System Exclusive (ie, SysEx)

Category: System Common

Purpose

Used to send a large amount of data to a MIDI device, such as a dump of its patch memory or sequencer data or waveform data. Also, SysEx may be used to transmit information that is particular to one model device. For example, a SysEx message might be used to set the feedback level for an operator in a Roland Physical Modeling Synth. This information would likely be useless to an AKAI sample playing device. (By contrast, virtually all devices respond to Modulation Wheel control, for example, so it makes sense to have a defined Modulation Controller message that all manufacturers can support for that purpose).

Status

Begins with 0xF0. Ends with a 0xF7 status (ie, after the data bytes).

Data

There can be any number of data bytes inbetween the initial 0xF0 and the final 0xF7. The most important is the first data byte (after the 0xF0), which should be a Manufacturer's ID.

Errata

Virtually every MIDI device defines the format of its own set of SysEx messages (ie, that only it understands). The only common ground between the SysEx messages of various models of MIDI devices is that all SysEx messages must begin with a 0xF0 status and end with a 0xF7 status. In other words, this is the only MIDI message that has 2 Status bytes, one at the start and the other at the end. Inbetween these two status bytes, any number of data bytes (all having bit #7 clear, ie, 0 to 127 value) may be sent. That's why SysEx needs a 0xF7 status byte at the end; so that a MIDI device will know when the end of the message occurs, even if the data within the message isn't understood by that device (ie, the device doesn't know exactly how many data bytes to expect before the 0xF7).

Usually, the first data byte (after the 0xF0) will be a defined ***Manufacturer's ID***. The MMA has assigned particular values of the ID byte to various manufacturers, so that a device can determine whether a SysEx message is intended for it. For example, a Roland device expects an ID byte of 0x41. If a Roland device receives a SysEx message whose ID byte isn't 0x41, then the device ignores all of the rest of the bytes up to and including the final 0xF7 which indicates that the SysEx message is finished.

The purpose of the remaining data bytes, however many there may be, are determined by the manufacturer of a product. Typically, manufacturers follow the Manufacturer ID with a Model Number ID byte so that a device can not only determine that it's got a SysEx message for the correct manufacturer, but also has a SysEx message specifically for this model. Then, after the Model ID may be another byte that the device uses to determine what the SysEx message is supposed to be for, and therefore, how many more data bytes follow. Some manufacturers have a checksum byte, (usually right before the 0xF7) which is used to check the integrity of the message's transmission.

The 0xF7 Status byte is dedicated to marking the end of a SysEx message. It should never occur without a preceding 0xF0 Status. In the event that a device experiences such a condition (ie, maybe the MIDI cable was connected during the transmission of a SysEx message), the device should ignore the 0xF7.

Furthermore, although the 0xF7 is supposed to mark the end of a SysEx message, in fact, any status (except for Realtime Category messages) will cause a SysEx message to be considered "done" (ie, actually "aborted" is a better description since such a scenario indicates an abnormal MIDI condition). For example, if a 0x90 happened to be sent sometime after a 0xF0 (but before the 0xF7), then the SysEx message would be considered aborted at that point. It should be noted that, like all System Common messages, SysEx cancels any current running status. In other words, the next Voice Category message (after the SysEx message) must begin with a Status.