

Jeff Cazal MIDI Software

Power User Information

IMPORTANT NOTE: Reading this file is completely OPTIONAL! You do not have to read nor understand this information to run either the *Rhodes Warrior* or the *SongCanvas* editors!

Overview

This document contains detailed technical information to help "power users" get the most out of Jeff Cazal MIDI programs. This information applies to the DOS versions of both the *Rhodes Warrior* and the *SongCanvas* editor/librarian programs..

This document discusses:

- General DOS optimization
- Running under *Windows 3.1* • and coexisting with MME programs (e.g., *Wincake* and *WinJammer*)
- Running under *DESQview 2.4*
- Complete command-line options for *Play/D*, the MIDI engine

General/DOS

The start-up batch file loads *Play/D*, runs the editor, unloads *Play/D*, and runs "EXIT". This batch file will thus work both inside and outside multi-program environments. Within *Windows/DESQview*, however, the **PLAYD -R** is unnecessary; whether or not you unload the TSR when finished is irrelevant since **EXIT** closes the window. And outside of *Windows/DESQview*, the **EXIT** is irrelevant (since it will normally have no effect). You will thus want to customize the batch file for your normal operating environment. While you're at it, you may want to permanently set the batch file's replaceable parameters (%1 %2 %3 %4 %5).

Help files (*.HLP) load into virtual memory as the program starts up. This is extended or expanded memory on a 286 or better (if available); it is the hard disk if not available. If you have the XMS and/or EMS memory, both start-up and help will run faster.

Windows 3.1

General

The **POWERUSR.ZIP** archive file contains program information files (*.PIF), icons (*.ICO) and a virtual timer (**TIMER.386**) for operation under *Windows 3.1*. (I have not tested either program under *Windows 3.0*.)

You can tweak the PIFs to your liking (especially the memory settings which I have not yet optimized). I recommend leaving the "full-screen/windowed" option set to "windowed", however; this allows you to use the arrow (graphical) mouse cursor instead of the block (character). Early versions of the PIFs did not specify the **C**lose **W**indow **o**n **E**xit setting; you should ensure that this box is checked so the window closes automatically when finished. You can run multiple copies of both editors under *Windows 3.1*.

Generally, you should **not** unload *Play/D* (i.e., **PLAYD -R**) at the end of the batch file when you're running under *Windows 3.1*. For MPUs, the unload resets the interface into "smart" mode which will disable other *Play/D* sessions and MME programs. You can re-enable the *Play/D* sessions via the MIDI Menu's RESET option, however.

Note also that *Wincake* supplies a virtual device driver (**VMPUD.386**) that disables DOS MPU programs. *Play/D* won't work at all if you install this in your **SYSTEM.INI** file; this is probably the first thing to check if you're a *Wincake* user and having problems. The *SoundBlaster* has a similar driver that will thwart **its** DOS programs, but I don't have complete information on this.

MME

When running *Windows 3.1* in standard mode (or in enhanced mode without MME programs running), the *Play/D* MIDI engine will run fine with no special consideration above what you've already read in **WARRIOR.TXT** and/or **CANVAS.TXT**. In enhanced mode with MME programs also running, however (i.e., where multiple MIDI "drivers" run simultaneously), you will need to specify different command-line options for *Play/D*. Be sure to 'unSET' your OPTn environment variables in this case so *Play/D* doesn't pick up any unintended settings.

When running in enhanced mode at the same time as MME programs, use *Play/D*'s **-e** option (instead of **-d**) and specify an IRQ of 0 (e.g., **-empu:0:336**). Turn the editor's MIDI **Thru Off** via <Alt-T> before touching your MIDI controller (if you need **Thru On** (probably), have the MME sequencer provide it). Don't use *Play/D*'s **-s** option and don't unload it with **-r** when finished with the editor.

Note that you should temporarily stop playback on the sequencer when actually sending sysex messages so the sysex and channel data don't mix.

Windows 3.1 (continued)

System Timing

You might have to experiment with your system timing to get accurate *Play/D* song playback when running concurrently with an MME sequencer. **TIMER.386** is a virtual device driver that should help by improving timing stability under enhanced mode.

To set-up this virtual timer, add **-h** (and perhaps **-m2**) to the *Play/D* command-line, copy **TIMER.386** to your **\WINDOWS\SYSTEM** directory, and add the line **DEVICE=TIMER.386** to the **[386Enh]** section of **SYSTEM.INI**. Then, make sure the **Exclusive in Foreground** option is unchecked in *Control Panel's 386 Enhanced* set-up. Lastly, you may need to increase the background priority for the program.

DESQview 2.4

The **POWERUSR.ZIP** archive file contains program information files (*.DVP) for operation under *DESQview 2.4*.

DESQview/386 seems unstable when using MIDI **Thru**; the virtual machine locks up frequently. You may want to turn the editor's **Thru Off** under *DESQview*; if not, you should probably save your patch data frequently.

Play/D Information

Play/D Command-Line Options

- d Device**—specifies the type of MIDI interface in use. **-d** is also used to give additional information about the device if a non-standard setup is being used. For the majority of users, this option will not be needed. The supported MIDI interfaces are listed below. Use **-d**, followed by the device name as shown to select one.

<i>Roland MPU-401</i> and -compatibles	-dmpu
<i>IBM PC Music Feature</i>	-dmfc
<i>Sound Blaster</i> MIDI interface	-dsbmidi
<i>Key Electronics MIDIator</i>	-dmidiator

The first 3 letters of the device name are sufficient (e.g., **-dmid** for *MIDIator*).

Note: The *MIDIator* cannot be detected automatically; if you use one, you must always specify the **-dmid** option.

You can add non-standard IRQ and I/O port information following the device id if necessary: **-d<dev>:<irq>:<ioaddr>**

For example, to define an MPU-compatible interface using IRQ 5 and I/O address 336 (hex), use **-dmpu:5:336**. If only the I/O address was required but the default IRQ was alright, use **-dmpu::336**.

irq and **ioaddr** are not relevant to the *MIDIator*, though the first option number can be given to indicate use of serial port **COM2** instead of the default **COM1** (ex: **-dmid:2**).

- e Exclusive**—similar to **-d** (with same syntax) but safer under *Windows 3.1* in enhanced mode. Prevents any other drivers from even being looked at.
- s Size**—determines the size of the MIDI receive buffer. Follow **-s** with a number up to 63, indicating the size of the buffer (in K bytes; 1K=1024 bytes). By default, no buffer is used, and *Play/D* operates in polled (versus interrupt-driven) input mode.
- r Remove**—removes *Play/D* from memory.
- h High Resolution Timer**—tells *Play/D* to use the **TIMER.386** services as described on page 3. This option is irrelevant outside of enhanced mode *Windows*.
- a Allocate**—indicates the amount of memory (in K bytes; 1K = 1024 bytes) to allocate for MIDI file storage. The default is **-a24** (24K bytes).

Play/D Command-Line Options (continued)

-m Mode—selects the operating mode: 1, 2, or 3. The default is **-m1**.

Mode 1: Use background interrupt stepping (default mode).

Mode 2: Same as **Mode 1**, plus conditional timer stepping.

Mode 3: Same as **Mode 1**, plus unconditional timer stepping.

Mode 1 relies on the underlying TSR mechanism, *TesSeRact*, to determine when it is absolutely safe to interrupt DOS in order to play.

Mode 2 adds use of the system timer (about 18 ticks per second) to advance play. This permits play to continue while running programs which otherwise would not allow *Play/D* to operate at all. Any DOS activity will temporarily suspend play, however. See warnings for **Mode 3** next.

Mode 3 combines idle time processing with forced timer stepping to assure high accuracy and continuous play regardless of other system activity. While it is unlikely that timer interrupts will cause any problems, this is technically "unsafe".

Users of 2/3/486 processors may select an additional command line option, **-c**, to specify AT real-time clock usage, in which case the **Mode 2** and **3** interrupt interval changes to 5 milliseconds.

-l LIM—specifies that **LIM EMS** (expanded memory) is to be used for MIDI file storage if available. This reduces the resident size of *Play/D*.

-v Verify—shows the MIDI device, I/O address, and IRQ number selected prior to program startup. You may wish to include this option if you suspect that automatic detection is not choosing the desired options, or to verify the IRQ being used.

Using Environment Variables

Since it may be inconvenient to add these options every time *Play/D* is run, options may be set up ahead of time through the use of DOS environment variables. A pool of environment variables named "opt1" through "opt9" are available. Pick any of the 9 available variables for the options which you would like to preset. For example, if you would normally use **-dmpu:5** and **-m2**, you could use the DOS **Set** command as follows:

```
SET OPT1=-dmpu:5  
SET OPT2=-m2
```