SEQUENCER MIDI

FUNCTION		TRANSMITTED	RECOGNIZED	COMMENTS
Channel	1	yes	yes	Great division
	2	yes	yes	Pedal division
	3	yes	yes	Swell (Theatre Solo)
	4	yes	yes	Choir/Positive (Theatre Accomp.)
	5	yes	yes	Solo (Theatre Orchestral)
	6	yes	yes	Echo/Antiphonal/Gallery
	7	yes	yes	Bombarde
8 n/u	9	yes	yes	Great (second touch)
	10	yes	yes	Pedal (second touch)
	11	yes	yes	Swell (second touch)
	12	yes	yes	Choir (second touch)
	13	yes	yes	Solo (second touch)
14 n/u	15	yes	yes	Theatre Effects
	16	yes	yes	Controls (tabs, etc.)
Mode		Mode 3	Mode 3	OMNI OFF, POLY
Note Number		36-67 36-96	36-67 36-96	Pedal (Ch. 2 and 10) (See note 1) Manuals (Ch 1, 3-7, 9, 11-13)
Velocity		optionally from expression pedals only	no but is transmitted out expander port	
After Touch		no	no	
Pitch Bender		no	no	
Controllers	7	no	no	Volume
	11	yes	yes	Expression (See note 2)
	18	yes (note 4)	yes (notes 5,6)	Midi memory level (1-n)
	64	yes	yes	Sustain - sent out Midi expander port
	80	yes	yes	Turn tab *t* off (*t* = 0 - 127)
	81	yes	yes	Turn tab *t* on (*t* = 0 - 127)
	82	yes (note 4)	yes (note 5)	Crescendo stage number (0-n)
	83	yes (note 4)	yes (note 5)	Memory level (1-n)
	100,101	yes	yes	RPN 02 coarse tuning (transposer)
Program Change		yes (note 4)	yes (note 4)	combination pistons and cancels
System Exclusive		yes	yes	bulk stop-change data / memory data
System Common		no	no	
System Real Time				
Clock		no	no	
Start		no	yes	See Note 3.
Stop		no	yes	Treated as "All Notes Off"
System Reset		yes	yes (note 4)	Restart console CPU, attached CPUs.
All notes off		no	no	

NOTES

1 Note numbers received outside of this range are transposed one or more octaves up or down as required to get into the allowed range.

2 Expression is only sent on channels which have an associated expression pedal (swell show) or other means of expression.

3 CPU transmits status of all tabs via Sys.Ex., swell shoes via controller 11 on affected channels; memory l evel, midi memory level, and transposer data (RPN 02), on receipt of "START".

4 May be disabled via Soft-Switch

5 May be disabled via MidiRegOff piston

6 May be disabled via Multi-Midi on

EXPANDER MIDI

FUNCTION	TRANSMITTED	RECOGNIZED	COMMENTS
Channel 1	yes	yes	MIDI stop 1
2	yes	yes	MIDI stop 2
...			... (See note 2)
16	yes	yes	MIDI stop 16
Mode	Mode 3	Mode 3	OMNI OFF, POLY
Note Number	24-111	0-127	Note received transposed to within 24-111 range.
Velocity	optionally from expression pedals only	ignored	See notes 4 and 1.
After Touch	no	no	
Pitch Bender	no	no	
Controllers			
1	yes	yes	modulation
7	yes	yes	volume
10	yes	yes	pan
11	yes	yes	expression (See note 3)
18	yes	no	Midi memory level
64	yes	yes	Sustain
82	yes (note 6)	no	Crescendo stage number (0-n)
83	yes (note 6)	no	Memory level (1-n)
Program Change	yes	yes	
System Exclusive	yes	yes	See notes 5 and 1.
System Common	no	no	
System Real Time			
Clock	no	no	
Reset	yes	no	
Aux. Messages	no	no	
All notes off			

NOTES

1 All MIDI data received on MIDI-IN is re-transmitted on MIDI-OUT.

2 From one to sixteen MIDI-stops may configured. At least one MIDI-stop is configured for each keyboard, even if no tab (ON/OFF) is present. An initialization message, which is sent via MIDI-Out when a MIDI-stop is turned on, may contain various elements, such as Program Change, Volume, Modulation, Pan, Expression, GS-MIDI Bank, etc.

3 Expression is only sent on channels which have an associated expression pedal (swell shoe), or other means of expression. An expression pedal may send expression data as any controller number (configurable).

4 Velocity value sent may be preset, or derived from expression pedal.

5 A Sys.Ex. message may be included in an initialization message.

6 May be disabled via Soft-Switch

MIDI IN/OUT
ON THE
CONSOLE CONTROL COMPUTER
BY
Classic

2800 John Street, Unit 4
Markham, Ontario, Canada
L3R 0E2

Tel.: (905) 475-1275
Fax.: (905) 475-2735

E-mail: classic@organworks.com
WWW: http://www.organworks.com

1. INTRODUCTION

The Classic Console Control Computer features two MIDI ports: a pair of MIDI in/out jacks for sequencers and a pair for expanders. This document describes the MIDI data that is available from these ports.

2. SEQUENCER MIDI IN/OUT

The Sequencer MIDI output provides note on/off data, expression data, and combination piston data for the keyboards and pedals on the MIDI channels as follows:

Manual	MIDI Channel	Types of Messages
Great	1	key, exp., div. pistons
Pedal	2	key, exp., div. pistons
Swell	3	key, exp., div. pistons
Choir	4	key, exp., div. pistons
Solo	5	key, exp., div. pistons
Antiphonal (Echo or Gallery)	6	key, exp., div. pistons
Bombarde	7	key, exp. ,div. pistons
	8	div. pistons for couplers
Great second-touch	9	key., div. pistons
Pedal second-touch	10	key., div. pistons
Swell second-touch (Solo)	11	key., div. pistons
Choir second-touch (Accomp)	12	key., div. pistons
	13	
	14	
Effects pistons (theatre organs)	15	key (representing a percussion device)
Control messages	16	stops, transposer, general pistons

SEQUENCER MIDI CHANNEL TABLE

2.1 KEY DATA

Key-on data is sent as MIDI *note on*. If a velocity-sensing keyboard is being used, the velocity value is obtained from the keyboard; otherwise if a normal organ keyboard is used, a velocity value of 127 is sent. Key-off data is sent as MIDI *note on* with velocity 0. For example, 092h,40h,7Fh means note # 64 on with velocity 127. At present, only MIDI Expander voices respond to velocity values.

2.2 EXPRESSION DATA

Expression data is sent as MIDI Controller #11 and a number from 0 to 127, with the controller data value obtained from an expression shoe. For example, 0B2h,0Bh,3Fh means set Swell expression to value 63

2.3 STOP-TAB DATA - Small Organs

When stop-tabs (including drawknobs, lighted rockers, lighted pushbuttons or whatever the device for turning individual stops on or off) are moved individually by hand, MIDI controller messages are sent on MIDI channel 16. Controllers 80 (50h) and 81 (51h) are most commonly used. These controllers can handle all tabs on an organ with up to 128 tabs, and can be used to control the first 128 tabs on organs with more than 128 tabs.

Whenever a tab is turned on, a controller message for controller 81, with the appropriate tab-number, is sent on Sequencer MIDI OUT. Channel 16 is used, so the hexadecimal form of the message is:

0BFh 51h *nn*

where *nn* is from 0h to 7Fh representing tab-numbers 1 to 128.

When a tab is turned off, the message sent is:

0BFh 50h *nn*

When a controller 80 or 81 message is received on sequencer MIDI, regardless of the MIDI channel, the appropriate tab will be turned off or on, depending on the controller number.

Controllers 80 and 81 are restricted to tabs 1 through 128 by the MIDI data format. Organs with more than 128 tabs avoid this limitation by using the controller pairs, 16, 48 (10h, 30h) to turn tabs off and 17, 49 (11h, 31h) to turn tabs on.

2.4 STOP-TAB DATA - Large Organs

For organs with more than 128 stop-tabs, when a high-numbered tab is turned on, a controller 17 (11h) message is sent giving the multiples of 128 in the tab number, then a controller 49 (31h) message is sent giving the remainder. For example, if tab 261 were turned on, the resulting hexadecimal message would be:

0BFh 11h 02h 31h 04h

11h and 31h are the controller numbers. 02h and 04h form the hexadecimal tab number: 2 times 128 plus 4. In MIDI messages, tabs are counted from 0 for the first tab; in common speech we count from 1, so the calculated number indicates tab 261.

If the same tab were turned off, controllers 16 (10h) and 48 (30h) are used and the message sent would be:

0BFh 10h 02h 30h 04h

If one of the two messages shown above were received via sequencer MIDI IN, Tab 261 would be turned on or off.

Where tabs in the same range (e.g. 129 to 256, 257 to 384 etc.) are turned on together, only one controller 17 (11h) message need be sent. This technique is known as *maintaining a running status*. If tabs 263 and 265 are turned on together, sequencer MIDI OUT would show:

0BFh 11h 02h 31h 06h 31h 08h

The controller 17 (11h) setting is saved and applied to the two controller 49 (31h). Similarly, to turn off tabs 263 and 265, the messages sent would be:

0BFh 10h 02h 30h 06h 30h 08h

The pairs of controllers 16,48 and 17,49 work independently. Controller 16 sets the range for use by controller 48 in setting tabs off, controller 17 sets the range for controller 49 in setting tabs on.

The MIDI running-status technique is used on MIDI OUT. Where successive controller messages are sent on the same channel, the controller status byte (the 0BFh in the examples above) only needs to be sent before the first pair of controller-number/controller-value data bytes. Remember this if you are examining messages in a MIDI data stream.

2.5 STOP-COMBINATION DATA

When pistons are pressed, one of two types of MIDI messages are sent out the sequencer-MIDI port:

1 Piston-data, or
2 System Exclusive data.

Which of the two types of messages is sent depends on the setting in the Console Control Computer of an internal switch known as a *softswitch*. How to display and change the status of the softswitch is described in the Organ Reference Manual.

2.5.1 PISTON DATA

If the softswitch on the Console Control Computer is set for piston data, then a MIDI Program Change message is sent whenever a combination piston is pressed. The channel-number of the message corresponds to the division of the piston (see Sequencer MIDI Channel Table above) with general pistons being on channel 16, and the program-number corresponds to the number of the piston, with program-number zero being for cancel pistons. In other words, Divisional Cancel = Program Number 0 on the channel associated with that division (see Sequencer MIDI Channel Table above); divisional pistons (1-n) = Program Number 1-n; General Cancel = Program Number 0 on channel 16; General Pistons (1-n) = Program Number 1-n on channel 16.

Program Change messages received on a MIDI channel will activate the piston or cancel of the division associated with that channel.

2.5.2 SYSTEM EXCLUSIVE - Registration Change Message

If the softswitch on the Console Control Computer is set for registration data to be sent and received via MIDI System Exclusive (SysEx) messages, then when a divisional or general piston is pressed, the status of all the stop tabs on the organ is sent out the sequencer MIDI-OUT as a SysEx. Similarly, when a sequencer sends in one of these SysEx messages, the stop tabs are adjusted as if a general piston had been pressed.

The following are examples for an organ with 40 tabs and 8 lighted reversible pistons (LDR's). The format of the SysEx message is (all numbers are hexadecimal values):

```
(F0) Start of SysEx (MIDI standard)
 | (00,20,1C) Artisan Classic Organ Inc. - Manufacturer's ID
 | |             (00) Function Code (0==> Registration change)
 | |              |  (aa) First of up to eight 7-bit words which
 | |              |   |   combine to make up to 56 tab settings.
 | |              |   |   (See note below)
 | |              |   |                     (F7) End of SysEx
 | |              |   |                      |
F0 00 20 1C 01 01 00  00 01 00 00 00 00 00  F7  ==> tab#1 only
F0 00 20 1C 01 01 00  00 02 00 00 00 00 00  F7  ==> tab#2 only
F0 00 20 1C 01 01 00  00 04 00 00 00 00 00  F7  ==> tab#3 only
F0 00 20 1C 01 01 00  00 08 00 00 00 00 00  F7  ==> tab#4 only
F0 00 20 1C 01 01 00  00 0F 00 00 00 00 00  F7  ==> tab#1,2,3,4
F0 00 20 1C 01 01 00  00 10 00 00 00 00 00  F7  ==> tab#5 only
F0 00 20 1C 01 01 00  00 20 00 00 00 00 00  F7  ==> tab#6 only
F0 00 20 1C 01 01 00  00 40 00 00 00 00 00  F7  ==> tab#7 only
F0 00 20 1C 01 01 00  40 00 00 00 00 00 00  F7  ==> tab#8 only
F0 00 20 1C 01 01 00  40 70 00 00 00 00 00  F7  ==> tab#5,6,7,8
F0 00 20 1C 01 01 00  40 7F 00 00 00 00 00  F7  ==> tab#1-8 inc.
F0 00 20 1C 01 01 00  00 00 01 00 00 00 00  F7  ==> tab#9 only
F0 00 20 1C 01 01 00  20 00 00 00 00 00 00  F7  ==> tab#16 only
F0 00 20 1C 01 01 00  04 00 00 00 00 00 00  F7  ==> tab#40 only
F0 00 20 1C 01 01 00  00 00 00 00 00 00 01  F7  ==> tab#41 (LDR#1)
F0 00 20 1C 01 01 00  02 00 00 00 00 00 00  F7  ==> tab#48 (LDR#8)
```

REGISTRATION CHANGE SYSEX

Note: Registration Data is the binary image of the tab data table in the memory of the organ's Console Control Computer. Each seven bytes of data in the Console Control Computer's table are sent as eight bytes in the MIDI message. The first byte (shown as 'aa') contains the bits representing tabs #8, #16, #24, #32, #40, #48 and #56. The next one to seven bytes contain the bits representing tabs #1 through #7, #9 through #15, and so on up to tabs #49 through #55 for the seventh byte. If there are more than 56 tabs and reversibles, the (up to) eight-byte groups are repeated as many times as necessary.

2.6 MEMORY-DATA TRANSFER

Using a sequencer and the Sequencer MIDI In/Out port of the Console Control Computer, various parts of the Console Control Computer's memory can be saved or restored. The parts of the memory that can be saved are:

1. Piston-memories
2. Crescendo-pedal setup
3. MIDI-tab memories
4. Registration-sequencer memory
5. Configuration data
6. Non-Configuration data (full backup)

This feature can be useful if an organist has a lot of combinations set up on the pistons, etc. and does not want to lose them when another organist plays the organ. The first organist can save all he needs to, let the second organist set whatever he wants and then when the second organist is finished playing, the first organist can restore all his combinations. Another use for this feature can be when some sort of service is being done on the organ, especially the Console Control Computer. To ensure that nobody tampers with the set-up, it can be saved before the service is performed and then restored afterwards.

This data in the Console Control Computer's memory is transferred to and from the computer by sending and receiving a number of different MIDI System Exclusive messages. They are as follows.

Equipment Identification

a. General Information Identity Request.

F0 7E *dd* 06 01 F7

Request the identity and special characteristics of a device. *dd* selects a particular device in the MIDI chain. *dd* = 7Fh selects any and all devices at once.

b. General Information Identity Reply

F0 7E *dd* 06 02 00 20 1C *pp pp mm mm ss ss* 00 00 F7

dd identifies the device which is responding.

pp pp = 01 00 for Console Control Computer
02 00 for TG114
03 00 for Pipe Control Computer

mm mm gives the organ model number.

ss ss identifies the major and minor software revision level.

The TG114 and the Pipe Control Computer are two other products by Classic.

Registration Data Formats

Several formats for transmitting registration data are recognized to ensure that data saved with earlier versions of the Console Control Computer software is still usable.

a. F0 00 20 1C *dd* 01 00 *oo* <Registration data, 8-for-7 format> F7

dd is the device ID of the device responding

oo identifies the byte offset within TDT for the following data (0..127)

This format is sent and recognized by all Console Control Computers from version 4.22 on. The 8-for-7 format is described in reference [b] page 13.

b. F0 3F 00 <Registration data, 2-for-1 format> F7

This is an old experimental format. It is recognized but never sent by current organs.

c. F0 00 20 1C 00 <Registration data, 8-for-7 format> F7

Recognized, but not sent, by all Console Control Computers. Because of the existence of this format, the device id numbers used in Classic products must run from 1 to 127. A device id of 0 (zero) cannot be used. The 8-for-7 format is described in reference [b] page 13.

d. F0 00 20 1C *dd* 01 01 <sub-command> <TBA> F7

dd is the device ID of the sending device. This format is reserved for future registration messages.

Combination Memory

These formats are identified by a command-id of 02.

a. Request organ configuration parameters and data sizes.

F0 00 20 1C *dd* 01 02 01 F7

dd is the device ID of the device for which the memory dump is requested.

b. Organ Configuration Data dump

F0 00 20 1C *dd* 01 02 02 *vv vv* <Combo-Memory parameters> F7

dd is the device ID of the sending device.

vv vv is the major and minor software version numbers.

Combo-memory parameters are:

nn	size of this data block (number of data bytes from version number (*vv vv*) to SQMAX).
nn	number of bytes per general piston (GNUM)
nn	number of extension bytes per combo-memory (ENUM)
nn	number of divisions (DIVSNS)
nn	number of crescendo stages (NCNUM)
nn	number of general pistons (GENS)
nn	number of sforzando pistons (NSFORZ)
nn	number of divisional pistons per division (DIVSS)
nn	number of ventils (NVENTL)
nn	number of combo-action memories (MEMNUM)
nn	number of registration-sequencer stages (SQMAX)
nn nn	number of bytes of MIDI configuration data (MIDTSZ)
nn nn	the model number of this organ (JOBNUM)
nn	number of crescendi (NUM_CRESC)
nn	0/1 independent mask for each piston (MASK_PER_PISTON)
nn	0/1 independent memory for each division (INDEPENDENT_DIV_MEM)
nn	size of expression limit table (ROM_EXP_SIZE)
nn	size of Pizzicato table (ROM_PIZZICATO_SIZE)

c. Request combo-memory data

F0 00 20 1C *dd* 01 02 03 *mm* F7

dd	device ID of the device from which the memory is requested
mm	memory-number between 1 and 99

d. Combo-memory data transfer

F0 00 20 1C *dd* 01 02 04 *mm oo oo ss* <*data*, 8-for-7 format> F7

dd	device ID of sending device
mm	memory number 1..99
oo oo	offset of this data within memory *mm*'s data, low-order 7 bits first.
ss	number of bytes of original data transferred in this block.
data	combo-memory data in 8-for-7 format (See reference [b] page 13.)

Crescendo-Memory

These formats are identified by a command-id of 03.

a. Request for crescendo data

F0 00 20 1C dd 01 03 03 *mm* F7

dd	device ID of device for which data is requested
mm	crescendo number 1-n

b. Crescendo data transfer

F0 00 20 1C *dd* 01 03 04 *mm oo oo ss* <*data*, 8-for-7 format> F7

dd	device ID of sending device
mm	crescendo number 1-n
oo oo	offset of this data within crescendo data, low-order 7 bits first
ss	number of bytes of original data transferred in this block.

data crescendo data in 8-for-7 format (See reference [b] page 13.)

Registration Sequencer Memory

These formats are identified by a command-id of 04.

a. Request for Registration Sequencer data
F0 00 20 1C *dd* 01 04 03 00 F7
dd device ID of device for which data is requested

b. Registration Sequencer data transfer
F0 00 20 1C *dd* 01 04 04 00 *oo oo ss* <*data*, 8-for-7 format> F7
dd device ID of sending device
oo oo offset of this data within sequencer data, low-order 7 bits first
ss number of bytes of original data transferred in this block.
data Registration Sequencer data in 8-for-7 format (See reference [b] page 13.)

MIDI Tab Memory

These formats are identified by a command-id of 05.

a. Request for MIDI-tab data
F0 00 20 1C *dd* 01 05 03 *mm* F7
dd device ID of device for which data is requested
mm midi memory number 1-n

b. MIDI-tab data transfer
F0 00 20 1C *dd* 01 05 04 *mm oo oo ss* <*data*, 8-for-7 format> F7
dd device ID of sending device
mm midi memory number 1-n
oo oo offset of this data within MIDI-tab memory data, low-order 7 bits first
ss number of bytes of original data transferred in this block.
data MIDI-tab data in 8-for-7 format (See reference [b] page 13.)

Configuration Memory Send

This format is identified by a command-id of 06.

a. Request for Configuration Memory (received only by CCC)
F0 00 20 1C 01 01 06 03 00 F7

b. Configuration Memory transfer
F0 00 20 1C *dd* 01 06 04 *00 oo oo ss* <*data*, 8-for-7 format> F7
dd device ID of sending device
oo oo offset of this data within Configuration memory data, low-order 7 bits first
ss number of bytes of original data transferred in this block.

All Memory Send

This format is identified by a command-id of 07. The block consists of all non-configuration data (Crescendo, Combination Piston, Registration Sequencer, and MIDI-Stop Memories).

These formats are identified by a command-id of 03.

a. Request for All Memory Data transfer (received only by CCC)
F0 00 20 1C 01 01 07 03 00 F7

b. All Memory Data transfer
F0 00 20 1C *dd* 01 07 04 *00 oo oo ss* <*data*, 8-for-7 format> F7
dd device ID of sending device

oo oo offset of this data within memory data, low-order 7 bits first

ss number of bytes of original data transferred in this block.

2.7 TRANSPOSER DATA

The organ's transposer controls generate values for MIDI Registered Parameter 2 (coarse tuning). In accordance with MIDI standards, the parameter is adjusted by the following sequence of controller messages.

0BFh 65h 0
0BFh 64h 2
0BFh 06h *nn*

where *nn* is the tuning value from 59 to 70, for tuning from 5 semitones below standard to 6 above. A value of 64 is standard pitch. The first two controller messages specify that MIDI Registered Parameter 2 is to be adjusted. The third message specifies the value.

Receiving this sequence of messages in the sequencer MIDI IN causes the organ to set the transposer. Values outside the range of 59-70 are ignored

2.8 MEMORY LEVEL

Combination action memory level is sent/received via controller #83.

0BFh 53h *nn* where nn = 1...maximum for this organ

Midi memory level is sent/received via controller #18.

0BFh 12h *nn* where nn = 1...maximum for this organ

Combination action memory level is sent whenever the level changes for any reason (up/down pistons, direct access pistons, etc.). The send may be disabled via soft-switch #24.

Midi memory level is sent when there is a change in midi memory level. The send may be disabled via soft-switch #24. Reception may be disabled via Multi-Midi on, where the midi level would be always tied to the combination action level, regardless of the midi level received via the sequencer port.

3. EXPANDER MIDI IN/OUT

The second pair of MIDI jacks is primarily intended for driving a MIDI expander (a synthesizer, usually without keyboard, which produces a variety of musical sounds, in response to keying information supplied through a MIDI input). An Expander allows the organist to expand the tonal resources of the organ simply by plugging in any MIDI-compatible sound generator. Sounds (voices) available on the expander can be assigned to MIDI stop-tabs on the organ console, so that these MIDI stops may be controlled in the same manner as pipe ranks, or electronic stops. The MIDI stops will respond to the couplers, transposer and expression pedals in exactly the same way as other organ stops in the same division. The MIDI stop-tabs can be programmed so that, when the tab is turned on, a set-up message is sent to the expander to select a voice (patch), adjust volume, etc. The Roland CM-32P, SC-55, or EMU Proteus, or similar units can easily be connected.

The expander MIDI-IN may be connected to a keyboard or a controller such as the Roland CN-20 or Lexicon MRC. Messages received on expander MIDI-IN are copied to the expander MIDI-OUT. With a CN-20 connected to the expander MIDI-IN and a synthesizer of some sort connected to the MIDI-OUT, you can adjust control values on the synthesizer that are not controlled by the organ console itself. Since the CN-20 can be switched to send on any channel, you can adjust the synthesizer settings for any MIDI stop.

The Expander MIDI-IN may also be used to connect a synthesizer or controller keyboard which is velocity sensitive, has after-touch, etc., or other features which are not normally available from the regular organ keyboards.

The organ for which the following table applies, has four MIDI-stops defined on stop tabs, configured (by default) as follows:

Normally, a different MIDI channel is assigned to each MIDI-stop. A synthesizer that can produce sounds for several

MIDI-STOP CONTROL	DIV.	CHANNEL	VELOCITY	PATCH	VOLUME	PAN	MOD.	EXPRESS.
Tab#9	Pedal	1	127	#1	127	64	0	100
Tab#23	Swell	2	127	#2	127	64	0	(swell shoe)
Tab#37	Great	3	(exp.ped.)	#3	127	64	0	127
Tab#38	Great	4	127	#4	100	64	0	(exp.ped.)

different stops (ie. multi-timbral) can distinguish between the stops by the channel-number that each is sending on. However, there are no restrictions and it is entirely possible to have any or all MIDI-stops sending on the same MIDI channel, for example, if you want a *layered* sound.

3.1 MIDI STOP-TAB CONFIGURATION

When a MIDI stop-tab is turned from off to on, pre-programmed messages are sent in that MIDI-stop's channel to set the voice-type (patch), volume, and other controls to the proper values for that stop.

Each MIDI-stop has a variety of parameters that can be set by the organist. Those parameters that are sent out the expander MIDI OUT are as follows.

MIDI Parameter	Sent on Activation	Range	Description
Channel Number	no	1-16	The MIDI channel. Normally, a different channel-number is assigned to each MIDI stop.
Program Number	yes	ϕ, 0-127	The MIDI Program Number (also known as Patch Number)
volume level	yes	ϕ, 0-127	
pan	yes	ϕ, 0-127	Left/right panning value sent via controller 10. A value of 64 will send equal signal levels to both audio channels.
modulation	yes	ϕ, 0-127	The modulation or tuning sent with controller 1.
expression	yes/no	ϕ, 0-127	Expression value, sent on activation if not null and if stop not in velocity mode. See note 1 below.
velocity	no	ϕ, 0-127	The velocity value normally sent with note-on messages when MIDI-stop not in velocity mode. See note 1 below.
expression controller	no	ϕ, 7, 11	The MIDI controller number which specifies volume levels through the operation of the swell-pedal associated with this MIDI-stop's channel. ϕ means no expression values are sent. See note 2.
GS-MIDI Bank Select (LSB)	yes	ϕ, 0-127	Low-order byte of Bank Select value. See note 3.
GS-MIDI Bank Select (MSB)	yes	ϕ, 0-127	High-order byte of Bank Select value. See note 3.

Note 1: If the organ does not have velocity-sensing keyboards, the velocity value sent with each note-on message can be set to a fixed value, or if desired, the MIDI stop can be set to *velocity mode*, where the velocity is derived from the position of any expression pedal (swell shoe) in the organ. Furthermore, the minimum and maximum velocity or expression values derived from an expression pedal can be limited and scaled to provide a usable output range for any expander, independently for each MIDI-stop.

Note 2: Independently for each MIDI-stop, the controller-number used for expression (derived from the swell shoe) can be changed to suit the expander being used. For example, most Roland equipment uses Controller 11 for expression, while using Controller 7 for voice-volume setting, while an EMU Proteus may more easily be configured to use Controller 7 for expression control.

Note 3: Some GS-MIDI devices, in addition to the standard GS-MIDI sounds, have variations of some of the sounds. The way a particular variation of a GS-MIDI sound is activated is by send a Program Change message and a Bank Select message to the expander. When a MIDI-stop is activated, one of the messages sent to the expander MIDI OUT is the Bank Select message, which is two controller messages:

0BFh 00h *uu* 0BFh 20h *ll*

where *uu* and *ll* are the upper (most significant byte or MSB) and lower bytes (least significant byte or LSB) of the bank number, and 00h and 20h (32) are the controller numbers.

There are a few other MIDI-stop parameters that can be set which have an effect on the MIDI data that is sent out the expander MIDI OUT:

1. The pitch of the MIDI-stop can be transposed one or two octaves up or down, if desired.

The keyboard range over which the MIDI-stop works may be adjusted. This also allows two MIDI-stops on the same manual to be configured to play in different parts of the keyboard.

Full instructions on the use and configuration of all MIDI-stops provided on an organ, are given in the Classic Organist's Reference Manual.