hex.

Response: A SYSEXC_VALUE_DUMP message.

SYSEXC_VALUE_WANT 0x3b

Data: A ascii key value in hex.

Response: A SYSEXC_VALUE_DUMP message.

SYSEXC_TJ_PRESETS_WANT 0x48

Data: None.

Response: The unit sends a SYSEXC_TJ_PRESETS_DUMP message

SYSEXC_TJ_PRESETS_DUMP 0x49

Data: A dump of the all the presets in the unit. See Appendices for more information.

Response: n/a

SYSEXC_TJ_SYSVARS_WANT 0x4c

Data: None.

Response: The unit sends a SYSEXC_TJ_SYSVARS_DUMP message.

SYSEXC_TJ_SYSVARS_DUMP 0x4d

Data: A dump of the current system parameters. This message is to be used for backup purposes and will not be further documented.

Response: n/a

SYSEXC_TJ_PROGRAM_WANT 0x4e

Data: None.

Response: The unit sends a SYSEXC_TJ_PROGRAM_DUMP message.

SYSEXC_TJ_PROGRAM_DUMP 0x4f

Data: A dump of the current preset. See Appendices for more information.

Response: n/a

SYSEXC_TJ_ALL_WANT 0x50

Data: None.

Response: The unit sends a SYSEXC_TJ_ALL_DUMP.

SYSEXC_TJ_ALL_DUMP 0x51

Data: The entire state of the unit. This is a single message, comprising a SYSEXC_TJ_SYSVARS_DUMP followed by a SYSEXC_TJ_PRESETS_DUMP

Response: None.

MIDI Sysex Messages for the Eventide Factor series pedals

<i>SYSEXC_TJ_REBOOT_SEND 0x56</i>
--

Data: None.

Response: The unit sends a SYSEXC_TJ_REBOOT_ACK message and then resets into update mode. Note that the reset operation will cause the USB connection to become invalid.

<i>SYSEXC_TJ_REBOOT_ACK 0x57</i>

Data: None.

Response: None.

Appendix A – Program Dump Format

```
[8] 0 2
3 3f20 0 7fe0 102 3ee0 7ec2 36c0 65e0 1a80 7fe0 3ff0
3ea0 3f20 0 0 4000 7fe0 60 5dc0 0 3ee0 7fa0 0 0 0 0 7fe0 19a0 1a80 7fe0 7fe0 0 18eb 0
65000.000 65000.000 65000.000 65000.000
C_1322
```

The first line in this example has the following fields:

preset number, starting at 1
algorithm number. This is usually 0
dump format number. This is currently always 2. If it changes, the dump format may be different

[8] 0 2

The second line in this example has the following fields:

effect number (encoder setting), 0 to 9
knob values in hex, 0 to 7FE0, reading from bottom left knob (*Xknob, D-Mod*) to top left knob (*Mix/Intensity*)
pedal value in hex, 0 to 7FE0

3 3f20 0 7fe0 102 3ee0 7ec2 36c0 65e0 1a80 7fe0 3ff0

The first 20 values on the next line are in pairs, one pair for each knob and cover the pedal to knob mapping. The first value in each pair is the knob value when the pedal is at minimum (up), the second is the knob value when the pedal is at maximum (down). They are between 0 and 7FE0. They are in the same order as the knob values above.

3ea0 3f20 0 0 4000 7fe0 60 5dc0 0 3ee0 7fa0 0 0 0 0 7fe0 19a0 1a80 7fe0 7fe0 0 18eb 0

The remaining three values are:
unused - leave at 0
tempo*100
tempo on/off (1/0)

The next 4 values are too hard to explain and have functions that will vary from product to product. They will either be 65000.00 (inactive) or some small floating point value. Either leave them as they are or set them inactive.

65000.000 65000.000 65000.000 65000.000

The final line is a **checksum**, which is the integer sum of all the values except for the first line. This is used as an integrity check - a preset with a bad checksum will be ignored. If your arithmetic is not very good, use the value **C_XXXX** instead, and it will be assumed correct.

C_1322