MIDAS for Windows 1.01

DOS version and libraries by Petteri Kangaslampi & Jarno Pannanen
Windows MCI version by Benjamin Cooley of CINEMATIX

Introduction

Welcome to MIDAS for Windows, the first Module player driver for Microsoft Windows. MIDAS for Windows allows MOD and S3M files to be played from any application which uses the Windows Media Control Interface (MCI). Now MOD's can be played from multimedia authoring systems, Visual Basic (TM), used in games or C programs, or embedded as objects in documents, in exactly the same way MIDI and WAV files are.

Features of MIDAS for Windows:

o Plays the popular 4 channel MOD files or 16 channel Scream Tracker 3 files (S3M).
o Uses the windows WAV device for output to any Windows compatible sound card.
o Special support for the Advanced Gravis Ultrasound card for extra high quality mixing with almost no CPU usage.
o Supports 16 bit mixing and stereo.
o Supports mixing for 11025khz, 22050khz, and 44100khz.
o Supports Dolby (TM) surround sound panning for Scream Tracker 3 files.
o Allows multiple pass through effect channels for video game sound effect mixing.
o High performance mixing routines for the WAV device minimize CPU usage.
o Supports Microsoft Windows Media Control Interface.

Installation

MIDAS for Windows is an MCI device driver which must be installed in your Windows control panel. Once MIDAS is installed, all applications which use MCI will recognize MIDAS and list it as a media player device.

To install MIDAS, first unzip the midas zip file to an empty directory or disk, then double click on the Control Panel program in the Windows Program Manager to run Control Panel.

In Control Panel, double click on the Drivers icon to run the Drivers applet. The Drivers applet window will appear listing the drivers currently installed in the system.

Click on the Add button in the Drivers applet to add a new driver. A dialog listing the stock drivers which may be installed in Windows will appear.

Click on Updated or Unlisted Driver in the list of drivers, then click on Ok. A dialog requesting the path name for the unlisted driver will appear.

Enter the path that you unzipped MIDAS for Windows in, then press Ok. The drivers will be copied to the Windows system directory, then the MIDAS configuration dialog will appear.

Configuring MIDAS

When the MIDAS drivers have been successfully installed, the MIDAS configuration window will appear. The configuration window will also appear when you choose the 'Configure' option or menu item from an MCI aware application. The configuration window allows you to change the output device, module play parameters, and mixing parameters for MIDAS for Windows.

Device:

The device box allows you to choose between outputting via the Windows WAV device or the Gravis Ultrasound. When outputting to the Windows WAV device, you may select the mixing rate, mixing resolution, and mixing quality (Low or Normal only). When outputting to the Gravis Ultrasound the mixing rate is always 44100khz, the mixing resolution is always 16bit, and the mixing quality is always High. See 'Advanced Gravis Ultrasound' below for a description of the Gravis Ultrasound output device.

Surround Sound:

Checking this enables Dolby (TM) surround sound panning support for Scream Tracker 3 files (S3M)'s. Checking this box increases memory usage because of the need to store both surround sound and normal sample in the intrument table. Gravis Ultrasound owners should turn this option off if they have less than 512k on their Ultrasound card (the original GUS only comes with 256k).

Protracker Tempo:

Click this box if you wish to use protracker compatible tempo control for MOD files.

Protracker Panning:

Click this box if you wish to use protracker compatible instrument panning for MOD files.

Mode:

Click Mono to play in mono, or stereo to play modules in stereo. On soundcards which do not support stereo, modules will always play in mono mode. Using stereo mode doubles CPU usage when playing using the WAV device since MIDAS must mix each channel independently. Using mono mixing is a good way to reduce CPU usage while not noticably degrading sound quality for CPU hungry applications (i.e. games).

Mixing Rate:

Select either 11025hz, 22050hz, or 44100hz. The lower the mixing rate, the lower the quality. Using 22050hz has the effect of filtering some of the high end when playing a song, while 11025hz filters out even more (it's yucky). The mixing rate is also intimately tied to CPU usage. Since hz means 'times per second', 44100hz means that the mixer must mix 44100 samples per second to supply the WAV device with an uninterrupted stream of audio data to play. Playing at 22050hz reduces this by half, while playing at 11025hz halves this amount again. For games, 22050hz is recommended unless running on a Pentium system. For soundcards which do not support 22050hz (??) or 44100hz, choosing these mixing rates will have no effect.

Resolution:

Select either 8 or 16 bit mixing. Though all modules supported by MIDAS (MOD's and S3M's) use 8 bit instruments, 16 bit mixing can slightly improve the quality of the mixed music. Enabling 16 bit mixing has little effect on the amount of CPU usage. If a soundcard does not support 16 bit output, choosing 16 bit will have no effect.

Quality:

Choose from Low Quality or Normal Quality for the Window WAV device (the GUS device is always High Quality). In this version of MIDAS, Normal Quality and Low Quality are identical.

Effect Channels:

Choose the number of pass through effects channels. Effect channels allow games and other applications to play sounds while a music file is playing. Note: enabling effects channels will reduce the total number of channels available for playing MOD's and S3M's. Some S3M's will fail to play if too many effects channels are used.

Mix Buffer Size:

Set the size of the mixing buffer for the WINWAV device. The larger the buffer, the further 'ahead' the digital signal mixer can mix, reducing the possibility of breaks in the music when the cpu is used for other tasks. Note: in this version of MIDAS, it is recommended that you set the mix buffer to a number below 20000 when playing out an 8bit card or with 8bit resolution as MIDAS uses an internal buffer which will not be large enough to provide data for a mixing buffer of 20000 or more.

Mix Blocks:

The Windows WAV device uses wave blocks to feed waveform data to your sound card. MIDAS divides the memory set by the 'Mix Buffer Size' above into the number of blocks specified by 'Mix Blocks' and creates a Windows WAV block for each. The blocks are

played in a loop to the Winows wave device, meaning that when a block is done, it is immediately sent back to the device to play again. The MIDAS mixing routines are called 50 times per second to fill any blocks that have already been played. This means that changing the number of blocks has the effect of changing the granularity of the buffer, allowing MIDAS to fill the buffer closer to the current play position, and divide the mixing cpu usage more evenly among the 50 times persecond update calls.

If you experience breaks when playing back a MOD or S3M, changing the mix buffer size and mix blocks will usually eliminate them.

Advanced Gravis Ultrasound (known as the GUS to us propeller heads)

Special support was provided for the Gravis Ultrasound because of its somewhat unique ability to play MOD and S3M files. The Gravis Ultrasound is one of the few sound cards which support wavetable sythesis AND uses RAM instead of ROM to store the wavetable instruments. Thus, the intruments in the MOD and S3M files may be 'uploaded' to the Ultrasound card and played like ordinary MIDI instruments with little CPU usage.

The quality when playing via the Gravis Ultrasound is also much higher than the Windows WAV device even when playing with 44100khz and 16bit. This is due to the Ultrasounds hardware oversampling (combining and averaging a larger number of samples for each sample outputted to decrease fuzz when playing instruments at a higher pitch), and interpolation (interpolating samples when instruments are played at a lower pitch to decrease 'steps').

Support for oversampling and possibly interpolation will probably be added in the future to MIDAS, but emulation of these in software is CPU intensive, and will not be practical if the CPU is needed for other chores (i.e. video, games, etc.)

Other sound cards currently support digital instrument RAM. Currently these include the Turtle Beach MAUI card, and the Sound Blaster AWE32. Support for these cards will most likely be included in future versions of MIDAS for Windows.

About Sound Modules, MOD's, and S3M's

MOD sequencer files originated on the Commodore Amiga computer, one of the first computers to include four channel digital audio output. Now with the advent of the Gravis Ultrasound, the PC world has finally caught up to, and indeed surpassed the capabilities of the Amiga computers. With the GUS, pc owners can finally play back MOD files without the crackling and fuzz associated with the software 'mixing' needed by other sound cards, and hear their MOD's the way they were supposed to sound.

Their are several MOD and MOD derivative formats now floating around. There is the 669 format, MED, STM, S3M, 6 and 8 channel MOD, etc. These formats have their various advantages and disadvantages, but all share a common structure based on the original MOD format. MIDAS can play MOD files (files with extensions of .MOD), and

also S3M files. MIDAS will eventually (??) be able to play many of the more popular formats including MTMs, MEDs, 669's, and 6 and 8 channel MODs.

A MOD file consists of an instrument waveform table, and a pattern table. The instrument waveform table holds the digital wave forms of each of the instruments used in the song in a format similar to Windows WAV files or the Ultrasound's PAT files.

The pattern table stores the note on/off information for the song, or the music sequence. A song is divided up into blocks of 64 beats, called patterns. The pattern table indicates which of these patterns is played at each point in the song. Usually a MOD file will contain a small number of patterns, each of which are repeated several times in different sequences as the song is played. MIDAS stores the pattern table in the computer's memory, and uses the computer's timer to play the various notes in the table in the proper sequence.

Although all MOD and MOD derivative formats are basically the same, the MOD format differs quite a bit from the MIDI (files with .MID extension) format. The first difference is that a MOD file stores its instruments in the file itself, while the MIDI file contains only note information and the instrument number that the synthesizer must use to play those notes. This means that a MOD file is capable of playing any instruments which the composer can digitize (or get samples for), while MIDI files are forced to use whatever instruments are provided by the sound card or synthesizer they're played on (usually only the orchestral General MIDI instruments.. yuck). However, this also means that the sample resolution of the instruments for the MOD will always be 8 bit (since the Commodore Amiga could only play back at 8 bit), regardless of the sample rate of the sound card. The Ultrasound card will play sampled sound at 16 bits so MOD instruments tend to sound duller when compared with the Ultrsound's '.PAT' instruments.

MOD may also only play 4 instruments and notes at a time, while MIDI files may play up to the maximum number of instruments and notes supported by the sound card. This limitation comes from the Commodore Amiga's ability to play only 4 digital channels at a time. New module formats, including the 6 and 8 channel MODs, ultratracker's 32 channel mods, and of course Scream Tracker 3's 16 channel MODS allow many more channels than the original MOD format.

Another more esoteric difference is in how note information is stored in a MOD vs. a MIDI file. MOD notes are stored in repeatable patterns of 64 beats with one column for each channel available and one row for each beat, while MIDI notes are stored serially as time stamped note messages.

Note: The upcoming MIDAS embedded patch MIDI synth will bridge the gap between MOD and MIDI. The MIDAS MIDI synth will include a patch librarian applet which will add and delete digital MIDI patches to the MIDI 128 patch table for composition with standard MIDI sequencing programs, and will embed the patches used by a midi file as SysEx messages so the file can be distributed and played with the correct instruments on another system (assuming it has an embedded pach aware synth).

Using Midas in Applications

The MIDAS driver is a Media Control Interface driver. For details on using MCI, see the documentation included with your compiler or authoring system.

To open the MIDAS module player, use the device name "modplayer" for the MCI_OPEN message. Elements must have an .S3M or MOD extension.

MIDAS responds to all basic MCI messages.

Extended MIDAS MCI commands:

Not Completed Yet.. Should be complete in a week or so...

Commands will include the ability to set MIDAS play parameters from software that are now only changable from the configuration dialog. The ability to add sound effects to the instrument list by passing raw sound data, waves, or wave file names, to remove sound effects, and to play sound effects. The ability to control sound effect panning (including surround panning). The ability to get a module information structure which describes the playing module in detail. The ability to have MIDAS send a play info message back to the calling applications window for every pattern beat which describes the current state of the module being played (for play status updating).

IMPORTANT: MIDAS for Windows may be used in free or public domain applications only. For a comercial license, see the information below.

The Future of MIDAS!

There are two groups behind the creation of MIDAS for Windows: Petteri Kangaslampi and Jarno Pannanen which coded the original MIDAS DOS libraries, and Benjamin Cooley and CINEMATIX which coded the Windows MCI version of MIDAS and wrote this file.

Basic MIDAS system (Petteri and Jarno):

I'm not exactly sure what they plan, but I assume we can expect more module formats to be supported (i.e. MTM, 6 & 8 channel MOD, ULT's, MED's, etc.). They may also eventually support more sound cards (AWE32 or MAUI) depending on their schedules and their interest. I'm sure we'll eventually have oversampling, and possibly interpolation, and 16 bit samples (for ULT's and other 16bit sample formats).

Note: Last time I talked to Petteri, he said they had another version of the DOS stuff, so I guess I'll convert it and put it in the next version of this MCI thing.

Windows MIDAS system (Ben & the rest of CINEMATIX):

Well, we're pretty heavily into the production of MAYHEM (a shareware game), so whatever we need for that product will almost certainly appear before the game does. This most likely will include WAV device pass through, and the MIDAS embedded patch MIDI synth (most likely MAYHEM's sound files will be embedded patch MIDI). We will also probably code support for the AWE32 if Jarno or Petteri don't in the near future.

Oh, and we should have a new version of MODUS that will use MIDAS (don't confuse these two programs though). MODUS is a nice Windows module player that I coded one night which currently only works with MOD's, and currently only plays on the GUS. Should have this done reasonably soon.

License

Ben's Note: Since MIDAS for Windows is a derivitive of the orginal MIDAS libraries, this license still applies.

HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR INABILITY TO USE, OF MIDAS SOUND SYSTEM.

As you have not signed this license, you are not required to accept it. However, as MIDAS Sound System is Copyright 1994 Petteri Kangaslampi and Jarno Paananen (the authors), only this license grants you the right to use, distribute and modify it. Therefore, by using, distributing or modifying MIDAS Sound System, you indicate that you understand and accept all the terms of this license.

1. MIDAS Sound System may freely be copied and distributed in its original, unmodified form as long as no fee is charged for the MIDAS Sound System itself. A fee, not exceeding 10USD, may be charged for the physical act of copying, as long as it is clearly stated, that the MIDAS Sound System itself is free and may always be distributed free of charge. When ever the MIDAS Sound System or portions of it are distributed, this license must be included in unmodified form, and all files must contain their original copyright notices. If only a portion of a file is being distributed, the appropriate copyright notice must be copied into it from the beginning of the file, and this license must still be included in unmodified form.

2. The MIDAS Sound System files may freely be modified, and these modified files distributed under the terms of Section 1 above, provided that the following conditions are met:

2.1. All the modified files must carry their original copyright notice, in addition to a list of modifications made and their respective authors.

2.2. Modified versions files belonging under this license still do so, no matter what kind of modifications have been made.

3. MIDAS Sound System may not be used in commercial programs, including shareware. No program using MIDAS Sound System may be sold for any price, and neither may any fee be charged for the right to use the program. Programs using MIDAS Sound System may always be freely distributed if no fee is charged for the distribution and may freely be used by anyone. The Program and its documentation must clearly state that the program uses MIDAS Sound System and that it may freely be used and distributed under the terms of this Section 3.

4. MIDAS Sound System, or portions of it, may not be included as parts of a commercial software package, in source code, object module or any other form. Neither may MIDAS Sound System, or portions of it, be printed in a commercial publication, including, but not limited to, magazines and books. Any software package or publication including portions of MIDAS Sound System may freely be used and distributed, as stated in Section 3.

MIDAS Sound System may, however, be included in free software packages as long as the whole software package can freely be distributed under the terms of section 3 and all conditions of this license are met. MIDAS Sound System, or portions of it, may also be printed for personal or educational usage, provided that no attempt is made to restrict the copying and usage of any parts of the printout or the publication in which portions of MIDAS are included.

Comercial licensing

Contact Petteri Kangaslampi (address below) for comercial licensing information. Remember, you MUST have a comercial license if you wish to include MIDAS in any shareware or comercial application. If you want to include MIDAS in a free or a public domain application however, MIDAS does NOT require a comercial license.

Jarno and Petteri's DOS MIDAS README.1ST File

MIDAS Sound System v0.31 revision b

This is the second public release of MIDAS Sound System, and, as the version number suggests, this is not a finished version. The most important feature lacking from this version is full documentation. Some documentation is included in the DOCS\ directory, which you should read through before proceeding, but mostly you are currently on your own. This release is, therefore, intended for experienced programmers only. The best source for learning to use MIDAS Sound System is the source code, so take a good look at the example programs, experiment with them and modify them. After all, that is what having the source code is all about.

This revision of MIDAS fixes all known bugs and compilation problems.

Sound Blaster series Sound Device and Gravis Ultra Sound Sound Device had both a single serious bug which caused them not to work with some card versions. Also all known compilation problems with TASM versions prior to 4.0 have been fixed, although the fixes have not been tested with TASM versions other than 4.0. In addition to this some minor changes were made and an example on how to use sound effects was added. See REVISION.031 for a complete list of changes made.

Now also included are precompiled object files of assembler modules for the Pascal version, as Turbo Pascal does not come with an assembler. To use these files, copy them from the PAS_OBJS\ directory to the main MIDAS directory. Note that you can not use the MAKETPU.BAT to compile MIDAS.TPU, as it would attempt to compile the object files, but use MAKE -fMIDASTPU.MAK instead. Object files for the C version are not included, as Borland C comes with TASM.

If you have any questions, comments or bugs reports, or just want to share some thoughts about programming in general, please contact us. We hope that you understand that supporting a free product is not always very motivating, especially under heavy pressure from outside world like school, and so without any feedback from you the MIDAS project will probably not have a long life. So, whether you like MIDAS or not, please let us know. And if you find MIDAS useful and use it in your programs, a souvenir postcard from your home city would be nice...

We will be extremely grateful for any kind of support you can offer to help us in continuing to make these free public versions available. We would need support BBSes, a HQ BBS from the H{me are in Finland and having a "official" support FTP-site would be nice too. And of course we can use practically anything computer related: programming information, example code, software, hardware etc.

Finally, a note to all sound card manufacturers and marketers: MIDAS does not support very many sound cards yet, but we will add support to practically any sound card, provided that we can get the programming information and a card for testing. So if you have a sound card that you would like us to support in its native mode, contact us.

Windows Information (Ben's Stuff)

Windows Version (V 1.00)

This beta Windows port by Benjamin Cooley. All original license and Copyrights still apply.

Additions and changes:

Version 1.01 (fixed!):

* Fixed GLARING bug with Sound Blaster and other cards (hey... it worked with my GUS, so what's the problem). Would not work unless mixrate, resolution, and channels matched exactly what the card was capable of outputting! Sorry.

* Added the ability to set the mix buffer size and number of blocks to the configuration dialog.

Version 1.0 (the buggy version):

* New WINWAV.C Windows wave audio device which replaces all other dsm devices (no specific SB, PAS, WSS, etc.). GUS is still specifically supported via the GUS.ASM file however.

  Sound devices are now: 1 - Gravis Ultrasound, 2 - Windows WAV.

* Changed ASM file routines to use Windows _hread and _hwrite because of segment/selector incompatibilities.

* Converted to MIDAS.DLL with all ordinary midas functions exported. .LIB file created with implib.exe used to link midas into a Windows C program. For other languages, see the 'using external DLL's' for information on how to use the MIDAS.DLL. Since DLL's support all languages, there is no specific Turbo Pascal Windows MIDAS.

* All fprintf's and error outputs changed to call MessageBox().

* ASM modules for SB, PAS, WSS, etc. are removed from link. Also the DMA.ASM file is also removed as its functions are replaced by Windows 'FAKE' dma functions in WINWAV.C.

* TIMER.ASM is replaced by TIMER.C which is reprogrammed to use the Windows Multimedia timer device (which actually uses the AT timer in a way which is very similar to how MIDAS was using it). NOTE: the vertical retrace functions are not supported in this version of Windows MIDAS.DLL (sorry).

* DSM.ASM changed to request a 16k buffer ALWAYS from the dmaAlloc function. This allows smooth mixing in Windows.

* Changed from using waveOutWrite from within the wave callback to calling waveOutWrite() from the multimedia timer to implement a ring buffer.

* Made GUS device detect via the information in the SYSTEM.INI file.

* Made GUS device save and restore Window GUS state so WAV and MIDI devices worked after runing MIDAS. Removed accesses to the mixing control register which seems to have been a problem in Windows

* Changed PlayModule() function for module players to use start and stop positions instead of loop start and stop positions. Removed looping, and added a void (*stopfunc)() callback function to report back to the main program when a module is complete.

* Added MCI.C for the mci interface to MIDAS.

* Added WINMIDAS.C to replace MIDAS.C which contains the MCI compatible midasXXX() functions (i.e. midasLoadModule(), midasStopModule(), midasPause(), etc.).

* Used .C files when I could. Why go to all the trouble to code everything in .ASM when .C is just as groovy?

Contacting the Guilty Parties

How to contact Ben and the rest of CINEMATIX...

* Send email to enigma@indirect.com

* Send snail mail to:

CINEMATIX STUDIOS
63 E. Main #713
Mesa, AZ 85201
U.S.A.

* Fly to Arizona, U.S.A., take a cab to 950 E. Southern, walk to my apartment, and knock on the door. (KIDDING.. please don't try this!)

* Any questions on LICENSING FOR COMERCIAL PURPOSES (including this Windows version), or the basic MIDAS system send to Petteri or Jarno. Please do not bother them with bugs and problems specific to this Windows version however (unless they say it's ok).

Contacting information for Petteri and Jarno
---

e-mail: (preferred)

pekanga@freeport.uwasa.fi (Petteri Kangaslampi)
Jarno_Paananen@sonata.fipnet.fi (Jarno Paananen)
gurus2@freeport.uwasa.fi (Jarno Paananen, possibly slower but more likely to get through)

FidoNet: (SLOW!)

Petteri Kangaslampi, 2:221/319 (Maximus Filecenter)

voice:

+358-31-646764 (Petteri Kangaslampi)
+358-31-3422147 (Jarno Paananen)
Please restrict your calls to 10.00 - 21.30, Finnish time.

normal mail:

Petteri Kangaslampi
Simeoninkuja 4
FIN-36240 Kangasala 4
Finland

Jarno Paananen
Puskalantie 6
FIN-37120 Nokia
Finland

BBS: No "official" support BBS yet, want to be one?

Whom to contact?

If you have a question about a particular subject, here is a list of which of us did what. Naturally you can contact either one of us - we will forward the messages as necessary.

Petteri Kangaslampi:

* General questions about MIDAS
* MIDAS architecture, including error handling and memory allocation
* Pascal version
* Pro Audio Spectrum, Windows Sound System and Sound Blaster Sound Devices
* Mixing routines (DSM).

Jarno Paananen:

* Gravis Ultra Sound Sound Device
* Protracker and Scream Tracker 3 module players