

## **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](#)

[Configuration](#)

[Acknowledgment](#)

[Support](#)

[Using the FM MIDI Synth Driver](#)

[FM MIDI Synth Control](#)

[MIDI Implementation](#)

[System Exclusive](#)

## Installation

The first step is to unzip the FM Synth driver and related files (including the one you are reading) to either a diskette, or a directory on your hard drive. The driver will be installed from this location (make note of the disk and/or directory name).

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.

The FM Synth driver is installed via the Windows Control Panel **Drivers** applet. The following steps are performed when installing a new driver:

**1) Remove any existing FM driver.** It is recommended that any existing FM driver be removed before installation is attempted. While it is possible to install more than one FM driver, you must ensure that only one is in use at a given time. Note: Only the FM driver should need removal, not the Wave, MIDI or Aux. driver. To **remove** a driver, open the **Drivers** applet, 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). Press the **remove** button and answer **OK** to the dialog box which warns that the driver may be needed by the system.

**2) 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.

**3) 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**.

**Note:** If you receive a message stating that the **CTL3DV2.DLL** driver is already on the system, or a message stating that the driver is in use (when you know you've removed it and restarted), don't despair! The message simply refers to the fact that some other application is currently using the **CTL3DV2.DLL** file. The driver has most likely been successfully installed: don't restart yet. Repeat the 3rd installation step, but this time select **Current**, when asked whether to use **New** or **Current**.

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 FM MIDI Synth and pressing the **Setup** button).

**Note:** When re-installing a newer version of the FM Synth driver, you must first remove the old one as detailed above.

[Configuration](#)  
[Contents](#)

## 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 possess. 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, 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. 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 9 melodic, and 5 percussive simultaneous voices).

It should be noted that 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

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.

## Using the FM MIDI Synth 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 loading drivers.

An application shipped with Windows 3.1 and 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. You may create a new Mapper setup without too much frustration: Open the MIDI Mapper, press **New**, Select FM Synth Driver for channels 1-16, and save the setup under a new name. You can then use the FM Synth Control Panel applet to decide which channels to actually use.

## FM MIDI Synth Control

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 MIDI Synth driver -- 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 choose **OK** to close the applet, except for the **Reset Driver** button which takes effect immediately. Open the FM Synth applet by double-clicking on it.

### **Controls**

[Stereo Mode](#)

[Percussion Mode](#)

[Reset Driver](#)

[Vibrato & Tremolo Depth](#)

[Bend Range](#)

[Channel Map](#)

[Timbre Banks](#)

[Save Settings](#)

## Stereo Mode

Checking the **Stereo** box will allow smooth panning of voices from left to right on a 2 OPL2 or 1 OPL3 card, but available polyphony (the maximum available simultaneous voices) will be cut in half. On an OPL3 card, stereo mode should be used sparingly -- the OPL3 always supports a coarse *left-center-right* stereo placement of voices, even when stereo mode is unchecked. 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.

## 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 unchecking 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 12 semitones (one octave). This can be quite useful in the case of instruments with fixed bend sensitivities (i.e. guitar controller).

## 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. 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 select message (see below). 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 overridden 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.

## **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) Control Change
  - 0 Bank Change (MSB Only)
    - Bank1: 0, 5-15, 80-127
    - Bank2: 1, 16-31
    - Bank3: 2, 32-47
    - Bank4: 3, 48-63
    - Bank5: 4, 64-79
  - 1 Modulation Wheel (0-63 off, 64-127 on)
  - 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 (Tuning, Bend Sensitivity)
  - 101 RPN MSB
  - 121 Reset All Controllers
  - 123 All Notes off
- 4) Program Change
- 5) Pitch Bend
- 6) System Exclusive

## Control Change Notes

### Bank Change Message

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	

### 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

### 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. The general format of a SysEx message is as follows:

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

#### Details

Command ID

Address and Size Bytes

Timbre Data

Timbre Exclusive Format

Percussion Timbres

System Parameter 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 Bytes**

The address bytes, together determine where data will be stored. See below 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.

Addr MSB	Addr LSB	Description
00	00	Bank 0, Timbre 0 -- this refers to the first timbre in the first bank -- each timbre is 20 bytes long (see below for format)
00	01	This refers to the second timbre in the first bank
...	...	... The rest of the bank
00	7F	The last timbre in the first bank
01	00	The first timbre in the second bank
...	...	... The remaining timbres
04	7F	The last timbre in the fifth bank
<b>Percussion Parameters</b>		
10	00	First Percussion timbre
10	2E	Last Percussion timbre (0-46)
11	00	Percussion Map -- Size: 94 bytes
<b>System Parameters</b>		
20	00	Location of the System Parameters -- Size: 24 Bytes

## Timbre Exclusive Format

Offset	Description	Allowed Values
01	Modulator Attack Rate	0 - 15 higher is faster
02	Modulator Decay Rate	0-15
03	Modulator Sustain Rate	0-15
04	Modulator Release Rate	0-15
05	Modulator Trem,Vib,EnvScale,KSR	4 lower bits indicating on or off:  ... AM VIB EGT KSR
06	Modulator Frequency Multiplier	0-15 (0 = 1/2)
07	Modulator KSL	KSL: 0-3
08	Modulator Output Level (attenuation)	0-63 zero is loudest
09	Modulator Wave Form	Wave: 0-7 (0-3 on OPL2)
10	Modulator Feedback, Op Connection	... F F F C  - FB: 0-7, FM: 0, Add Syn: 1
11	Carrier Attack Rate	0 - 15 higher is faster
12	Carrier Decay Rate	0-15
13	Carrier Sustain Rate	0-15
14	Carrier Release Rate	0-15
15	Carrier Trem,Vib,EnvScale,KSR	4 lower bits indicating on or off:  ... AM VIB EGT KSR
16	Carrier Frequency Multiplier	0-15 (0 = 1/2)
17	Carrier KSL	KSL: 0-3

18	Carrier Output Level (attenuation)	0-63 zero is loudest
19	Carrier Wave Form	Wave: 0-7 (0-3 on OPL2)
20	Percussion Voice#	Used for Percussion patches: 6=BD, 7=SD, 8=TT, 9=CYM, 10=HH
21	Transposition Offset	... S TTTTT  S=0 pos,1 neg;Trans: 0-31
22	Reserved	Set to 0 -- Future Use
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).

## 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:

<b>Offset</b>	<b>Description</b>	<b>Allowed Values</b>
01	Stereo Mode	0 = Mono, 1 = Stereo
02	Percussion Mode	0 = Off, 1 = On
03	Percussion Channel	if Percussion mode = 1: channel = 0-15 (i.e. MIDI channel 10 is value 9)
04	Default Bank	0-4 (Bank1 - Bank5)
05	Vibrato Depth	0 = light, 1 = deep
06	Tremolo Depth	0 = light, 1 = deep
07	Bend Range	0-12 semitones (1 octave)
08	Save these settings	1 = Save the settings in the INI file
09	Channel 0 -- MIDI Channel Map	0 = Off, 1 = On
:	:	The channel map specifies which channels
:	:	to receive on.
24	Channel 15	0 = Off, 1 = On

## 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:

<b>Offset</b>	<b>Percussion Map Slot Meaning</b>
<b>01</b>	Timbre number to sound (0-46)
<b>02</b>	MIDI Note to play for the key (60 = Middle C)

## Examples

The following message will send a timbre to the first bank, timbre slot 12 (Vibraphone). All numbers are in hexadecimal:

<b>F0, 00, 00, 5B, 7F, 01, 12</b>	Header & Send Request
<b>00, 0B, 00, 28</b>	Bank#1, Timbre#12, Size: 40
<b>0F, 0B, 0C, 01, 06, 03, 01, 01, 03, 07</b>	Data: Operator Parameters
<b>0D, 0A, 0C, 02, 07, 02, 01, 03, 02, 00</b>	
<b>1F, 2A, 0B, 00, 06, 03, 01, 02, 02, 05</b>	Data: 2nd Timbre Operator Parameters
<b>03, 0D, 0C, 01, 06, 00, 00, 03, 03, 00</b>	
<b>F7</b>	End of Exclusive

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

<b>F0, 00, 00, 5B, 7F, 01, 11,</b>	Header & Dump Request
<b>01, 0C, 00, 3C, F7</b>	Bank#2, Timbre#14, Size: 60

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

<b>F0, 00, 00, 5B, 7F, 01, 16, F7</b>	Header & Reset Request
---------------------------------------	------------------------

## 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. 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 [73030,351] or Sysop.

You may also use US Mail:

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



