FM MIDI Synthesizer driver

The FM MIDI Synth driver is a replacement driver for sound cards that implement FM synthesis. It features an extensive MIDI implementation as well as support for loadable instrument banks and a Control Panel applet for run-time configuration of driver parameters. It is designed to work with as many different sound cards from various manufacturers as possible.

Contents

Installation

The install instructions are grouped by category so you can go directly to an area for detailed information. If you have never installed a driver before, or don't remember how you did it, then you might want to use the **Browse Buttons** on the tool bar (**>>**) to move through each subject in order. The **Basic Driver Installation** section is included for those without prior experience.

If you *are* experienced at Driver installation, FM Synth is installed the same way all drivers are installed, so you may only need to check the **Configuration** section to be sure you know the necessary information about your particular sound card. Be sure to **remove your old driver**, even if it is an earlier version of FM Synth before installing the new one.

Basic Driver Installation

Drivers are specially written software that connect **applications** with **devices** such as printers, monitors, and sound cards. Windows maintains in the SYSTEM.INI file a list of drivers that you have specified should be loaded every time you start Windows. These drivers are almost always located in the **Windows\System** directory.

The installation of a driver, then, involves two steps. 1) Copying the necessary files to the **Windows\System** directory, and 2) Adding the names of the drivers to the list in the **SYSTEM.INI** file.

Because the correct procedure is critical to the success of this operation, Windows supplies the ability to automate this process through the use of the **Drivers** applet in the **Control Panel** application that comes with Windows. The **Control Panel** application is located by default in the **Main** program group in Program Manager. Double-clicking on the **Control Panel** icon brings up a Window with more icons. Double-clicking on the **Drivers** icon brings up the dialog box that shows you what optional drivers you already have installed, and provides you with the opportunity to remove a driver or add additional drivers.

If you elect to add a driver, the next box shows a list of drivers you can add. Note, however, that this list only includes drivers that originally shipped with the Windows installation disks. In almost all cases you will select the option to add an "Unlisted or updated driver". This will bring you to the last box, which asks you to provide the **location** of this new driver. This location can be either a **floppy disk** or a **directory** on your hard drive. In this location, along with the necessary driver(s), there must be a file called **OEMSETUP.INF**. Windows reads this file to determine the instructions that it must carry out to install the new driver. Frequently these instructions will involve obtaining more specific information from you.

After a driver has been installed, Windows must be **restarted** before you can use the new driver. This is because Windows only loads drivers when it starts up. The install process adds the driver to the list in the **SYSTEM.INI** file that Windows reads when it is loading.

Preparation

The first step is to unzip the FM Synth driver and related files to either a **diskette**, or a **directory** on your hard drive. The driver will be installed from this location. Be sure to note the disk and/or directory name, because you will have to provide this information when Windows asks you where the new driver is located

More hardware configuration parameters are provided than are available for most drivers.   In order to optimally configure the FM Synth driver, it is important to have prior knowledge of the hardware configuration of your sound card. You must know the address of the **I/O port** and you must know what **type of FM chip** is on your sound card. See the Configuration section for more information about how to determine these two items.

Since you will naturally be concerned that this new FM Synth driver won't work, (but relax) you should also make sure that you can return to your present configuration easily if you decide you want to. This involves locating the **driver** and its **OEMSETUP**.INF file that you are presently using. This may be on a disk supplied by the manufacturer of your sound card, or the manufacturer may have put these drivers in a directory on your hard disk. Soundblaster Windows drivers are located in the \win31 directory of the main Soundblaster directory, for example. When you have the floppy disk or you know where the installation directory is, it is a simple matter to restore your original setup.

File List

The following files must be present on the installation disk or in the installation directory:

FMTASK	EXE	This program loads the FM Synth Control Panel
FMSYNTH	DRV	This is the driver itself
FMCFG	DLL	This is the FM Synth Control Panel
OEMSETUP	INF	This is the file Windows reads to install FM Synth
ADDMAPFM	REC	This recorder macro automatically adds FM Synth to the MIDI Mapper if you so request.
ADDMAPFM	EXE	This program is required to start the recorder macro
FMSYNTH	HLP	This help file

All of these files except the OEMSETUP.INF file will be copied to the **Wiindows\System** directory and may be deleted from the installation directory after you have restarted Windows. You may want to keep them in the installation directory if you need to re-install FM Synth at a future date, however.

Driver Removal

Remove any existing FM driver. Any existing FM driver should be removed before installing a new driver. This is true even if you are installing a new version of the same driver, including a new version of FM Synth. While it is possible to install more than one FM driver, you must ensure that only **one** is in use at a given time. Unfortunately, this is not as easy to ensure as it sounds. We have found that some drivers under certain conditions may interfere with the operation of another driver even when they are not explicitly selected. Pay particular attention to the driver selected by the **MIDI Mapper** if you decide to load multiple drivers. Even if you are not specifically using the MIDI Mapper, you should configure the Mapper for the same driver you want to use with another program.

To **remove** a driver, open the **Drivers** applet in the Control Panel and select the FM driver to remove (this could be the Adlib driver, the SB Pro 2 FM Synth, the Voyetra OPL3 FM driver, or some other name). Only the FM driver needs to be removed, not the Wave, MIDI or Aux. driver. Press the **remove** button and answer **OK** to the dialog box which warns that the driver may be needed by the system, or asks you if you are sure. The driver is not actually deleted, in any case. It is merely removed from the load list in the SYSTEM.INI file.

Restart Windows. After removing a driver, you must restart or exit Windows before adding the FM Synth driver. This is because the driver has only been removed from the SYSTEM.INI file, and is still present in memory -- a new driver can not be loaded until the current one is removed from memory.

Driver Installation

Install the FM Synth driver. Open the **Drivers** applet from the Control Panel and press the **Add** button. Select **Unlisted or Updated Driver** from the list and press the OK button.   Type the full directory name, or floppy drive (ex: B:\ or C:\NEWDRV or wherever you put it) of the location where the driver resides into the dialog box and press **OK**. A dialog box containing the name: **Jamie O'Connell - FM Synth MIDI Driver** will be presented. Ensure that it is selected and press **OK**.

Next the FM Synth hardware configuration dialog is displayed. This dialog only needs to be filled out once, but the selections are critical to proper driver operation (the dialog can be opened again later, if required, by opening **Drivers**, selecting the **Jamie O'Connell - FM Synth MIDI Driver** and pressing the **Setup** button).

There is a **HELP** button in the setup box that will bring up this help file if you have any questions at this point.

Configuration

The FM Synth **Setup** dialog contains two sections: **I/O Port Address** and sound card **FM chip type**.

I/O Port Address

The I/O Port Address is the **physical port** of the FM sound chip. In some sound cards this may be different from the cards **base address**. For instance, the original Sound Blaster 1.5 has a base port address of 220 hex, but the FM chips are located at port address 228 hex. You would choose port 228 for this card.

FM Chip Type

The FM chip type depends on which card you posses. The earliest Adlib and Sound Blaster FM Cards contained a single Yamaha OPL2 sound chip. The first version of the Sound Blaster Pro added a second OPL2 chip. Early Media Vision Pro Audio Spectrum cards also have the 2 OPL2 configuration. Newer sound cards, including the SB Pro 2, SB16, and the MV PAS 16 use a single Yamaha OPL3 chip. The OPL3 contains the equivalent of 2 OPL2 chips plus some other features such as 4 Operator support. If your sound card is advertised as having a "20 voice, 4 operator" FM chip then you have the OPL-3.

Almost all sound cards implement Adlib compatibility: Port: 388 hex, 1 OPL2 chip, though choosing this setting may reduce the capabilities of the driver. (The Adlib is a mono only card capable of 6 melodic, and 5 percussive simultaneous voices. It can do 9 melodic voices when percussion is disabled.)

The FM Synth driver does not require knowledge of either the DMA or IRQ for the sound card, as these are only used for the digital sound, and MIDI portions of the cards. Following is a table of settings for some common sound cards with factory setup configurations. If your card is not listed, or has been changed from the factory setup, consult the user manual issued with the card:

Name	I/O Port (hex)	Chip Type
Adlib FM Card	388	1 OPL2
Sound Blaster 1.5	228	1 OPL2
Sound Blaster Pro (original)	220	2 OPL2
Sound Blaster Pro 2	220	1 OPL3
MV PAS (original)	388	2 OPL2
MV Pro Audio Spectrum 16	388	1 OPL3
FUSION 16	388	1 OPL3

If the Port address for your card is not listed in the dialog box, select **Other** and type the hexadecimal port address into the edit field. Next choose the chip type and press the **OK** button. If an FM chip is detected, the dialog box will close, and a **restart** warning is issued. Go ahead and restart Windows once again. If an FM chip is not detected, a message stating that the hardware settings on the card do not match the selected driver I/O Port address is issued. In this case, you must change the I/O Port Address setting -- as stated before, Port 388 can almost always be specified.

Optional MIDI Mapper Setup

After you choose to restart Windows, another dialog box will pop up offering you the opportunity to let the install process add **FM Synth** to your present **MIDI Mapper** setup. This is a non-destructive process that will preserve your present settings, if any, and add the FM Synth driver to them.

If you select **OK**, when Windows restarts, a **RECORDER** macro will run that will add FM Synth to your Mapper configuration.

If you choose not to run the macro, then see Using the FM Synth MIDI Driver to read about installing it yourself.

Adding an FM Synth Icon

After you have installed the FM Synth driver and restarted Windows, a new Control Panel applet is available for use: **FM Synth**. The FM Synth applet allows run-time configuration of the FM Synth MIDI driver. Open the Control Panel and then double-click on the new FM Synth icon.

It is also possible to create a regular program icon to start the FM Synth applet. When launched this way, you won't see the Control Panel, only the FM Synth applet. To do this, open the **Program Manager**, and then the Program Group where you want to store your FM Synth Icon. From the **FILE** menu, select **NEW** and then **New Program Item**. In the Description box, type the label you want to appear under the new icon. In the Command Line box type: **FMTASK.EXE**. You can skip the Working Directory, and can optionally add a shortcut-key sequence. An icon has been provided, but you can choose the Change Icon button to assign an icon of your choice. To launch FM Synth, double click on the icon, or use the shortcut-key if you assigned one.

Version 2.0 of FM Synth added a **Stay On Top** and a **Minimize** box. To toggle the **Stay On Top** option, open the System Menu (click the button in the top left corner) and select the item. When the **Stay On Top** item is checked it is in effect, and will cause the applet to remain on top of all other Windows. This allows you to keep the applet open and accessible when running a Sequencer or other Windows software. The **Stay On Top** is initially set to **On** (checked)

FMSYNTH.INI

The first time you run FM Synth it adds the **FMSYNTH.INI** file to your Windows directory. This allows FM Synth to start according to the customized options you have selected. These options are stored in the .ini file when you close the FM Synth applet with the Save Settings box checked on the Control Panel

If you are familiar with editing .ini files with Windows **Notepad** or a similar plain text editor, additional customization is possible.

If for some reason you do *not* want to load the FM Synth applet in the Control Panel, you can change this behavior by opening the **FMSYNTH.INI** file and in the **[applications]** section, add the line "LoadCPL=0". When you restart Windows, the FM Synth applet will not be loaded in the Control Panel. If you later decide you *do* want to load the applet, either delete the "LoadCPL=0" line, or replace the zero with a "1". This is an example of how the section of the fmsynth.ini file would look if you *didn't* want to load the applet:

```
[Application]
 LoadCPL=0
```

An ultra-efficient write scheme has been implemented in **version 2.1**. Since this has not been fully tested in a wide variety of machines, if you feel this may be causing problems it can be disabled by adding the following to the **Mode** section

```
[Mode]
RegisterShadow=0
```

Acknowledgment

I'd like to take this opportunity to acknowledge and thank some of the people that made this driver possible.   First of all, **Jim Maki** (Wiz-Op on the **Compuserve** MIDI forum) saw the need for a generic driver, and arranged for sponsorship.   **Microsoft Corporation** was kind enough to provide a copy of the Windows 3.1 DDK.   **Dan McKee** of **WinJammer Software Ltd**. (WinJammer Sequencer) has allowed use of their MMA ID to support the system exclusive portion of the driver.   **Greg Hendershott** of **Twelve Tone Systems** (Cakewalk Sequencer) has allowed use of their General MIDI timbre bank for the internal sounds. **Jerry Jorgenrud** expanded and enhanced this help file. Thanks to all.

Jamie O'Connell -- Boston 1993

Support

If you need help or wish to discuss any aspect of the driver and have a **Compuserve** account, go to the **Sight and Sound** forum (GO SSFORUM) or the **MIDI** forum (MIDIFORUM) and leave a message for Jamie O'Connell [72662,1433] or Sysop.

You may also use US Mail:

Jamie O'Connell
191 Park Drive #44
Boston, MA 02215

Using the FM Synth MIDI Driver

The FM Synth driver is loaded when Windows starts up. It can be used by any **Windows 3.1 MIDI** application that makes use of installable drivers. Check the application documentation for information about using drivers with that particular application.

An application shipped with **Windows 3.1** that can use the FM Synth driver indirectly is the **Media Player.** It requires the use of the dreaded **MIDI Mapper**, however. A **MIDI Mapper** setup for the FM Synth driver has not been included in this package, because installing it would overwrite any existing MIDI Mapper setup. We have, however, provided a macro which will automatically add a new Mapper setup via the use of the **Windows RECORDER** application (See Optional MIDI Mapper Setup). Setup automatically installs this file in your **Windows\System** directory as **ADDMAPFM.REC**. If this doesn't work properly for any reason, you can manually add a new setup without too much frustration: Open the **MIDI Mapper**, press **New**, type:

Name: FM Synth All
Description: FM Synth Driver

Press **OK**. Select the **Jamie O'Connell - FM Synth MIDI Driver** for channels 1-16 under the Port assignments, and answer **OK** to saving the setup. You can then use the FM Synth Control Panel applet to decide which channels to actually use.

FM Synth MIDI Control

Open the FM Synth applet either from the Control Panel or from the icon you (optionally) created. You can change various parameters without having to restart Windows. Certain changes made will cause the driver to be *reset* though, so you probably don't want to do this while a song is playing. Changes take effect after you push the **Apply** button.

Controls

Stereo Mode

The original SB cards (and their clones) only had a single Mono **OPL2** chip. The next generation (SB Pro) had 2 **OPL2** chips: one chip mapped to the left speaker and one chip mapped to the right speaker. Smooth panning between the left and right speakers is possible by assigning the same timbre to a voice on both the left and right chips and adjusting the volume sent to each -- this is true stereo, but at the expense of cutting the polyphony in half.

The current generation (SB Pro 2, SB16) has a single **OPL3** chip. Any voice on this chip can be assigned full left, center or full right. We can achieve a "coarse" panning (Stereo Button UNCHECKED) without losing polyphony, by adjusting the timbre to one of the 3 positions. If you DO select the **Stereo** option we do the smooth panning (described above) by assigning the same timbre to 2 voices -- one full left and the other full right, but the effective polyphony is cut in half. On an **OPL3** card, **stereo mode** should be used sparingly. If you experience **clicking** during a song, it may be the result of the driver running out of voices, and stealing a playing voice to play the next required note. The driver uses a dynamic voice allocation algorithm which will steal the oldest voice playing the same timbre, when available voices are exhausted. You can increase the number of available voices by un-checking stereo. The number of available voices is calculated and displayed in the top of the dialog box.

Percussion Mode

The percussion mode check box enables and disables percussion mode. For songs that contain no drums (or if you have a separate drum machine), un-checking percussion provides an additional 3 melodic voices. When percussion mode is on, you can further select the MIDI channel that the drums will respond to -- change it by entering a different percussion channel number.

Version 2.0 of the driver introduced a **Percussion Boost** option. This parameter affects the apparent level of the percussion section against the melodic voices. It is initially set to a value of 10, but may be changed to values ranging from -64 to +63. 0 means No Boost; negative values decrease the percussion level (attenuation).

Reset Driver

The Reset driver button performs a complete **initialization** sequence on the card. It is intended to be used in emergencies -- sort of a Panic button.

Vibrato and Tremolo Depth

The vibrato and tremolo depth on the FM card can each be set to light or deep by checking or un-checking the appropriate box. The FM Synth driver is unique in providing adjustable vibrato and tremolo control through the **modulation wheel controller**. The check boxes allow adjustment of the overall effect, although the effect is rather subtle.

Bend Range

Bend sensitivity (the number of tones that the bend controller covers) is adjustable. It can be set to a value of 0 to 24 semitones (two octaves). This can be quite useful in the case of instruments with fixed bend sensitivities (i.e. guitar controller).

Bend Sensitivity is stored separately for each **channel**: the number displayed in the control window is the Range which **most** channels are set to. However, if you change the range in the control window the value is applied to **ALL** channels. You can set **individual channel** bend sensitivities via MIDI messages. See RPN LSB

Channel Map

The driver can be set to respond to any or all of the 16 MIDI channels. By default, **all** boxes are checked (all channels active), but you may prevent certain MIDI channels from being played by un-checking the corresponding box. This can be very handy when trying to play MIDI files that conform to Microsoft's standard. These MIDI files, which are often contained in only one MIDI file track, contain 2 different arrangements of the same song. To hear one arrangement only, check boxes 13-16 only. Check boxes 1-10 only, to hear the other arrangement.

Timbre Banks

The FM Synth driver contains an **internal** timbre bank, which is initially mapped into all 5 bank positions. This internal bank conforms to the **General MIDI** standard for instruments and their assigned locations. Any (or all) of the banks can be loaded from **external** IBK timbre bank files. To load an **external** bank, select the Bank (1-5) and press **browse**. A file open dialog allows you to select the IBK timbre bank to use. The path name of the bank file will be displayed below the Timbre box. These settings are remembered over Window sessions, so the bank will continue to be loaded in the future. To revert to the **internal** bank file, select the bank number and press **remove**. The banks are actually switched upon reception of a MIDI Bank Change. IBK bank files may be created or modified by using the shareware **SBTimbre** program, or converted to by use of the freeware **SBANK** utility. Both utilities are available on Compuserve (**Sight & Sound** and **MIDI** forums) as well as other BBS's.

If you check the **Current** check box, the currently highlighted bank becomes the default: all channels begin playing timbres from the selected bank. A channel's default bank is overriden after a MIDI bank change message is received. Alternatively, you can determine the current default bank by scrolling through the bank list: the **Current** check box will be selected for the default bank.

Version 2.0 of the driver added the capability of loading a **Percussion IBK Bank**. The percussion bank is loaded in the same fashion as the melodic banks. There is only **one** percussion bank, consequently, it is always the **current** bank. If specified, it replaces the internal percussion bank. You can revert to the internal bank by **removing** the bank selection.

Save Settings

If the Save Settings check box is checked, the current settings are saved permanently to the **FMSYNTH.INI** when the applet is closed.

MIDI Implementation

The following MIDI Messages are supported:

1) Note On
2) Note Off
3) Polyphonic key pressure/Aftertouch
4) Control Change

0	Bank Change (MSB Only)
1	Modulation Wheel (0-63 off, 64-127 on)
6	Data Entry MSB
7	Volume Control
10	Pan Control
38	Data Entry LSB
64	Sustain Pedal
92	Tremolo Depth
96	Data Increment
97	Data decrement
100	RPN LSB
101	RPN MSB
121	Reset All Controllers
123	All Notes off

5) Patch Change
6) Channel Pressure/Aftertouch
7) Pitch Bend
8) System Exclusive

Control Change

Control changes are MIDI messages that you can embed in a MIDI file to change the behavior of FM Synth. The specification that controls what numbers are assigned to what controllers is part of the MIDI 1.0 Specification. The full specification can be obtained from the:

International MIDI Association
23634 Emelita St.
Woodland Hills, CA 91367

(818) 598-0088

Not all synths support all of the controllers, nor is this required. But **if** a control is implemented then the controller number is determined by the specification. In addition, not all controller numbers are specifically assigned, meaning that a manufacturer can implement these unassigned controllers for purposes specific to that manufacturer. If you play a MIDI file that has a controller message that your synth does not support, the message is simply ignored. This list shows the controller messages that the FM Synth driver **will** respond to.

To use this feature, set the controller number specified in the list that corresponds to what you want to do. Then set the value of that controller.

Bank Change

For the Bank Change message, only the Most Significant Byte (Controller 0) is looked at. The LSB (controller 32) is ignored. In an attempt to conform to several bank change schemes, the following mapping is implemented:

Name	Controller 0	Ranges
Bank1	0	5-15; 80-127
Bank2	1	16-31
Bank3	2	32-47
Bank4	3	48-63
Bank5	4	64-79

Following a bank change you must immediately (one tick later is the convention) send a patch change to use a timbre from the new bank. **Note** that the bank change/patch change **only** affects the **channel** the change was sent on. The bank change remains in effect on that **channel** until another bank change is received or the driver is **Reset**. A MIDI song that switches banks with embedded Controller 0 messages can leave the driver set up incorrectly for the next song. If a song played correctly in the past but now is not, suspect that an earlier song changed the bank selection, and Reset the driver to load the correct patches.

Pan Control

As previously mentioned in the Stereo Mode topic, the sound cards support both smooth and coarse stereo panning. This is enabled by sending Controller 10 messages with values ranging from 0 for full left to 127 for full right. A value of 64 is centered. In coarse panning, the choices are only full left, centered, or full right. Other values are rounded up or down to the nearest appropriate number, i.e. 0, 64, or 127.

Tremolo Depth

Tremolo depth is a rather subtle effect, but 4 ranges correspond to the master Vibrato and Tremolo bit settings on the card:

Controller Value	**Tremolo on Card**	**Vibrato on Card**
0-31	light	light
32-63	light	deep
64-95	deep	light
96-127	deep	deep

RPN LSB

RPN stands for Registered Parameter Number. Various synths respond to various parameter numbers. FM Synth implements **Bend Range** and **Tuning Sensitivity.**

To change one of these parameters you have to send several MIDI messages. First, you notify the synth what parameter you want to change. This requires sending two controller messages, controller 100 **(RPN LSB) and** 101 **(RPN MSB).** To signify a **Bend Range** message, use values of 0 in both controller messages. To signify **Tuning Sensitivity**, send values of 1 in Controller 100 and 0 in Controller 101.

Once you have specified the parameter you want to change, you have to send the value to change to. The values are sent in Controller messages 6 **(Data Entry MSB)** and 38 **(Data Entry LSB)**. Values of 0, 0 would set the bend range to 0. For each number increase in the Controller 6 value, the bend range increases by one semitone. Sending it a value of 12, 0, then, would raise the bend range one octave (12 semitones). Since Bend sensitivity is a channel message, FM Synth will support a different Bend Range on each of the 16 MIDI channels (excluding the percussion channel).

The procedure to change the Bend Range on Channel 1 by one octave, for example, is:

Channel	**Controller**	**Number**	**Value**
1	RPN LSB	100	0
1	RPN MSB	101	0
1	Data Entry MSB	6	12
1	Data Entry LSB	38	0

Tuning Sensitivity is used to tune the synth to the pitch level you prefer. For standard A=440Hz tuning, the value for Controller 6 should be 64. Each increment/decrement from 64 changes the pitch approximately 1/100 of a semitone. Since you are limited to the value range 0-127, the lower and upper limits are approximately 416Hz and 462Hz. Two popular historical tunings are A=430Hz (Controller 6 value = 36) and A=460Hz (Controller 6 = 115)

When you have specified an **RPN**, that parameter stays selected until a different one is entered. This means that once you have the initial data entered for an **RPN**, you can send Controller messages 96 **(Data Increment)** or 97 **(Data Decrement)** to increment or decrement that setting. To change Bend Range, for example, once the **RPN** has been entered and set, say to 12, you can change it to 10 by sending a single Controller 97 message with a value of 2.

Patch Change

The instrument patches played by the FM chip on a sound card are stored in the driver, not in the chip. You can change the type of instrument playing on any MIDI channel, except the percussion channel, by sending a **patch change** to the driver on that channel. Note that the patch change sent to the FM Synth driver is a **number,** in a range from 0-127, that represents to the driver a **location** in memory. What **instrument** actually is in that location depends on the Patch Bank that is currently selected.

The **General MIDI** standard attaches names to the patch numbers in an attempt to ensure that a patch change of 0, for example, is a Grand Piano no matter what synth it is sent to. This standard makes sharing MIDI files easier. The internal patches for the FM Synth conform to the **General MIDI** standard. The best procedure, then, is to leave the internal patches selected on the first bank, Bank 0. That way the instruments will be in their proper locations for any song that was created according to the **General MIDI** standard.

However, if you are making MIDI songs to play only on your own equipment, you can store any instrument you care to in each location through the use of custom .ibk instrument files. But then the patch changes in your MIDI song may not represent the instruments you had in mind if someone else plays the song with a **General MIDI** setup.

System Exclusive

The FM Synth driver is a **bi-directional** driver; it provides a MIDI input port in order to implement System Exclusive dumps. The driver can send or receive System Exclusive data and receive commands. You need to use a sequencer that is capable of sending and receiving SysEx information in order to use this feature. If you send a **dump request** from your sequencer to the FM Synth driver it sends back its current status. You can then edit this information and send the new SysEx data back to the driver. The general format of a SysEx message is as follows:

F0	MIDI SysEx Identifier
00, 00, 5B	Manufacturers' ID: WinJammer Software Ltd.
7F	Device ID -- Must be present
01	Model ID -- FM Synth Driver
xx	Command ID -- 11 Dump request, 12 Send request, 16 Reset
aa	Address MSB -- Location: System, Working storage, or Bank#
aa	Address Channel# or Timbre#
ss	Size MSB of following Data Block or #Bytes to dump
ss	Size LSB
data	Data if send request, otherwise blank
F7	End of Exclusive Command

Details

Command ID
Address and Size Bytes
Timbre Data
Timbre Exclusive Format
Percussion Timbres and Map
System parameter Exclusive Format
Examples

Command ID

Code (Hex)	Meaning
11	Request Data from Driver
12	Send Data to the Driver, Send Data to the Application
15	Reset Driver to Power-on state
16	Reset Timbre Banks

In order to receive system exclusive dumps from the driver, the FM Synth SysEx Input driver must be attached to an application **input** port. It is then a simple matter of sending the desired starting address, and a size describing the amount of data to dump. The data received from the driver contains a command id of 12, so that the data is in the proper format to be shipped back to the driver at any time.

The Reset Timbre Bank message and the Reset Driver message contain only a single Command ID code -- no address or size bytes.

Address and Size Bytes

Address Bytes

The address bytes together determine where data will be stored. See the Timbre Data section for specific locations.

Size Bytes

The total size of the data block is determined by combining the size bytes (7 bits of each). Data is stored or dumped consecutively in Address order -- this means that you can fill or dump the entire contents of the driver with one data block.

Timbre Data

Individual timbres or entire banks may be uploaded to the driver. There are three groups of data: standard melodic timbres, percussion timbres and map, and system parameters. Most functions available using the Control Panel applet, are also available through a System Exclusive message.

The new timbre parameters remain until the driver is reset. You can send each group SysEx block as if the Banks and timbres were stored consecutively. Individual parameters can be written to by placing a value in the address offset (LSB).The new timbre will not take effect until a program change is made to access the timbre.

For **Version 2.0** of the driver 2 addresses have been added: 12 for the new percussion bank, and 22 for extended system parameters. All the older addresses are still supported.

The original percussion bank only held 47 drum keys, as defined by **General MIDI**. The new percussion bank is a full 128 timbre IBK bank. It maps the entire keyboard across the bank.

Addr MSB	**Addr LSB**	**Description**
0	0	Bank 0, Timbre 0 (first bank, first timbre)
0	1	This refers to the second timbre in the first bank
...	...	... The rest of the bank
0	7F	The last timbre in the first bank
1	0	The first timbre in the second bank
...	...	... The remaining timbres
	7F	The last timbre in the fifth bank
		Percussion Parameters
10	0	First Percussion timbre
10	2E	Last Percussion timbre (0-46)
11	0	Percussion Map -- Size: 94 bytes
12	0	**Version 2.0** Percussion timbre 0
12	7F	**Version 2.0** Percussion timbre 127
		System Parameters
20	0	Location of theSystem Parameters -- Size: 24 Bytes
22	0	Location of **Vers 2.1** System Parameters--Size 27 Bytes

Timbre Exclusive Format

Offset	Description	Range
1	Modulator Attack Rate	0-15
2	Modulator Decay Rate	0-15
3	Modulator Sustain Rate	0-15 (See note below)
4	Modulator Release Rate	0-15
5	Modulator Tremolo, Vibrato, Sustain, Envelope Scale	4 lower bits, on=1 or off=0: |...|AM|VIB|EGT|KSR| (See chart)
6	Modulator Frequency Multiplier	0-15 (0 = 1/2)
7	Modulator Keyboard Scaling Level	KSL: 0-3
8	Modulator Output Level (attenuation)	0-63 (See note below)
9	Modulator Wave Form	Wave: 0-7 (0-3 on OPL2)
10	Modulator Feedback, Operator Connection (FM or Additive)	|...|F|F|F|C| ; FB: 0-7; FM: 0, Add Syn: 1 (See chart)
11	Carrier Attack Rate	0-15
12	Carrier Decay Rate	0-15
13	Carrier Sustain Rate	0-15 (See note below)
14	Carrier Release Rate	0-15
15	Carrier Tremolo, Vibrato, Sustain, Envelope Scale	4 lower bits, on=1 or off=0: |....|AM|VIB|EGT|KSR| (See chart)
16	Carrier Frequency Multiplier	0-15 (0 = 1/2)
17	Carrier Keyboard Scaling Level	KSL: 0-3
18	Carrier Output Level (attenuation)	0-63 (See note below)
19	Carrier Wave Form	Wave: 0-7 (0-3 on OPL2)
20	Percussion Voice#	6=BD, 7=SD, 8=TT, 9=CYM, 10=HH
21	Transposition Offset	|...|S|TTTTT| S=0 pos,1 neg;Trans: 0-31
22	Note # driver plays if percussion	New in Version 2.0
23	Reserved	Set to 0 -- Future Use
24	Reserved	Set to 0 -- Future Use

Each timbre is 24 bytes long.   Each bank holds 128 timbres.   There are 3,072 bytes in an entire Bank message (plus header). All five banks can be filled by one 15,360 byte message (plus header).

Note: The actual number stored for the modulator and carrier sustain and output levels is an attenuation level. It is the number you would subtract from the maximum to obtain the "real" number. For example, the maximum number for the output level is "63". If the output level is at its maximum, then the number stored is "0"

Chart for offset 5 and 15: (An X under a column means the parameter is selected, a O that it is not)

HexNo.	Tremolo	Vibrato	Sustain	Envelope
0	O	O	O	O
1	O	O	O	X
2	O	O	X	O
3	O	O	X	X
4	O	X	O	O
5	O	X	O	X
6	O	X	X	O
7	O	X	X	X
8	X	O	O	O
9	X	O	O	X
0A	X	O	X	O

0B	X	O	X	X
0C	X	X	O	O
0D	X	X	O	X
0E	X	X	X	O
0F	X	X	X	X

Chart for offset 10: Feedback and Synthesis

Hex	**Mod Feedback**	**FM Synthesis**	**Add Synthesis**
No.			
0	0	X	O
1	0	O	X
2	1	X	O
3	1	O	X
4	2	X	O
5	2	O	X
6	3	X	O
7	3	O	X
8	4	X	O
9	4	O	X
0A	5	X	O
0B	5	O	X
0C	6	X	O
0D	6	O	X
0E	7	X	O
0F	7	O	X

Example: This is the SysEx dump for the internal piano sound stored in the first timbre position:

Hex	**Parameter**	**Meaning**
No		
0F	Modulator Attack Rate	15
2	Modulator Decay Rate	2
5	Modulator Sustain Rate	10
2	Modulator Release Rate	2
2	Modulator	Tremolo=OFF,Vibrato=OFF,Sustain=ON,EnvScale=OFF
1	Modulator Frequency Multiplier	1
1	Modulator KSL	1
0F	Modulator Output Level	48
0	Modulator Wave Form	0
6	Modulator FB, Op Connection	Feedback=3, FM Synthesis
0D	Carrier Attack Rate	13
2	Carrier Decay Rate	2
7	Carrier Sustain Rate	8
3	Carrier Release Rate	3
3	Carrier	Tremolo=OFF,Vibrato=OFF,Sustain=ON,EnvScale=ON
1	Carrier Frequency Multiplier	1
0	Carrier KSL	0
0	Carrier Output Level	63
0	Carrier Wave Form	0
0	Percussion Voice#	Not a Percussion patch
0	Transposition Offset	0
0	Note # driver plays if perc.	0
0	Reserved	Set to 0 -- Future Use

0	Reserved	Set to 0 -- Future Use

Percussion Timbres and Map

General MIDI defines 47 drum notes, and the driver has storage for 47 drum timbres.The percussion timbres use the percussion voice# parameter to decide which of the 5 drum voices to sound for the drum tone. In addition, a percussion mapping of Key to Timbre is implemented, and may be dumped and loaded. There are 47 slots in the percussion map, which cover MIDI notes 35 to 81 (in other words, array element 0 corresponds to MIDI note 35). Each slot is 2 bytes, so the map is 94 bytes long:

Percussion Map Slot

Offset	**Meaning**
1	Timbre number to sound (0-46)
2	MIDI Note to play for the key (60= Middle C)

NOTE For **Version 2.0** of the driver the above bank and map are still supported. However, the percussion bank has been extended to a full 128 timbre IBK bank. Consequently the original 47 note mapping falls within the middle of the full bank. The new percussion bank contains the information which used to be stored in the map within the timbre.

System Parameter Exclusive Format

Certain driver configuration parameters may be sent to or received from the driver. Following is the format for the system parameter dump and load:

Offset	**Description**	**Range**
0	Stereo Mode	0 = Mono, 1 = Stereo
1	Percussion Mode	0 = Off, 1 = On
2	Percussion Channel	channel = 0-15 (i.e. MIDI channel 10 is value 9)
3	Default Bank	0-4 (Bank1 - Bank5)
4	Vibrato Depth	0 = light, 1 = deep
5	Tremolo Depth	0 = light, 1 = deep
6	Bend Range	0-12 semitones (1 octave)
7	Save these settings	1 = Save the settings in the INI file
8	Channel 0	MIDI Channel Map (0 = Off, 1 = On)
::	::	The channel map specifies which channels to receive on.
23	Channel 15	0 = Off, 1 = On
	For SysEx Address 22: all the above are transmitted and received plus:	
24	Percussion Boost	0 to 127 maps into values -64 to +63
25	Parallel Pitched Percussion	0=Off, 1=On
26	Register Shadowing	0=Off, 1=On

Examples

The following message will send a timbre to the first bank, timbre slot 12 (Vibraphone). All numbers are in hexadecimal.
The message would be sent in a string; the formatting below is just for purposes of explanation.

F0, 00, 00, 5B, 7F, 01,	Header
12,	Send Request
00, 0B,	Bank#1, Timbre#12
00, 18,	Size: 24
0A, 0A, 01, 02, 09, 03, 02, 17, 02, 0E,	Data:Modulator Parameters
0A, 0C, 02, 01, 09, 01, 00, 00, 00,	Carrier Parameters
00,00, 00, 00, 00,	Perc., etc. (See explanation of timbre format)
F7	End of Exclusive

This message requests the driver to send timbre # 13, 14, and 15 from the 2nd Bank

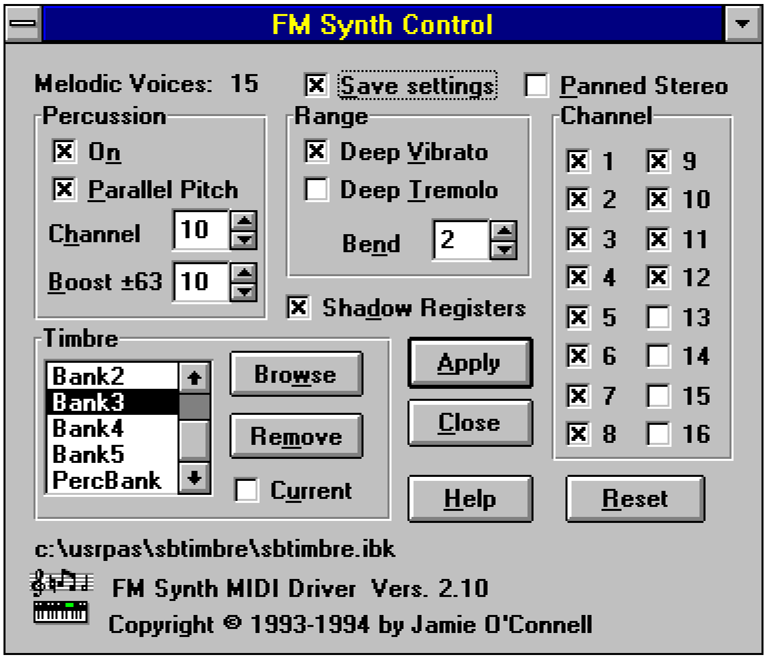
F0, 00 , 00 , 5B, 7F, 01,	Header
11,	Dump Request
01, 0C,	Bank#2, Timbre#13
00, 48,	Size: 72 (3 timbres X 24 bytes/timbre)
F7	End of Exclusive

The next message requests the driver to reset all the patch banks to the startup state:

F0, 00 , 00 , 5B, 7F, 01,	Header
16,	Reset Request
F7	End of Exclusive

Control Panel

Click with the **mouse** on the control that you would like more information about, or use the **TAB** key to cycle through the controls and press ENTER when the control is highlighted.



Melodic Voices: 15

☒ Save settings

☐ Panned Stereo

Percussion

☒ On

☒ Parallel Pitch

Channel 10

Boost ±63 10

Range

☒ Deep Vibrato

☐ Deep Tremolo

Bend 2

☒ Shadow Registers

Channel

☒ 1 ☒ 9
☒ 2 ☒ 10
☒ 3 ☒ 11
☒ 4 ☒ 12
☒ 5 ☐ 13
☒ 6 ☐ 14
☒ 7 ☐ 15
☒ 8 ☐ 16

Timbre

Bank2
Bank3
Bank4
Bank5
PercBank

☐ Current

c:\usrpas\sbtimbre\sbtimbre.ibk

FM Synth MIDI Driver Vers. 2.10

Copyright © 1993-1994 by Jamie O'Connell

FM Synth Version 2.10

MSB and LSB are terms used to define a large 14 bit word made from two 8 bit bytes. Only the lower 7 bits from each byte are used. Thus, the largest value that an MSB or LSB can have is 7F(hex), rather than the FF(hex) that a normal byte can have.

the maximum available simultaneous voices

The number shows how many melodic voices, or timbres, can play at one time. It changes as various other items are selected to reflect the maximum polyphony.

Checking this box selects fully panning stereo mode. It is recommended that this box remain unchecked, as it cuts the available polyphony in half when selected.

Checking this box turns on percussion mode. If it is unchecked, three more melodic voices become available.

This box shows what midi channel the percussion tracks will play on. A different channel may be selected. The General MIDI specification states that channel 10 be used for percussion.

This box controls how much the volume of the percussion channel is boosted or attenuated relative to the melodic voices.

Selecting this box turns on deep vibrato. It is a rather subtle effect.

Selecting this box turns on deep tremolo. It is a rather subtle effect.

This box selects how many semitones a note can be bent with the pitch bend controller. It has a range of 0 to 24 half steps.

A check mark in the channel box shows which channel numbers are processed by the FM Synth. You may sometimes wish to unselect channels 13-16 for MS style "dual" arrangement MIDI files.

This list shows the five banks and one percussion bank that can be loaded into the driver. Selecting a bank causes information to be displayed below the box: either the word <internal> or the path name of the .ibk file assigned to that bank. To make that selected bank the active bank, the "Current" box must then be checked. If the bank you select is already the current bank, then an "X" will appear in the "Current" box.

Pushing this button causes a file list box to open from which you can select an .ibk timbre file to be loaded into the bank that is selected.

When you select a bank in the list box and push the *Remove* button, the external .ibk file assigned to that bank becomes unassigned, and the internal patches are used.

This box shows if the selected bank in the list box is the currently active bank. If it is not, clicking this box will make it the currently active bank.

When you push *Apply*, all of the settings showing on the control panel become the active settings.

Pushing *Close* shuts down the FM Synth control Panel. It does NOT *apply* the changes before closing, so you may wish to do this before *closing.*

Help brings up this help file.

Selecting this box causes the current settings to be saved in the FMSYNTH.INI file in the Windows directory, when the *Apply* button is pressed. These settings will then be the default settings the next time you open FM Synth.

Pushing this button resets the FM Synthesizer on the sound card to its startup condition. It can be used as a *Panic* button when things go wrong.

This line displays either the word, **<internal>**, or the path name of the .**IBK** file assigned to the selected bank.

When selected, this causes FM Synth to use an alternate form of choosing percussion instrument pitches: only the Bass Drum, and the Tom-Tom instruments are allowed to freely change pitch. The Snare Drum is always pitched 7 semitones above the current Tom-Tom pitch, and the Cymbal and Hi-hat follow the others. The Cymbal and Snare Drum share one frequency parameter, and the Hi-Hat and Tom-Tom share another.

When the checkbox is not selected the percussion instruments are free pitched: the specified frequency is obtained from the timbre parameters.

When this is selected, an image of the card's registers is maintained in RAM. In the case of any card I/O register output, the memory image is examined, and the I/O only takes place if the memory image is different, making the driver more efficient.

Because another program could possibly perform direct I/O writes to the card, and confuse FM Synth, this parameter is selectable: if it is off all I/O is sent to the card.