

Contents

Welcome to WAVmaker for Windows Help!

If you are new to WAVmaker, check out the [Tutorial](#) section. Users of WAVmaker 2.1 may want to skip directly to the [What's new](#) summary.

Basics

[Terms & Conventions](#) Things you never thought you'd want to know.

[Directory structure](#) A guide for the bewildered.

Operations

[Playing files](#) How to play MID and WAV files.

[Scanning files](#) How to find out more about MID and WAV files.

[Editing files](#) How to modify MID, WAV, PRG and LST files.

[Converting files](#) [How to render MIDs to WAVs](#), and much more!

Miscellanea

[Contacting the author](#) For questions, problems, suggestions...

[MIDI archives on the Net](#)

[Acknowledgments](#)

Biz

[How to REGISTER - and WHY!](#)

[Registration form](#)

[License information](#)

Welcome to the WAVmaker Tutorial!

[Index](#)

[Close](#)

WAVmaker's purpose is to aid you in the creation of high quality sound effects and music.

It offers a wide variety of sound editing functions and a feature unparalleled by any other sound editor on the market: [given a Standard MIDI file, it can create a CD quality song of virtually unlimited complexity and polyphony, on standard WAV format](#), ready to be included in your next program or multimedia production, or simply to be played back and enjoyed on its own - no matter if you own a wavetable card or not!

You can of course control all patch parameters and instrument assignments, add your own samples or create a wholly new instrument library. You can take a sound - any sound; your voice, a window breaking, a jet plane roaring by - modify it beyond recognition, add it to your sound library, and use it in your song. You can take a MIDI file, render tracks and channels separately, apply different effects to them, and mix them back down together. You can take a recorded vocal part, edit it, and add it to your song, resampling it if needed. WAVmaker turns your computer into a sophisticated, digital music studio where only your imagination sets the limits.

Does this all sound too good to be true? Well then - check it out! [Come along on a quick tutorial and demo tour of WAVmaker's capabilities!](#)



Tutorial index

The steps described below should be executed in the suggested sequence!

WAVmaker is assumed to be up and running; if it isn't, start it up now.

To find out more about the commands being demonstrated, click the **Help** button in the active WAVmaker window. For help with new terms or concepts, use the **Search** button and consult the **Terms & Conventions** section in WAVmaker Help.

In order to leave this tutorial, click the **Close** button in the Tutorial window (the one you're looking at now).

Step 1: Make WAVs

First of all, let's see to it that we have some material to play with!

Step 2: Check the WAV directory

Did it really work?

Step 3: Copy WAVs to Temp directory

Better use copies, so we won't make a mess of the instrument sound library.

Step 4: Play WAVs

Let's hear what they sound like...

Step 5: Scan WAVs

...for the real geeks, here are some statistics, too...

Step 6: View WAVs

...and here's what they look like!

Step 7: Mix WAVs

Learn about the most important of all WAV editing commands.

Step 8: Flange WAV

A fun DSP command.

Step 9: Extend WAV

Not all that fun, but needed for the next step.

Step 10: Echo WAV

More fun!

Step 11: Play MIDs

Let's hear what your sound card's built-in MIDI synth can do.

Step 12: Scan MIDs

Generate a report about the events in your MIDs.

Step 13: PRGed

A look at WAVmaker's patch editor.

Step 14: Render MID to WAV

The big one!

Step 15: Play song

I wonder what it will sound like!?!

Step 1: Make WAVs

Index

Next

Close

In order to run the examples, you need a few WAV sample files.

The sample files making up WAVmaker's sound library are shipped in a proprietary, strongly compressed format, and must be converted back to WAVs before they can be used (in general, this is only done once, when you use them for the first time). You can recognize such compressed sample files by their filename extension (DFT). While it is possible to expand DFTs one by one, using the **DFT to WAV** command (from the **Convert** panel), here we will take a quicker route.

In the **Selector** box, on WAVmaker's main panel, click the radio button marked **PRG**. This will cause WAVmaker to change working directory, to the path specified by the corresponding entry in the **Directories** box. Since the paths can be changed by the user at any time, we will always refer to all directories by their LOGICAL name, as shown in the **Selector** box. In this case, we are selecting the PRG directory.

The PRG directory should contain the files defining the instruments "played" by WAVmaker's MID to WAV converter. You can recognize such files by their filename extension (PRG). Right now, all you need to know about them is that, apart from various performance parameters, they contain references to the sample files containing the actual sound data. By reading these references, WAVmaker can determine which DFT files it needs to expand to WAVs before a given PRG, and hence a given instrument, can be used. It should not come as a great surprise that WAVmaker expects to find the necessary DFT files in the logical DFT directory, and that it expands them into the logical WAV directory.

For the time being, we will only be interested in the PRG file called **6.PRG**. This file describes General MIDI instrument (or "program", hence the choice of filename extension) number 6, which happens to be a harpsichord. On WAVmaker's main panel, scroll the filebox until you can see the file **6.PRG**, then select it (click on it).

In the **Operations** box, you should now see an active button labeled **Make WAVs**. This button is usually labeled **Convert** (this transformation act occurs only when you select a PRG file). Click the **Make WAVs** button!

WAVmaker will open a new panel containing a list of sample file names. If you wish to find out more about the **Make WAVs** command, you can click the **Help** button. In order to expand the sample files, click the **OK** button, and watch WAVmaker scroll down the list. When it's done, you will be returned to the main panel. This shouldn't take more than a couple of minutes (assuming that your computer is equipped with a Floating Point Unit).

Click the **Next** button in this window for tutorial Step 2.

Step 2: Check WAV directory



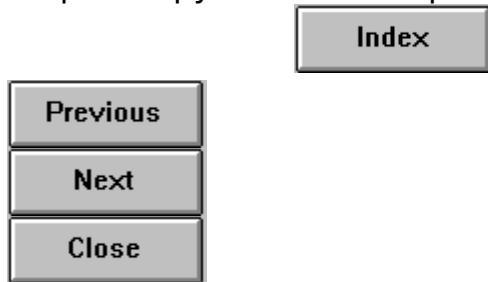
This is a very short step: we need to check that the sample files were indeed expanded into the WAV directory. In order to do so, [select the WAV directory](#) (click the **WAV** radio button in the **Selector** box on WAVmaker's main panel) and [verify that the following files \(with filename extension WAV\) are present:](#)

[gm6f#3, gm6f#4, gm6f#5, gm6f#6, gm6f3, gm6f4, gm6f5, gm6f6](#)

The naming convention should be fairly obvious: "gm" stands for General MIDI, then comes the MIDI program number (6), and finally the note at which the sound was sampled (the base note) in standard musical notation. All sample files in WAVmaker's General MIDI library follow this convention, except for the percussion sounds used in the drum kits; those are filed under descriptive names such as "KICK1" or "HIBONGO".

[Click the **Next** button in this window for tutorial Step 3.](#)

Step 3: Copy WAVs to Temp directory



As a safety measure, WAVmaker doesn't allow you to edit sample files in the WAV directory, since this could easily result in the loss or corruption of important instrument data. The recommended procedure is instead to copy the sample files which you wish to edit into the Temp directory (which you may think of as WAVmaker's scratchpad) and to work on them there. Once you're done, you can copy them back into the WAV directory (if they are to be used as MIDI instruments).

You are now going to copy the files gm6f4.WAV and gm6f5.WAV from the WAV directory to the Temp directory. There are two ways to do this; I suggest that you try both.

One file at a time (this allows you to change the destination file names):

- [Select the file gm6f4.WAV in the WAV directory.](#)
- [Click **Copy** in the **Operations** box.](#) A window entitled **Copy** is opened by WAVmaker.
- [In the **Copy** window, click the **Temp** radio button.](#) WAVmaker will update the destination file (**To**) edit box with the correct path, and enable the **OK** button.
- [Click the **OK** button.](#) The file is now copied to the Temp directory.
- [Repeat the above steps for the file gm6f5.WAV.](#)

Group copy (fastest route):

- [Select both gm6f4.WAV and gm6f5.WAV in the WAV directory.](#) Group selections can be made either by dragging the mouse or by pressing the **Ctrl** key (on the keyboard) while clicking the files to be selected.
- [Click **Copy** in the **Operations** box.](#) A window entitled **Group Copy** is opened by WAVmaker.
- [In the **Group Copy** window, click the **Temp** radio button.](#) WAVmaker will update the destination directory box and enable the **OK** button.
- [Click the **OK** button.](#) The files are now copied to the Temp directory.

Group operations are also supported by the **Move**, **Delete** and **Make WAVs** commands. All other commands act on individual files; when fed multiple file selections, they will use the last selected file.

[Once you are done copying, select the **Temp** directory from WAVmaker's main menu.](#)

[Click the **Next** button in this window for tutorial Step 4.](#)

Step 4: Play WAVs

[Index](#)

[Previous](#)

[Next](#)

[Close](#)

What can you do with your WAVs? Well, to start with, you can play them! [Select one in the Temp directory and click **Play** in the **Operations** box to bring up and start WAVmaker's WAV player.](#) The controls work just as on a tape recorder. Fast forward/backward move the current play position by 10% of the file size.

[Click the **Stop & Close** button in order to shut down WAV Play.](#)

This is a good time to check that your sound card is set up properly to work with the Windows MCI (Media Control Interface). WAVmaker doesn't "talk" directly to the hardware; it sends the data to Windows, which sends it on to your sound card, using a Windows driver supplied by the manufacturer.

If you can't get any sound out of your card, there are only three real possibilities:

- A really silly mistake, like leaving the sound card's digital audio volume set to zero. Use the control software which should have come with your card to check (and possibly correct) this.
- Your sound card is not properly installed to work under Windows. Refer to your Windows and sound card documentation.
- Your sound card can't handle WAV files of the kind you're trying to make it play. The SB Pro, for instance, can only play 8 bit files; WAVmaker by default uses (and produces) 16 bit files. The sampling rate may also be too high (WAVmaker defaults to 44100 kHz). In particular, all sample files in WAVmaker's General MIDI library are 16 bit, mono, 44100 kHz.

[Click the **Next** button in this window for tutorial Step 5.](#)

Step 5: Scan WAVs

[Index](#)

[Previous](#)

[Next](#)

[Close](#)

Scanning WAVs can be interesting, too. This gives you a quick summary of the sample file's characteristics, and can be quite useful (e.g. for checking compatibility with WAVmaker - not all WAV files are kosher - recording levels, saturation, or format information).

In order to scan a WAV file, select it from WAVmaker's main panel and click **Scan**. The data will be reported as it is being gathered, cumulatively. Hitting the **Cancel** button will abort the command and leave the partial results on screen. [Click the **OK** button to close the WAV Scan window.](#)

If WAV Scan can read a sound file without error, chances are that the file will work fine with all WAVmaker commands.

The RMS (Root Mean Square) and the number of saturated samples are especially interesting if you are checking a new sample file or the results of an editing operation:

- Many saturated samples (i.e. samples taking on the min/max values allowed by the word size employed) indicate too high a recording level, causing distortion.
- Too narrow a min/max range on the other hand gives you bad dynamic response and a low signal/noise ratio.

Ideally, you should always try to use the full word size, just barely avoiding saturation.

[Click the **Next** button in this window for tutorial Step 6.](#)

Step 6: View WAVs

[Index](#)

[Previous](#)

[Next](#)

[Close](#)

Very often, you'll want to look at a graphical representation of a sample file (this is particularly true when setting loop points for sounds to be used as MIDI instruments). You can do this from a couple of places in WAVmaker. For now, we'll use the WAV Edit window.

In order to open the WAV Edit window, select the file [gm6f4.WAV](#) from the Temp directory and click **Edit** in the **Operations** box. (Of course, any WAV file will do, but in the next step we'll be using the WAV editing functions on gm6f4.WAV, so we might as well open the WAV Edit window with it). WAVmaker will open a dialog (entitled Edit) asking you to confirm that you wish to edit the file as a WAV. [Confirm this by clicking OK.](#)

You should now be looking at the WAV Edit window. Don't be overwhelmed by its apparent complexity! Once you get the hang of it, it's really quite easy to use, while giving you a lot of control. [Please refrain from trying out the WAV editing commands at this time.](#) Instead, if you really can't resist, [click the Help button](#) and read about the various controls available to you. For now, we're only interested in one of them: the **View** button, sitting next to the source file (**From**) edit box, near the top of the window.

[In order to see a graphical representation of the selected file, click View.](#) This will cause WAVmaker to open a WAVview window. [In order to find out all you need to know about WAVview, click its Help button.](#)

[In order to close WAVview, click its OK button.](#) You will be returned to the WAV Edit window. Don't close it yet, as we'll need it in the next step!

[Click the Next button in this window for tutorial Step 7.](#)

Step 7: Mix WAVs

[Index](#)

[Previous](#)

[Next](#)

[Close](#)

You will now mix down the files gm6f4.WAV and gm6f5.WAV to the single file temp.WAV, all in the Temp directory. [Assuming that the WAV Edit window is still open, click the **Mix** button.](#)

You should now be looking at the WAV Mix window.

[The source file list should contain gm6f4.WAV, the destination file \(**To**\) should be temp.WAV. If not, edit these entries.](#)

Click the **Help** button to learn more about WAVmaker's mixer.

[In order to add gm6f5.WAV to the source file list, select it in the file box and click the **Add** button.](#) Leave the amplitude factors at their default values (1.0 - no amplification).

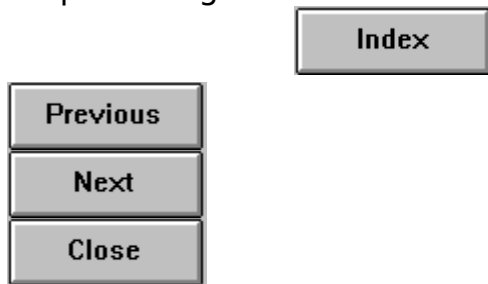
[In order to start the mixer, click **OK**.](#) WAVmaker will scroll down the list of source files, showing its progress for each one of them using the gauge in the panel's lower left hand corner. Upon completion, WAV Mix closes automatically, returning you to the WAV Edit window.

In the WAV Edit window, the **Swap** button (next to the destination - **To** - file edit box) should now have been activated. This button is available only when the destination file exists. Its purpose is simply to swap the source (**From**) and destination (**To**) filenames. This can be very convenient when performing a sequence of WAV editing commands. It's also the quickest way to check the result of a WAV editing operation, as we'll do next.

[Click the **Swap** button!](#) The file temp.WAV should now have become the new source file. [Next, click the **View** button.](#) WAVview should start as usual, allowing you to see and play temp.WAV. When you're done checking the result of the mixing operation, close WAVview by clicking **OK**.

[Click the **Next** button in this window for tutorial Step 8.](#)

Step 8: Flange WAV



You should now be looking at the WAV Edit window, and the source (**From**) file should be temp.WAV. We are about to apply the **Flange** command to this file; first, however, [change the destination \(To\) filename to temp2.WAV \(still in the Temp directory\)](#). Also [check that the Time unit is ms \(milliseconds\)](#). If it isn't, click the **ms** radio button.

[Next, click the Flange button](#). The Flange panel will open, showing the default parameter settings. If you do not already know about flanging, click the **Help** button. The Help entry will also teach you how to use the patch manager on the right side of the panel.

When you are done reading the Help entry,

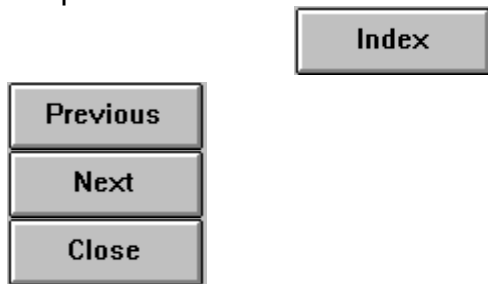
- [change the Original level value to 0.5;](#)
- [change the Effect level value to 0.5;](#)
- [change the Feedback value to 95;](#)
- [change the Variation value to 10;](#)
- [click OK.](#)

WAVmaker will execute the command, showing its progress as it goes along. The Flange panel is closed automatically upon termination, returning you to the WAV Edit window.

[In the WAV Edit window, click Swap again](#). This makes temp2.WAV the new source file. [Next, click the View button](#). WAVview should start as usual, allowing you to see and play temp2.WAV. When you're done checking the result of the flanging operation, [close WAVview by clicking OK](#).

[Click the Next button in this window for tutorial Step 9.](#)

Step 9: Extend WAV



This step is not particularly exciting, but it improves the result of the next one, when we are going to apply some echo to our sample file. In order to make room for a natural echo decay, we must first append some silent space at the end of the current source file, temp2.WAV.

Do the following:

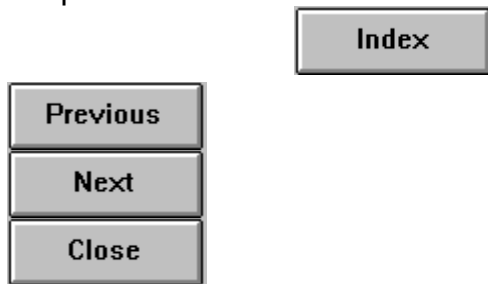
- Check that the **Time unit** is **ms** (milliseconds). If not, click the **ms** radio button.
- Change the name of the destination (**To**) file to temp3.WAV.
- Click the **Extend** button. WAVmaker will open the Extend panel.
- In the **Duration** field of the Extend panel, enter the value 3000 (for 3000 milliseconds).
- Check the **Append** box. This causes a silent section of the specified length to be added at the exact end of the source file. Otherwise, the silent section would be inserted at the location specified by the **Start** field in the WAV Edit window.
- In order to launch the **Extend** command, click **OK**.

WAVmaker will execute the command, showing its progress as it goes along. The Extend panel is closed automatically upon termination, returning you to the WAV Edit window.

In the WAV Edit window, click **Swap** again. This makes temp3.WAV the new source file. As it's just like temp2.WAV, plus three seconds of trailing silence, we won't bother looking at it.

Click the **Next** button in this window for tutorial Step 10.

Step 10: Echo WAV



Now let's add some echo! The source (**From**) file in the WAV Edit window should be temp3.WAV.

Do the following:

- Check that the **Time unit** is **ms** (milliseconds). If not, click the **ms** radio button.
- Change the name of the destination (**To**) file to temp4.WAV.
- Click the **Echo** button. WAVmaker will open the Echo panel.
- Change the **Feedback** parameter to 30.
- Change the **Delay** parameter to 800.
- In order to launch the **Echo** command, click **OK**.

WAVmaker will execute the command, showing its progress as it goes along. The Echo panel is closed automatically upon termination, returning you to the WAV Edit window.

In the WAV Edit window, click **Swap** again. This makes temp4.WAV the new source file. Next, click the **View** button. WAVview should start as usual, allowing you to see and play temp4.WAV. When you're done checking the result of the Echo command, close WAVview by clicking **OK**.

This ends the WAV editing section of our quick tour. We have just barely touched the WAV editing commands, but hopefully you've learned enough to continue their exploration on your own. Click **OK** in the WAV Edit window to close it and return to WAVmaker's main panel.

Click the **Next** button in this window for tutorial Step 11.

Step 11: Play MIDs

[Index](#)

[Previous](#)

[Next](#)

[Close](#)

Playing a MIDI file using your sound card's built-in synth (rather than WAVmaker's on wavetable sound) is quite analogous to playing a WAV file (Step 4 in this Tutorial).

- [On WAVmaker's main panel, select the MID directory.](#)
- [In the file box, click the file Solfeggi.MID.](#)
- [In the **Operations** box, click **Play**.](#)

[Click the **Stop & Close** button in order to shut down MID Play.](#)

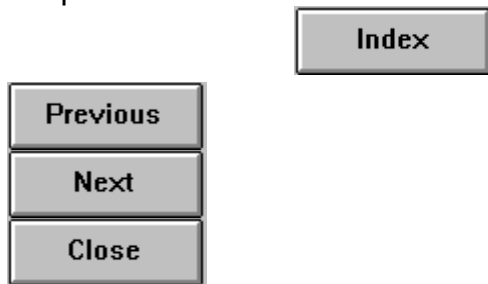
The MIDI player looks and works just like the WAV player - and just like the WAV player, it doesn't "talk" directly to the hardware; instead, it sends the MIDI data to Windows, which sends it on to your sound card, using a Windows driver supplied by the manufacturer.

Again, if you can't get any sound out of your card, there are only a few real possibilities:

- A really silly mistake, like leaving the sound card's MIDI volume set to zero. Use the control software which should have come with your card to check (and possibly correct) this.
- Your sound card is not properly installed to work under Windows. Refer to your Windows and sound card documentation. In particular, check that the MIDI Mapper has been configured correctly.

[Click the **Next** button in this window for tutorial Step 12.](#)

Step 12: Scan MIDs



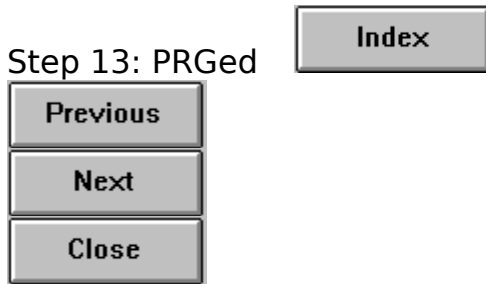
When applied to a MIDI file, the **Scan** command generates a text file describing (some of) the events in the MID, and lets you browse/print it using the default text editor. Unless you have specified a different text editor with the **Text editor** command in the **Edit** dialog, WAVmaker will invoke Notepad.EXE; see the [Editing text](#) entry in Help for details (and for tips on a great, freely available text editor!).

The report generated by a MIDI Scan can be quite useful for determining which patch (PRG) and instrument sample (WAV) files will be needed when rendering the MID to WAV. Also, if **Scan** can make sense of a MID, then chances are that all other commands will cope, too (there are many things that can go wrong with MIDI files, especially if you download them from an Internet site or from a BBS).

In order to scan e.g. Solfeggi.MID, select it (in the MID directory) and click **Scan** in the **Operations** box. Terminate the command by closing the text editor.

Click the **Next** button in this window for tutorial Step 13.

Step 13: PRGed



A full description of WAVmaker's patch editor, **PRGed**, is beyond the scope of this demo tour, but I want you to be aware of its existence. Using PRGed, you can modify all instruments "played" by WAVmaker - everything from sample files to envelope shapes and loop points to performance parameters like attack velocity sensitivity and bend range.

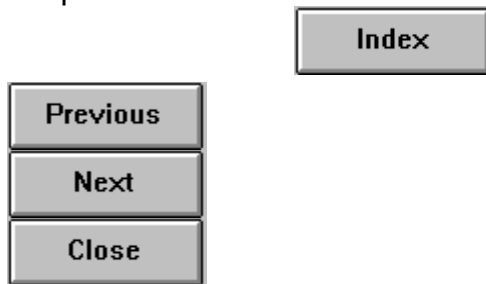
- [From WAVmaker's main panel, select the **PRG** directory.](#)
- [In the file box, select the file **6.PRG**.](#) This is the PRG file which WAVmaker will be using in the next Tutorial step.
- [In the **Operations** box, click **Edit**.](#) WAVmaker will now open the Edit dialog, and ask you to confirm that you want to edit the file as a PRG (it sometimes makes sense to edit PRGs as text files).
- [In the Edit dialog, click **OK**.](#)

You should now be looking at the patch editor window. [In order to learn more about it, click the **Help** button.](#)

[Close the PRGed window by clicking **Cancel**.](#) This ensures that any changes you may have made will not be saved in the PRG file.

[Click the **Next** button in this window for tutorial Step 14.](#)

Step 14: Render MID to WAV



This is really WAVmaker's business - and it's easier than most other operations you've been performing, too!

In order to render Solfeggi.MID, execute the following steps:

- On WAVmaker's main panel, select the **MID** directory.
- In the file box, select the file Solfeggi.MID.
- In the **Operations** box, click the **Convert** button.

You should now be looking at the Convert panel. You can find out more about it by clicking the **Help** button. When you are sufficiently confused...

- Click the **Mono** radio button. This instructs WAVmaker to produce a monophonic WAV (surprise!), which saves both disk space and time (since disk read/write operations are what really takes time when rendering MID to WAVs). There is no quality loss since Solfeggi.MID only uses a lone harpsichord, without any stereo panning at all. If you were to produce a stereo file, the two channels would be identical.

You should leave the other default selections as they are. In particular, forcing WAVmaker to produce 8 bit files from 16 bit sample files (all sample files in WAVmaker's GM library are 16 bit) is usually a bad idea, since this means that mixing is done in 8 bit precision, with samples obtained by discarding the low byte of each 16 bit word. The problem is that this procedure will often leave only 3-4 significant bits per sample, introducing LOTS of noise.

If you only have an 8 bit card, such as a SB Pro, let WAVmaker produce a 16 bit file, then convert it to 8 bits using the Mix command (described in [Step 7](#) of this Tutorial).

- Click the **OK** button. WAVmaker will now open the MID to WAV window. As usual, you can read all about it by clicking the **Help** button. For the time being, you don't really need to concern yourself with the controls in this window; the default selections will work just fine.
- In order to start MID to WAV rendering, click the **OK** button. This is not a fast command; the actual time needed depends on your CPU, available memory, and disk performance. My old 486DX66 needs 5 minutes to create solfeggi.WAV, running WAVmaker alone and using no disk cache.

If you haven't already executed Tutorial [Step 1](#), WAVmaker will present you with a **Make WAVs** dialog prior to actual rendering. Use the Help button (or see Tutorial Step 1) to find out more about it.

The MID to WAV (and LST to WAV) command creates its output in the Song directory.

If you decide that it's taking too long, you can abort the rendering process by clicking the **Cancel** button. Allow for some reaction time: WAVmaker will not check the button in the midst of heavy operations.

Assuming that you have successfully rendered Solfeggi.MID, you may also want to try the other demo MIDs. You can either let WAVmaker expand the necessary sample files automatically, at rendering time, or use the **Make WAVs** command, as in [Step 1](#) of this Tutorial: find out which MIDI programs (instruments) are required for each MID using MID Scan (as in [Step 12](#)) and then run **Make WAVs** on the corresponding PRGs. Keep in mind that drum kit PRGs have a leading "D" in their name (so the Standard GM drum set, program number 0, corresponds to the patch file D0.PRG). If you are in a hurry, you can use the following list:

- [Solfeggi.MID](#) requires [6.PRG](#) (Harpsichord).
- [Drums.MID](#) requires [D0.PRG](#) (Standard drum set).
- [SJam.MID](#) requires [D0.PRG](#) (Standard drum set), [3.PRG](#) (Honky-tonk piano) and [36.PRG](#) (Slap bass).
- [Air.MID](#) requires [6.PRG](#) (Harpsichord) and [48.PRG](#) (Strings).

I suggest that you try the MIDs in the order above. All files except Solfeggi.MID should be rendered in stereo, i.e. using the default selections in the MID to WAV window (16 bits, stereo, 44100 Hz).

Click the **Next** button in this window for tutorial [Step 15](#).

Step 15: Play song

[Index](#)

[Previous](#)

[Close](#)

By now, you've probably figured it out on your own, but I include this (last) step for completeness. In order to play the songs rendered from MID to WAV in Step 14,

- [select the **Song** directory](#) (in the **Selector** box on WAVmaker's main panel),
- [select the WAV file to be played in the file box,](#)
- [click **Play** in the **Operations** box,](#) and
- [ENJOY!](#)

This concludes our guided tour. We've only skimmed the surface, but I hope that you will be able to continue on your own from here. Remember that the Help file contains lots of useful and readily available information. I hope you'll have fun exploring WAVmaker's features!

[Click the **Close** button in this window in order to terminate the Tutorial.](#)

What's new

If you're using Windows 3.1, you won't notice much difference between WAVmaker 2.1 and 2.2. Those changes which have occurred are (apart from minor bug fixes) all intended to reflect modifications to the code motivated by the port to Windows 95/NT. The intention is that WAVmaker 2.2 should behave almost identically on all supported platforms.

If you're using Windows 95/NT, I hope you'll notice a speed improvement, apart from the benefits of improved multitasking. Make sure that you have the right package! The main window title for the 32 bit port is "WAVmaker for Windows 95/NT".

Well, that pretty much sums it up. [I hope you'll enjoy using WAVmaker 2.2!](#)

Terms & Conventions

This is a list of a few terms and conventions used in WAVmaker's Help system. If you know what you're looking for, use the **Search** button!

First of all, a few **conventions**:

- [Left and Right pointed brackets \(< and >\)](#)
- [Samples](#)
- [Filename extensions](#)
- [Directory references](#)

Now for some **terms**:

- [MIDI](#)
- [Standard MIDI files](#)
- [General MIDI \(GM\)](#)
- [GS](#)
- [PRG](#)
- [LST](#)
- [WAV Sample file format](#)
- [DSP \(Digital Signal Processing\)](#)
- [DFT](#)

Left and Right pointed brackets, < and >, are used to enclose symbols representing some piece of data. For instance, if you are told to enter a [<track>](#), you are not supposed to reproduce that line verbatim, but rather to replace [<track>](#) with an actual value, such as [10](#).

The word **sample** usually means "individual data word", as opposed to "sample file" (a file made up of samples). When counting samples, as in sampling rates, the count is [per sound channel](#). A stereo file sampled at 44100 Hz actually contains 88200 samples for each second's worth of data.

Filename extensions are always referred to **without** a leading period (i.e. "WAV", not ".WAV").

Unless otherwise stated, all directory references are to LOGICAL directories, i.e. to the symbolic directory names displayed in the main panel's **Selector** box (MID, LST, Song etc.). See [Directory structure](#) for details.

MIDI: Musical Instrument Digital Interface. This started out as a hardware oriented standard for connecting sequencers (back in the days when those were not programs running on PCs but actual, dedicated hardware boxes), synthesizers and sound modules made by different manufacturers, allowing them to exchange note on/off and controller information.

For the historically minded, the first public demonstration took place at the 1983 North American Music Manufacturer's convention in Los Angeles, where two synthesizers made by different companies were shown being played in unison from either keyboard.

The heart of the standard is a 31.25 kBaud serial interface, using common 5-pin DIN connectors and cables, and a data format for the description of note and controller events.

Various additions made after 1983 include the MIDI Sample Dump Standard (SDS), for the exchange of digital audio data, and Standard MIDI files, for the exchange of music sequences via disk files).

Standard MIDI files

WAVmaker recognizes **Standard MIDI files, Level 0 and Level 1**, as defined by the Standard MIDI file specification (version 1.0). These are the formats usually employed to exchange sequences in a manufacturer-independent way, and the ones which you are most likely to encounter on Internet sites and BBSs.

Standard MIDI Level 0 files consist of [one multi-channel track](#). With only one track, [the number of independent channels is 16](#). This is quite enough for most purposes, especially since all percussion sounds and special effects can be kept on one channel (each channel controls a virtual 128-note keyboard, so by assigning a different sample file to each key you can control as many as 128 different sounds with only one channel). [Channel 10 is usually reserved for percussion sounds](#). You can also change program (instrument) at any time, for any channel, so instruments played at different times can use the same channel.

Standard MIDI Level 1 files consist of [up to 65 535 simultaneous tracks](#) (a so called "vertically one-dimensional form"). [Each track is just like the single track in a Level 0 file](#). In principle, you could use this capacity to simultaneously control 65 535 tracks * 16 channels, for a total of 1 048 560 independent channels. Since each channel can control up to 128 different sounds, you could play up to 134 215 680 distinct sounds at the same time! In practice, you'll find few if any Level 1 files in public archives, partly because Level 0 is enough for most practical purposes, partly because most people can not be expected to have the hardware necessary to control more than 16 MIDI channels (the MIDI hardware interface with which the whole MIDI thing got started back in the early 80s - nowadays we sometimes tend to forget that MIDI really is short for Musical Instrument Digital Interface! - handles 16 channels; in order to have more, you need multiple MIDI ports and software to control them, e.g. a sequencer capable of sending different tracks to different ports). On the other hand, Level 1 files are THE format to use if you really want/need to take advantage of the polyphonic capabilities offered by WAVmaker (let me know if you ever come close to using it at FULL capacity!).

There is also a **Standard MIDI Level 2** format, used to store [multiple songs](#) (a "horizontally one-dimensional form"). Since there is little to be gained from using one Level 2 file instead of several Level 1/0 files, this format is almost never used or supported.

General MIDI (GM) is a set of specifications for sound generating devices covering such things as the minimum number of simultaneous voices, MIDI events which should be recognized and supported, and - most important! - [which sounds to assign to MIDI program numbers \(the GM Instrument Map\)](#). GM also includes a [Percussion Key Map](#), derived from the Roland/Sequential mapping used on early drum machines, which assigns percussion sounds to note numbers.

GS is a fairly compatible superset of General MIDI used by Roland equipment, including the popular Sound Canvas modules and cards. It adds some sound effect control (chorus, reverb), drum kit program number conventions (GM does not support multiple drum kits), and variations for each basic GM program (instrument). Not supported by WAVmaker, except for the adaptation of Sound Canvas drum kit program numbers.

While various interpretations of the letters GS have been proposed (e.g. General Synthesis), word from Roland is that GS is **not** an acronym; GS means GS, period.

PRG is the default filename extension for the [instrument files \(patches\)](#) used by WAVmaker.

Each PRG file corresponds to a MIDI program (instrument). The information needed to handle MIDI program 0 (Piano 1) is in 0.PRG; MIDI program 6 (Harpsichord) is described by 6.PRG; and so on. Drum sets have their program number preceded by a "D", as in D0.PRG (Standard Set).

PRGs do not contain any sound data; the actual samples are kept in WAV files. PRGs assign sample files to keys (notes) and impose performance parameters (base volume and pan position, controller sensitivities, envelope shape, loop points...). Each key on the virtual 128-note "keyboard" described by the PRG file can be assigned its own individual sample file and performance parameters.

In order to view and edit PRG files, see [Editing PRGs](#).

LST (WAVmaker playlist) file format

LST is the default filename extension for the [playlists](#) used by WAVmaker. Playlists contain musical event information (usually extracted by WAVmaker from a [Standard MIDI file](#); see [MID to LST](#)) and performance parameters (from one or more [PRG](#) files) in human readable form. They can be viewed and edited with any standard text editor before being [rendered to WAV](#).

The first line in a LST file is the max event time (usually, the last Note Off event) in milliseconds. Upon rendering, it is used by WAVmaker to preallocate an empty WAV file of the corresponding size. This way, (almost all) the necessary disk space is grabbed right away, reducing the risk of out-of-space errors being caused by somewhere in midrun by some other program also using the disk (although the WAV file will often need to be extended later on, to accomodate a release section following the last Note Off event; if you are out of luck, that release section will be the proverbial straw...). Also, fatal errors will still leave you with something to listen to (everything rendered before their occurrence) without having WAVmaker constantly updating the WAV header for the current file size (a slow affair, due to long disk seek times).

The rest of the LST file is made up of one or more Sample Blocks delimited by curly brackets (`{` and `}`).

Each Sample Block is headed by a line listing the sample file name, the base key number, the base volume, the base panpot, and the various controller and envelope parameters specified in the PRG file, in the same order as in the PRG file.

In the subsequent lines, there are one or more Note Blocks, one for each note produced using the current sample file (low notes are listed first). Note Blocks are delimited by square brackets (`[` and `]`).

The first line in each Note Block is the note number.

The subsequent lines are time-ordered events, on the general format `<event time in milliseconds> <event codes> <event parameter>`. The event code/parameter combination should be one of the following:

- `ON <velocity>` (Note On)
- `OFF <velocity>` (Note Off)
- `PRS <pressure>` (Channel or Polyphonic Pressure)
- `MOD <modulation>` (Modulation Wheel)
- `VOL <volume>` (Volume)
- `PAN <position>` (Stereo Panning)
- `EXP <expression>` (Expression)
- `BND <bend>` (Pitch Bend)

WAV Sample file format

WAV files are a special case of RIFF (Resource Interchange File Format) forms, a tagged file structure used for (MS Windows) multimedia resource files. They support many more features than those needed by WAVmaker, such as embedded foreign format files, comments, playlists, and more. Rather than to bloat WAVmaker's code with a full (=large and slow) WAV file parser, the decision has been made to restrict all WAV in/out to the bare minimum needed for digital audio, i.e. to files made up of only one WAVE form with the following properties:

- Data format category: [MS Pulse Code Modulation \(PCM\)](#).
- Bits per sample: [8 or 16](#).
- Number of sound channels: [1 or 2 \(mono or stereo\)](#).
- Sampling rate: [11025, 22050 or 44100 samples/channel/second](#).
- [No fact chunk](#).
- [No cue chunk](#).
- [No playlist](#).
- [No associated data list](#).

Essentially, this is just a raw sound data file + a header containing data format information. All WAV samples you encounter are likely to conform to these choices (apart from the sampling rate, which may vary, this is the format normally produced when a new WAV file is recorded).

WAVmaker 1.0 could only read WAV files strictly adhering to this specification. [As of version 2.0, it recognizes and skips any chunks preceding the first data chunk. Only the first data chunk is used.](#) Output is strictly as above.

The sampling rates are those considered "standard" for PCs. While supporting a continuous range would be easy for most applications, there are cases when the choice of sampling rate affects the operations to be performed in non-trivial ways (it may for instance be necessary to determine the prime number closest to the number of samples corresponding to a given time delay). Having to consider only a few different choices of sampling rates allows for the use of look-up tables, speeding up execution.

If you want to use a WAV file which doesn't conform to the subset recognized by WAVmaker, you can convert it with a sound editor supporting the full format definition.

DSP (Digital Signal Processing) is the art of manipulating digitally stored representations of real world signals. In WAVmaker, the signals are sounds and the representation is a WAV sample file. See Editing WAVs.

DFT is the default filename extension for compressed sound files created by the WAV to DFT converter. DFTs can be decompressed to playable WAV files using the inverse DFT to WAV converter.

Directory structure

All directories used by WAVmaker are shown on the main panel.

The quickest way to change working directory is by means of the **Selector** box. This causes WAVmaker to use the path specified by the corresponding entry in the **Directories** box.

You should understand the distinction between **LOGICAL** and **PHYSICAL** directories:

- **The Selector box displays the current logical directory.** Which logical directory you are in affects WAVmaker's operations. For instance, WAVmaker will not allow you to edit a WAV file directly in the logical WAV (sample file) directory (a safety measure intended to protect you from accidentally damaging your sample files); you are instead required to copy the file to be edited into the logical Temp ("scratchpad") directory.

You have the following logical directories at your disposal:

- MID
 - LST
 - Song
 - PRG
 - WAV
 - Temp
 - DFT
- **The Directories box displays the actual (physical) paths corresponding to the logical directories.** You can change these paths at any time. They are saved automatically when you **Exit** WAVmaker, and restored the next time you start up the program again.

The paths specified on the main panel are used by WAVmaker whenever it has to go looking for, or create, a file. MID to WAV rendering, for instance, uses the logical MID, PRG, WAV, LST and Song directories. By changing the path corresponding to the logical PRG directory, you can let WAVmaker use an alternate set of MIDI program patches. By changing the path corresponding to the logical WAV directory, you can let it use an alternate set of sampled instruments. This way, you can easily maintain any number of independent patch and sound banks.

You can also browse among physical directories in the **Files** box. This is particularly useful for quick escapades outside of WAVmaker's logical directory structure. Note that WAVmaker will keep behaving according to the current logical directory selection (in the **Selector** box). The most permissive logical directory (in terms of allowed operations) is Temp, so you may want to go there before you start browsing.

The logical directory **MID** is where I recommend that you keep all your MID files. If you like to divide your work into projects, consider creating a distinct physical MID directory for each project.

The logical directory **LST** is where I recommend that you keep all your LST files. The MID to LST and MID to WAV commands default to this directory when creating LSTs; unless you do a lot of manual editing, you will have little reason to bother with it (other than erasing old files).

Song is the logical directory where the MID to WAV and LST to WAV commands put their end product. If you like to divide your work into projects, consider creating a distinct physical Song directory for each project.

The logical directory **PRG** is where WAVmaker expects to find the PRG (patch) files needed by the MID to LST and MID to WAV commands. If you want to create an alternate patch bank, put it into a separate physical PRG directory .

The logical directory **WAV** is where WAVmaker expects to find the WAV files needed by the MID to WAV and LST to WAV commands. If you want to create an alternate sample file bank, consider putting it into a separate physical WAV directory.

The Make WAVs command uses the WAV directory as its destination path.

Note that WAVmaker does not allow WAV file editing in this directory. If you want to edit a WAV file, copy it to the Temp directory and work on it there, then move the finished result back into the WAV directory.

Temp is the logical directory functioning as WAVmaker's "scratchpad". Just about any operation is allowed there. Among other things, it's the preferred location for WAV editing.

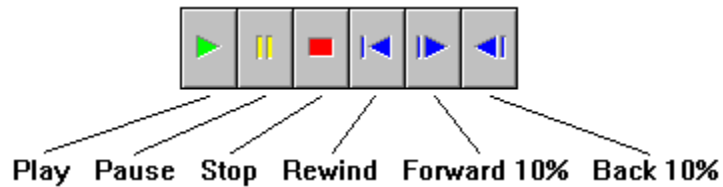
The logical directory **DFT** is where I recommend that you keep all your DFT files. It's also where the Make WAVs command expects to find the compressed files to expand back into the WAV directory.

In order to play a MID or WAV from the main panel, use the mouse to select the file. The **Play** button will be activated. Click it in order to bring up and start the file player. The filename extension must be MID or WAV.

It is also possible to play WAVs from the PRG editor, as well as from the WAVview window.

MID files are played using your soundcard's built-in MIDI synth chips. In order to get WAVmaker's wavetable sound, you must first render your MIDs to WAVs.

The player is controlled using the following buttons:



Note: WAVmaker uses the MS Windows Media Control Interface (MCI) to play MIDs and WAVs. Your sound card must have been properly installed under Windows for this to work. MID playback also requires the Windows MIDI Mapper to be correctly configured. Consult your Windows and sound card documentation in case of difficulties.

Scanning files

The **Scan** command allows you to obtain basic information about MID and WAV files.

In order to scan a file, select it in the file box on the main panel and click the **Scan** button. The filename extension must be MID or WAV.

It's a good habit to scan new files before doing anything else with them; if they pass the test, they will probably work just fine with WAVmaker.

- [Scanning MID files](#)
- [Scanning WAV files](#)

Scanning MID files

The MID **Scan** command causes WAVmaker to read the specified input file, assumed to be Standard MIDI Level 0 or Level 1, and to generate a listing (called **MIDscan.TXT**) in the Temp directory. The default text editor is then invoked to let you browse and/or print the listing.

Tip: Scan all your MIDs before letting the MID to LST and MID to WAV converters loose on them. If MID Scan can make sense of a MID, then chances are that the converters will cope, too (there are many things that can go wrong with MIDI files, especially if you download them from an Internet site or from a BBS). You will also find out exactly which notes must be covered by your PRG instrument definitions much quicker than by trial and error with MID to LST/WAV.

MIDscan.TXT contains the following information:

- MIDI file **name** and **type** (Level 0 or Level 1).
- A selection of events found, sorted by MIDI **track** and **channel** number:
 - **Time signature**
 - **Tempo**
 - **Program change**
 - **Note**
 - **Polyphonic pressure** (note numbers are listed)
 - **Modulation**
 - **Portamento time**
 - **Data entry**
 - **Volume**
 - **Panpot**
 - **Expression**
 - **Hold 1**
 - **Portamento**
 - **Sostenuto**
 - **Soft**
 - **Effect 1 depth** (Roland GS: reverb level)
 - **Effect 2 depth**
 - **Effect 3 depth** (Roland GS: chorus level)
 - **Channel pressure**
 - **Pitch bend**
 - **All sounds off**
 - **Reset all controllers**
 - **All notes off**
 - **Omni off**
 - **Omni on**
 - **Mono**
 - **Poly**
 - **System Exclusive**

Program numbers 0-127 are reported with the corresponding GM (General MIDI) instrument names. There is no GM standard for drum kits, so for Channel number 10 (the default drum channel) the following Roland Sound Canvas drum program assignments are used:

Number	Name
0	Standard Set
8	Room Set
16	Power Set
24	Electronic Set
25	TR-808 Set
33	Jazz Set
40	Brush Set
48	Orchestra Set

Scanning WAV files

The WAV **Scan** command cause WAVmaker to read the specified WAV file and to display a description containing the following information:

- [Word size](#) (8 or 16 bits).
- [Sampling rate](#) (11025, 22050 or 44100 Hz).
- [Number of sound channels](#) (1 or 2).
- [Duration](#) (number of samples per sound channel; milliseconds).
- [Min/Max sample values](#) (by sound channel).
- [Root Mean Square \(RMS\)](#) of sample values (by sound channel).
- [Saturated samples](#) (by sound channel).

The display is continuously updated while the file is being scanned. Hitting the **Cancel** button will abort the command and leave the partial results on screen. Click the **OK** button to close the WAV Scan window.

[If WAV Scan can read a sound file without error, chances are that the file will work fine with all WAVmaker commands.](#)

The RMS and the number of saturated samples are especially interesting if you are checking a new sample file or the results of a DSP operation:

- Many saturated samples (i.e. samples taking on the min/max values allowed by the word size employed) indicate too high a recording level, causing distortion.
- Too narrow a min/max range on the other hand gives you bad dynamic response and a low signal/noise ratio.

Ideally, you should try to use the full word size, just barely avoiding saturation. The Maximize command will scale up your WAV to full amplitude, but this is useful only to prevent further losses of precision when performing new operations on the sound; it certainly can't restore any information already lost!

Editing files

WAVmaker lets you modify four kinds of files:

- Text files (including PRGs and LSTs)
- Standard MIDI files (Level 0 and 1)
- PRG files
- WAV files

PRGs are listed twice because they can be treated as (structured) text files or edited with the PRGed editor.

In order to edit a file, select it from the main panel and click the **Edit** button. WAVmaker will try to guess which kind of editing functions you want to use (text, MIDI, PRG or WAV) based on the filename extension, and ask you to confirm (or modify) its choice. Click **OK** in the Edit dialog in order to launch the selected editor.

Editing text

In order to edit a text file (e.g. an LST or a PRG - but also see Editing PRGs for the latter), select it from the main panel and click the **Edit** button. If the filename extension is not PRG, MID or WAV, WAVmaker will suggest editing the file as text; if it doesn't, click the **Text** radio button in the **Edit as...** box of the Edit dialog. Launch the text editor by clicking the **OK** button.

WAVmaker doesn't come with its own text editor, since that would really be like reinventing the wheel! Instead, it allows you to invoke an external text editor with the selected file.

The default text editor invoked by WAVmaker is Windows' own Notepad.EXE. Unfortunately, Notepad can't handle large files (even in Windows 95 - hey, Microsoft, wasn't that supposed to be 32 bits?) so even modest LSTs are too much for it.

You can change the choice of text editor by clicking the **Text editor** button in the **Edit dialog**. This presents you with a standard file selection dialog; you can either type the name of the new editor or browse your way to it. WAVmaker will save this selection to disk upon exit and keep using it in future sessions.

If you do not already have a decent text editor for Windows, you may want to check out Alan Phillips' excellent **PFE** (Programmer's File Editor), which is [freely available](http://www.lancs.ac.uk/people/cpaap/pfe/) from the PFE Home Page <http://www.lancs.ac.uk/people/cpaap/pfe/>), as well as from many anonymous FTP sites around the world (e.g. CICA). If you can't find it, let Archie look for "pfe" or ask around on the Usenet group **comp.os.ms-windows.utilities**.

Editing MIDs

Any serious amount of MIDI editing requires a sequencer. However, one kind of operation on MIDs - [muting or isolating tracks and/or channels](#) - is so important, and so frequent (if you are creating music and not just listening to it), that it can be performed from within WAVmaker, without having to launch a dedicated sequencer. As of version 2.1, [you can also apply this function to other MIDI events than Note On/Off](#).

In order to edit a MIDI file, select it from the main panel and click the **Edit** button. If the filename extension is MID, WAVmaker will suggest editing the file as a MID; if not, click the **MID** radio button in the **Edit as...** box of the Edit dialog. Open the MID Edit window by clicking the **OK** button.

MID Edit window

In the MID Edit window

- select a command in the **Operation** box

The **Mute** command reads a Standard MIDI file (Level 0 or Level 1) and creates a file of the same type, substituting all selected events in the specified tracks and channels with Text events, effectively muting them.

The **Solo** command is complementary to Mute, i.e. all tracks and channels **except for the selected** ones are muted.

- check the events to be affected in the **Act on...** box.
- edit the track:channel list

Note that the track:channel list accepts the keyword **ALL** as a shortcut for specifying all 65535 tracks or all 16 channels. Keep in mind that the combination **ALL** tracks and **ALL** channels may result in a a trivial operation (like total silence, if you mute Note On/Off events).

- modify the destination filename (if needed)

By default, WAVmaker suggests the destination file **temp.MID** in the source directory. If the source file is called temp.MID, WAVmaker suggests **newtemp.MID**. **You are not allowed to overwrite the source file.**

- click the **OK** button to execute the command.

Using Mute and Solo, you can let WAVmaker render different parts of the same MIDI file separately, if need be using different PRG patch and/or WAV sample file banks, perform different DSP operations on the results, and finally mix the whole thing back down to one song. You can also use MID Edit to get rid of unwanted controller events without having to fire up your sequencer.

Editing PRGs

In order to edit a PRG file, select it from the main panel and click the **Edit** button. If the filename extension is PRG, WAVmaker will suggest editing the file as a PRG; if not, click the **PRG** radio button in the **Edit as...** box of the Edit dialog. Open the PRG Edit window by clicking the **OK** button.

Note that you can also edit PRGs as plain text files.

PRG Edit

PRG Edit is a dedicated editor for PRG files. While these can be modified using any text editor, they are NOT free format, so you should use PRG Edit with them unless you have a very good reason not to!

The best way to find out the meaning of each data field on a PRG data line is to select it in the data display grid and read PRG Edit's **Description**, above the comment box (on the lower half of the panel).

Navigating the grid is simply a matter of using your mouse and keypad as you would with any spreadsheet program. The three large buttons placed above the **Description** line, marked **Sample**, **Controller** and **Envelope**, are a further navigational aid: clicking them scrolls the grid sideways to the first column of the respective data section.



All changes made to the PRG file are saved upon exit, when you click the **OK** button. Clicking **Cancel** causes all changes to be discarded, leaving the file untouched.

One note about the **Key range** column is in place. As of version 2.0, WAVmaker implements layering: overlapping key ranges are not parsed exclusively, but inclusively, meaning that all patches (up to six) covering a key are actually used when that key is played. In other words, you can assign up to six independent patches (sample files and parameters) to each key; when the key is "pressed", they are all played in unison. This makes it easy to build complex instruments by combining several patches reacting differently to controller events (e.g. velocity and pressure).

It's also worth knowing that in a PRG file, lines starting by a # are comments. PRG Edit displays them separately from the actual data, in the comment box. Data lines preceded by comments are marked with a # in the second non-scrolling column of the data display grid.

>	#	Key range	WAV file	Base key
>	#	g10	GM6F6	f6
		f#10	GM6F#6	f#6

Editing comments is simply a matter of selecting a line in the data display and typing away in the comment box.

Note that the selected data display line is marked by a pointed bracket (>) in the first non-scrolling column of the data display grid.

The controls above the data display grid perform the following functions:



- **Insert** creates (and selects) a new data line in the grid, ABOVE the currently selected line, copying all its contents.

- **Add** creates (and selects) a new data line in the grid, BELOW the currently selected line, copying all its contents.
- **Delete** discards the currently selected line from the grid, and selects the next line (the previous one at the bottom of the grid). PRG Edit does not allow you to delete all lines in the grid; there must always be at least one data line.

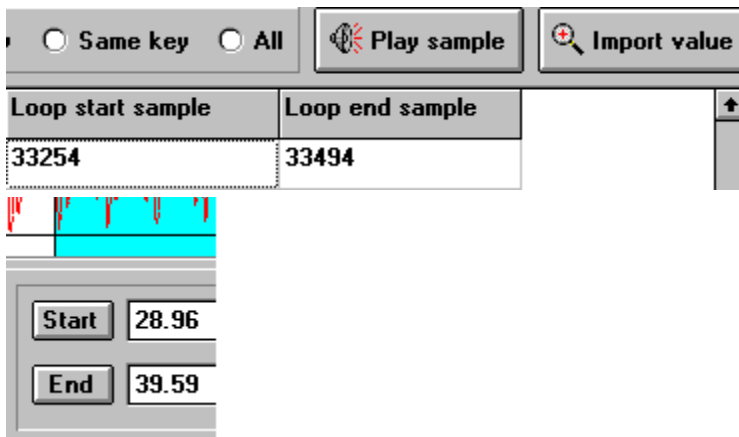


- **Copy to...** copies the value in the currently selected data grid field to the rest of the column, according to the criterium chosen with the radio button selectors **Same base key**, **Same key** and **All**.



- **Play sample** opens the aptly called Play sample window, which allows you to test what the patch will sound like when used with WAVmaker's MID to WAV and LST to WAV commands.
- **View sample** opens a modeless WAVView window on the sample file referred to by the currently selected PRG line, allowing you to switch freely between WAVView and PRG Edit.

While the WAVView window is open, this button carries the label **Import value**.



Clicking **Import value** while the **Start sample** or **Loop start sample** fields are selected in the data display grid causes the value in the **Start** edit box of the WAVView window to be copied into the grid (after a unit conversion, if needed).

When the **End Sample** and **Loop end sample** fields are selected in the grid, the effect depends on the current state of the WAVView window. If there is a selected range, the **End** value is imported. If there is no selected range, the **Start** value is imported.

If a length field (**Attack length** or **Decay length**) is selected in the data display grid, clicking **Import value** causes the difference between the **End** and **Start** values in the WAVView window to be imported. Note that PRG Edit uses integer length values. All decimals will be rounded off.

[Click here for an example of how to create a new PRG file!](#)

Play sample

The **Play sample** window allows you to test what the currently selected line (in the PRG Edit data display grid) will sound like when used with WAVmaker's MID to WAV and LST to WAV commands. It works by actually creating a temporary LST file in the Temp directory and then launching the LST to WAV converter with this file. This process is started by clicking the **Write file** button. In order to listen to the result, click **Play file**.

The remaining controls are fairly self-explanatory, and allow you to change some of the most important properties of the note event (as well as the destination file; occasionally, you may want to use this feature for WAV editing, e.g. resampling and envelope shaping). The only detail worthy of notice is [the **All layers** checkbox; checking this box causes WAVmaker to use all PRG data lines \(up to six of them\) covering the note to be played. By default, only the selected line is used.](#)

Close the Play sample window by clicking **OK**.

How to create a PRG file

The best starting point for the creation of a new PRG file is another PRG file, preferably one using the same [keyboard mapping](#).

Most PRGs in WAVmaker's GM sample library use sample files recorded at f3, f4, f5, f#3, f#4, f#5, and often also at f6 and f#6. Upon MID to WAV rendering, all other notes are obtained from these by resampling. The reason for having two adjacent base notes for each sampled octave is essentially mathematical: resampling by harmonic ratios (such as f to g or f# to g#) is much faster (for the same resulting sound quality) than resampling by inharmonic ratios (such as f to g# or f# to g). For even higher quality, you can use the mapping employed for the acoustic pianos, e.g 3.PRG (Honky-tonk): in this case, the full octave is obtained from d, d#, g# and a, by moving one note up or down from the appropriate base note.

If speed is paramount, you may want to try a few other mappings. Within the octave, the fastest interval is five semitones (as in f to a#). Resampling by a full octave is even faster - and needless to say, if you sample all notes you're going to need, there will be no need for any resampling at all! This is the usual situation when dealing with percussion sounds.

The most important message here is that you're not limited to any particular keyboard mapping. You can choose to work "MOD-style" if you wish, i.e. sample just one note, like c4, and derive all other notes from it. This only takes one data line, in which you declare the **Key range** to be [g10-c0](#) (more generally, remember that contiguous ranges sharing the same base key and other parameters need not be described key by key). If you only have a limited set of samples at your disposal, you may have to adapt the keyboard mapping accordingly.

Of course, most instruments (especially acoustic ones) do not only vary in pitch but also in timbre (i.e. in waveshape) when played at different notes (just think of an acoustic piano, played at opposite ends of the keyboard) . In order to reproduce this kind of behaviour, you'll need to spread out your base keys over a suitable range.

Anyway, in this example we're going to use the most common mapping, i.e. the one based on f and f#. Our template will be good old 6.PRG (Harpsichord).

Step1: In [WAVmaker's PRG directory](#), copy 6.PRG to TEMP.PRG.

We are not going to use a real program number (such as 48.PRG or D0.PRG) for our new PRG, since it's not meant for actual use. If we were redefining or replacing an existing PRG, we would first create a new directory for our alternate bank and copy the original files to the new directory. We would then work on the copies, keeping the originals as backups. It's easy to mess up a PRG, and re-creating it from scratch can be very painful!

Step 2: [Verify that you have the file GM48F3.WAV in the WAV directory](#). If not, [expand GM48F3.DFT](#).

We are going to use GM48F3.WAV in this example. More generally, you should check that the sample files which you intend to use are present in the current WAV directory before you set out to create a PRG for them.

Step 3: [Start up the PRG editor with TEMP.PRG](#).

Step 4: [Edit the comment field](#).

Before we do anything else, we should modify the leading comment field (there is one heading all GM library PRGs). This way, there will be no risk of confusion when we return to the file later.

As you know, data lines preceded by comments are marked with a # in the second non-scrolling column of the data display grid. Since the leading comment precedes any data, it's assigned to the very first data line, which is automatically selected when you start up the PRG editor. The leading comment is therefore already displayed and ready to be edited in the comment box (below the field **Description** line) where you should be seeing the following text:

```
#-----#  
# General MIDI Program Number 6: Harpsichord #  
#-----#  
#
```

The comment box is actually a simple text editor. Click it and [change the comment to](#) a suitable blurb, such as

```
#-----#  
# My first PRG! #  
#-----#  
#
```

The leading #s are mandatory. If you omit them, they will be inserted by the PRG editor when you save your new PRG. The trailing #s are just for looks, and may be left out.

Step 5: [Change sample files.](#)

Depending on our goals, this is a step which may be better tackled using a text editor. Remember, PRG files are just structured text files, so they can be edited that way too.

Suppose for instance that you want to change from Harpsichord to Strings sample files. The former are called GM6F3, GM6F#3, and so on. The latter are called GM48F3, GM48F#3, etcetera. Just about any text editor can do a global search-and-replace from GM6 to GM48 with only a couple of keystrokes - and *presto*, you're done! This is also worth keeping in mind when doing other large-scale PRG editing.

On the other hand, the dedicated PRG editor can do some things which a text editor can't, since it knows the meaning of the data. To see how, let's [scroll down to f3 in the Key Range column](#). The **WAV file** entry should read GM6F3.

- **Step 5.1:** [Change the f3 WAV file entry to GM48F3.](#)

Now f3 will be rendered using the GM48F3.WAV sample file (the WAV extension is always implied), but the other notes derived from f3 will still be Harpsichord, since they all have their own independent PRG lines. Do we have to edit them all manually, or use a text editor? No, all we have to do is...

- **Step 5.2:** [Verify that the radio button labeled Same base key \(over the data display grid\) is checked.](#) If it isn't, click it.
- **Step 5.3:** [Click the button labeled Copy to... \(immediately to the left of the radio button\).](#)

You should see the WAV file entry being changed to GM48F3 on all lines displaying f3 in the

Base key column. All other entries are left unchanged. You could have forced a global copy operation by checking the **All** radio button (especially useful for performance parameters such as controller sensitivities and envelope section lengths). When layering patches (i.e. referencing the same key on more than one line of the data grid) you may also find use for the **Same key** radio button.

Now we have a PRG which uses GM48F3 for a#3, g3, f3, d#3, d3, c3, and all corresponding notes for octaves 2, 1 and 0 (since those are all derived from octave 3). The rest is all Harpsichord. We could keep changing more **WAV file** entries, but what we've done will be enough for this example.

Step 6: Kill the old loop point settings.

6.PRG employs looping to get a decent release section out of the short GM6 WAVs. Now that we're using a different sample file, the old loop is no longer any good.

I find it useful to remove old loop point settings before I start looking for new ones - if I don't, I may easily forget to update all entries, leaving some of the old settings in place. Now that you know how to use the **Copy to...** button, you should have no problem with this operation: [scroll the data display grid sideways until you can see the **Loop start sample** entry for f3, change it to -1 \(meaning no loop point\) and copy it to all other lines sharing the same base key.](#) Note that the **Loop end sample** value for f3 is automatically changed to -1, too (an end point doesn't make much sense without a start point). [Complete the operation by selecting it and clicking **Copy to...** again.](#)

WARNING: If you save a PRG with some strange loop point settings, like -1 and 40703 (i.e. no start point but an end point) you'll have to fix the PRG with a text editor. The PRG editor will refuse to let you load a file containing obvious nonsense, and so will the MID to LST and MID to WAV commands! This may all seem unnecessarily harsh, but it actually makes it much easier to spot errors and to avoid wasting hours of rendering time due to bad PRGs.

Step 7: Find new loop points.

As anybody who's ever been involved with sampling will tell you, this can be a most frustrating task. You need to find two samples with similar (ideally, identical) values, separated in time by a multiple of the sound's wavelength. Since any reasonably interesting sample file will feature large envelope and timbre variations over time, you'll be tempted to use "tight" looping, i.e. to constrain the loop to a small region within which it's relatively easy to find corresponding samples - but if you do this, you'll suppress exactly those variations which make the sound interesting! 6.PRG gets away with tight looping because the loop is used to extend the low-volume release section, but in most cases, you'll use looping for the sustain section, which is central to how the sound is perceived.

Your best tool in this game is WAVView and its scaling/zooming controls. Here's what I do in a typical looping session:

- **Step 7.1:** [Select the **Loop start sample** entry for the base key](#) (in this case, f3).
- **Step 7.2:** [Click the **View sample** button to open a WAVView window.](#)
- **Step 7.3:** [Change the horizontal **Zoom** value in WAVView to 1.0.](#) This is useful for getting a first impression of the envelope and for identifying promising loop sections.
- **Step 7.4:** [Try marking various **Start** positions and use the crosshair to find one yielding a good fit to the amplitude at the corresponding \(candidate\) **End** position.](#)
- **Step 7.5:** [When there is a good candidate loop section, select it.](#)
- **Step 7.6:** [Transfer the **Start** and **End** sample positions from WAVView to the PRG Edit](#)

window by selecting the **Loop start sample** entry, clicking the **Import value** button, selecting the **Loop end sample** entry, and clicking the **Import value** button again.

- **Step 7.7:** Click the **Start** button in WAVview, so as to center the graphic display on the start sample.
- **Step 7.8:** Successively zoom in on the WAV while looking for a good loop start position. Some people will tell you that zero crossings are good loop points, but peaks and valleys are often better, since they are (near) zero-slope points, i.e. the waveshape is only changing slowly there. As a consequence, you're more likely to find at least one good fit at the opposite end of the loop section if you use peaks and valleys.
- **Step 7.9:** Mark the final (hopefully!) start point, making sure to align the horizontal line of the cross-hair with the sample value (the vertical position in the graph).
- **Step 7.10:** Click the **End** button in WAVview, so as to center the graphic display on the end sample.
- **Step 7.11:** Try finding the best fit to the **Start** sample. Move the end of the selected range there.
- **Step 7.12:** Repeat Step 7.6 (i.e. transfer the **Start** and **End** sample positions from WAVview to the PRG Edit window).
- **Step 7.13:** From the PRG Edit window, use the Play sample command to test your new loop points. Remember to set a sufficiently long duration value, so you get to hear the loop a few times over.
- **Step 7.14:** Sounds bad? Go back to Step 7.10, 7.7 or even 7.3, depending on how hopeless it sounds and on how many minor modifications you've already tried.
- **Step 7.15:** When it finally sounds OK, use the **Copy to...** button to update all **Loop start sample** and **Loop end sample** entries for the **Same base key**.
- **Step 7.16:** Make sure that the **End sample** values for the modified lines are equal to or larger than the corresponding **Loop end sample** values.
- **Step 7.17:** Use the **Play sample** command to check that the loop also sounds right for the resampled notes. If it doesn't, go to Step 7.14 (you didn't really think that you were done, now did you?)
- **Step 7.18:** Click the **OK** button in the PRG Edit window in order to save the new PRG (the file is not saved if you exit using the Cancel button).

If you think that this is a lot of work, imagine what it was like to create the full GM library! Anyway, try looping GM48F3. This is easier than most cases, in part because low notes have more samples per wavelength and so offer more points to choose from when looking for matches between loop start and end. When you get too sick of it all, you can look at 48.PRG for my suggestion.

Cheer up, though. After all, not all sounds need to be looped!

Step 8: Edit the other performance parameters.

There are quite a few of them, and I won't go through them all in detail. You can find out what they do by selecting them and reading PRG Edit's **Description** line. Experiment a bit if it still isn't clear to you. A few highlights:

- Raise the **Base volume** (the volume at which you want WAVmaker to believe that the sound was recorded) in order to have it play back (render) at a lower volume.
- Raise the **Base pan** (the stereo pan position at which you want WAVmaker to believe that the sound was recorded) in order to have it play back (render) further to the left.
- Set the **Start sample** to a value larger than zero if you want to skip a leading section of the WAV file.
- The **End sample** can be used to limit the max duration of un-looped sounds. If the sound

is looped, **End sample** must be equal to or larger than **Loop end sample**.

- Use the **Attack**, **Decay**, **Sustain slope**, **Release slope**, **Attack length** and **Decay length** fields to modify the envelope shape. These should all be fairly intuitive.

Editing WAVs

In order to edit a WAV file, select it from the main panel and click the **Edit** button. If the filename extension is WAV, WAVmaker will suggest editing the file as a WAV; if not, click the **WAV** radio button in the **Edit as...** box of the Edit dialog. Open the WAV Edit window by clicking the **OK** button.

Note that **WAVmaker does not allow WAV editing in the WAV directory**, as a safety measure against accidental loss of instrument sample data. If you want to edit an instrument sample file, copy it to the Temp directory and work on it there.

WAV Edit window

As a safety measure, [you are not allowed to overwrite the source file](#). The default destination filename suggested by WAVmaker is [temp.WAV](#) (in the source directory); if this is the name of the source filename, WAVmaker will suggest [newtemp.WAV](#). Both source and destination filename are freely editable, but WAVmaker will disable all editing controls if you specify the same file as both source and destination.

Before executing a WAV editing command, you can modify its scope:

- Edit the **Start** and **End** times (note that you can select the time unit, either milliseconds or [samples](#), in the **Time unit** box; the same unit will be used for any other time parameters which you may have to enter). By default, commands are applied for the entire duration of the source file.
- Select which channels to affect in the **Channel** box. This box is not active if the source file is monophonic. Note that you can use the [Mix](#) command in order to turn a mono sample file to stereo.

You could, for instance, time reverse only a section somewhere in the midst of a file, and only in one stereo channel.

[In order to view the source file, click the View button](#). Note that the button label changes to **Import**. Clicking this button while the [WAVView](#) window is open causes the **Start** and **End** values in the latter to be copied into the corresponding WAV Edit fields (after a unit conversion, if needed). WAV Edit opens WAVView as a modeless window, so you are free to switch back and forth between the two.

If the destination file exists (and is on a supported WAV format), you will also be able to use [the Swap button](#). This simply [swaps the source and destination filenames](#), making it easy to **View** the results of an operation. You can also employ this feature to recycle the source file as the destination for the next operation, when chaining several WAV editing commands (be careful not to overwrite the wrong file!).

WAVmaker implements the following WAV editing functions:

- [Low Pass](#) (low pass filter)
- [High Pass](#) (high pass filter)
- [Band Pass](#) (isolates a frequency band)
- [Notch](#) (suppresses a frequency band)
- [Ring Modulate](#) (amplitude modulation by a periodic function)
- [Vocoder](#)

- [Maximize](#) (maximizes amplitude)
- [Compress](#) (reduces the dynamic range)
- [Expand](#) (increases the dynamic range)
- [Fade](#) (ramps the amplitude up/down)
- [Gate](#) (suppresses low amplitude sections)
- [Gated Cut](#) (removes low amplitude sections)

- [Echo](#)
- [Reverb](#)
- [Extend](#) (inserts a silent section)
- [Cut](#) (removes a section)
- [Reverse](#) (time-inverts)

- Chorus
- Flange
- Mix (mixes/converts sample files)

Mix

Use the **Mix** command to merge and/or format convert WAV files.

Note that you can select an output format in any combination of 8/16 bits, 11025/22050/44100 Hz, mono/stereo, independently of the input sample format(s). You can also scale the volume of the left and right channel of each input file by an independent real factor (use negative factors for phase inversion).

Operation is straightforward: adding a new source file is simply a matter of selecting it in the filebox and clicking the **Add** button. As usual, you are not allowed to overwrite source files, and **WAVmaker will not let you add the destination filename to the source list**. If you change the destination filename (in the **To** edit box) to a list entry, the **OK** button will be disabled until you rectify the situation.

Use the **Remove** button to drop the currently selected entry from the list. There is always at least one entry (the **Remove** button is disabled when there is only one entry); after all, mixing nothing doesn't make much sense!

WAVView

The **WAVView** window can be opened from WAV Edit, by clicking the **View** button, or from PRGed, by clicking the **View sample** button. Either way, WAVView is opened as a modeless window, so you can switch freely between WAVView and the parent window.

First of all, you should know that WAVView's graphic display is politically correct: **the left channel is drawn in red** and **the right channel is drawn in blue**.

In order to scroll the graphic display over the length of the sample file, you can use the VCR-style buttons below the position gauge: the single arrow buttons scroll the display by one frame; the double arrow buttons by 10% of the file size. You can also center the display on the position shown in the **Start** and **End** edit boxes by clicking the corresponding buttons.

The frame size can be changed using the horizontal **Zoom** edit box. A zoom value of 1.0 will cause the whole file to be displayed. You can also edit the vertical zoom factor, in the upper right corner box (where you can also read off the vertical graphic cursor position). The original settings can be restored by clicking the **Reset** button. Note that this button becomes a **Stop** button during redraw, allowing you to abort it at any time.

In order to mark a position in the graphic display, simply click there. This will cause several things to happen:

- A cross-hair will appear, originating at the selected position. The horizontal line is especially useful when looking for loop points of equal amplitude, to be used in PRGed.
- The graphic cursor changes from a cross to a horizontal, bidirectional arrow (a "resize" cursor).
- The **End** button is activated, and the current (horizontal) graphic cursor position is now reflected in the **End** edit box.
- The contents of the **Start** edit box are frozen, reflecting the horizontal position of the cross-hair. You can reposition the cross-hair by editing the value in this edit box, and force the display to be centered on it by clicking the **Start** button.

Now you can do one of two things:

- In order to remove the cross-hair, click the mouse in a position preceding it, anywhere in the graphic display. Clicking the **Reset** button will also remove the cross-hair.
- In order to select a range, click the mouse in a position subsequent to the cross-hair, anywhere in the graphic display.

When you select a range

- The background color in the graphic display, from the cross-hair to the selected end position, is changed to Aqua.
- The graphic cursor changes from a bidirectional arrow to a standard Windows arrow.
- The contents of the **End** edit box are frozen, reflecting the horizontal end position of the selected range. You can resize the range by editing the value in this edit box, and force the display to be centered on it by clicking the **End** button.

- The **Zoom** button is activated. Clicking it will cause the graphic display to be fitted exactly to the selected range. Note that the **Reset** button becomes a **Stop** button during redraw, allowing you to abort it at any time.
- The **Play selected** button is activated. Clicking this button will cause the selected range to be played back on your sound card. Note that the button becomes a **Stop** button during playback.

In order to deselect a range, click anywhere in the graphic display (or click the **Reset** button).

The **Play shown** and **Play all** buttons are available at all times. Clicking them will cause the portion of the file shown in the graphic display and the whole file to be played back, respectively. Note that the buttons become **Stop** buttons during playback.

In order to close the WAVview window, hit the **OK** button.

Low Pass

The **Low Pass** command does just what its name implies: given a WAV, it performs low pass filtering on it, attenuating high frequencies.

The width of the filter is determined by the **HHF** parameter. HHF is short for Half Height Frequency, i.e. the frequency (in Hz) at which the amplitude factor will have fallen to 0.5 (from 1 at 0 Hz). Audio buffs will recognize this as the 3 dB point. The default HHF is extremely gentle: half the sampling frequency.

WAVmaker 2.0's Low Pass command used a one-pole filter. As of version 2.1, the code implements a higher order Butterworth filter, resulting in much better frequency resolution.

Note that you can let WAVmaker multiply the output signal by a constant **Amplification** factor, in order to compensate for the signal power being filtered out. It is better to use this feature than to amplify the resulting file in a second step, since there will be no precision loss (i.e. noise) induced by an intermediate truncation to integer resolution (in WAV files, samples are stored as integers).

High Pass

The **High Pass** command does just what its name implies: given a WAV, it performs high pass filtering on it, attenuating low frequencies.

The width of the filter is determined by the **HHF** parameter. HHF is short for Half Height Frequency, i.e. the frequency (in Hz) at which the amplitude factor will have fallen to 0.5 (from 1 at half the sampling frequency). Audio buffs will recognize this as the 3 dB point. The default HHF is extremely gentle: 0 Hz.

WAVmaker 2.0's High Pass command used a one-pole filter. As of version 2.1, the code implements a higher order Butterworth filter, resulting in much better frequency resolution.

Note that you can let WAVmaker multiply the output signal by a constant **Amplification** factor, in order to compensate for the signal power being filtered out. It is better to use this feature than to amplify the resulting file in a second step, since there will be no precision loss (i.e. noise) induced by an intermediate truncation to integer resolution (in WAV files, samples are stored as integers).

Band Pass

The **Band Pass** command does just what its name implies: given a WAV, it performs band pass filtering on it, attenuating frequencies outside of the specified frequency range.

WAVmaker 2.0's Band Pass command used a standard two-pole resonator. As of version 2.1, the code implements a higher order Butterworth filter, resulting in much better frequency resolution.

The **Center** and **Width** frequencies are expressed in Hz. At **Center \pm Width/2** the amplitude factor is approximately 0.5. To produce narrower filtering, lower the **Width** value.

Note that you can let WAVmaker multiply the output signal by a constant **Amplification** factor, in order to compensate for the signal power being filtered out. It is better to use this feature than to amplify the resulting file in a second step, since there will be no precision loss (i.e. noise) induced by an intermediate truncation to integer resolution (in WAV files, samples are stored as integers).

Notch

The **Notch** command is complementary to Band Pass: given a WAV, it performs band pass filtering on it, attenuating frequencies inside the specified frequency range. Ideally, mixing the output of a Notch and a Band pass operation on the same source should reproduce the original signal.

WAVmaker 2.0's Notch command used a standard two-pole resonator. As of version 2.1, the code implements a higher order Butterworth filter, resulting in much better frequency resolution.

The **Center** and **Width** frequencies are expressed in Hz. At **Center \pm Width/2** the amplitude factor is approximately 0.5. To produce narrower filtering, lower the **Width** value.

Note that you can let WAVmaker multiply the output signal by a constant **Amplification** factor, in order to compensate for the signal power being filtered out. It is better to use this feature than to amplify the resulting file in a second step, since there will be no precision loss (i.e. noise) induced by an intermediate truncation to integer resolution (in WAV files, samples are stored as integers).

Ring Modulate

Ring modulation is the process of multiplying a signal with a second, periodic one. A commonplace application is the creation of "robot" voices by the modulation of speech with a square wave.

The **Ring Modulate** command uses a sine wave as its default modulator, but this choice can be overridden with the **Square**, **Triangle** and **Saw** (wave) radio buttons (note however that the sharp "edges" present in the alternate waveshapes may induce audible clicks in the result).

The **Rate** parameter determines the frequency (in Hz) of the modulator.

Vocoder

WAVmaker implements no less than a 32-band stereo **Vocoder**!

What a vocoder does is [extract the formant envelope from a control signal and impose it on a carrier signal](#). A short description of the algorithm goes like this: take the control signal, run it through a set of band pass filters to partition it into adjacent frequency bands (in our case, up to 32 of them for each sound channel), read off the amplitude in each band; then take the carrier signal, partition it into frequency bands, too, and use the amplitudes you've read off from the control signal to modulate (amplify) the corresponding carrier bands.

WAVmaker 2.2's **Vocoder** is designed to give you full control over its operation. You can set the **Time Resolution** of the controller amplitude followers (note that too low a value will result in distortion; you want to modify the carrier's envelope, not its individual samples), the number of **Bands/channel** (1-32), the centers and widths of controller and carrier **Bands** (separately, allowing you to map e.g. a controller band centered at 10000 Hz and of width 1000 Hz to a carrier band centered at 3000 Hz and of width 2500 Hz) and even their weights (amplitude factors; think of this as having an equalizer cascaded after the vocoder proper).

The **Patch** box works as in the [Echo](#), [Reverb](#), [Chorus](#) and [Flange](#) windows. It allows you to save and retrieve arbitrarily named Vocoder settings. In order to add a new patch to the list, click the **Add** button, then edit the new name. You can copy the current settings to an existing patch by selecting it and clicking **Update**. The **Remove** button discards the currently selected patch (be careful!).

The default **Prescan & normalize** option causes the control signal to be analysed in two passes, with the first pass serving to determine the maximal amplitude in any band. The result is used to normalize the amplitudes during the second pass, thus maximizing the dynamic range of the destination file.

WAVmaker's Vocoder works by [suppressing](#) frequencies in the carrier (the **From** file specified in the [WAV Edit window](#)) to varying degrees. No frequencies are boosted, but you may specify an overall **Amplification** factor. It is better to use this feature than to amplify the resulting file in a second step, since there will be no precision loss (i.e. noise) induced by an intermediate truncation to integer resolution (in [WAV](#) files, samples are stored as integers).

A commonplace musical application of the vocoder is to use a human voice as the control signal and an instrument as the carrier, making the instrument "speak" (listen to Kraftwerk's LP "Man Machine" for some pioneering examples of this). Many interesting effects can be created with instruments serving both as carriers and as controllers. A simple way to add a lot of rhythmic life to e.g. a bass line is to let it be controlled by the kick and snare drums playing in the same piece.

Keep in mind that carrier and controller should contain useful amounts of overlapping frequencies for vocoding to work. Trying to control a bass with a cymbal is not a good idea! Most often, you'll want both controller and carrier to have broad spectra, or the audible result will just be a variation in volume - hardly exciting. This is what you'll get if you try to control a single instrument note with a human voice, for instance. Use a chord instead!

Maximize

The **Maximize** command finds the min/max sample values in the source WAV file and scales the output signal so as to use the full available word size (8 or 16 bits).

Cut, **Maximize** and Reverse are the only WAV editing commands requiring no other parameters than those given on the WAV Edit window. They are executed immediately upon selection.

Compress

The **Compress** command allows you to reduce the dynamic range (the difference between the lowest and the highest sound level) of a WAV.

The input signal is time-averaged prior to amplitude determination (in general, you want to affect the amplitude [envelope](#), not individual sample values) so you must also provide a **Time resolution** parameter. The default value works fine for most purposes. Too low a setting will tend to cancel the sound entirely; too high a setting will make compression inefficient.

Compression itself is simply the act of dividing the original signal by the amplitude of the filtered signal, raised to some power (the **Compression** parameter). Since this lowers the amplitude, you will probably want to rescale the output using the **Amplification** parameter. It is better to use this method than to amplify the resulting file in a second step, since there will be no precision loss (i.e. noise) induced by an intermediate truncation to integer resolution (in WAV files, samples are stored as integers).

Fade

The **Fade** command amplifies a WAV by a linearly varying, real factor. It allows the creation of smooth fade in/fade out effects. As a special case, you can use it for constant amplification and phase inversion (negative amplitude factor).

Expand

The **Expand** command allows you to either increase the dynamic range (the difference between the lowest and the highest sound level) of a WAV. One possible application is noise reduction.

The input signal is time-averaged prior to amplitude determination (in general, you want to affect the amplitude [envelope](#), not individual sample values) so you must also provide a **Time resolution** parameter. The default value works fine for most purposes. Too low a setting will tend to cancel the sound entirely; too high a setting will make expansion inefficient.

Expansion itself is simply the act of multiplying the original signal by the amplitude of the filtered signal, raised to some power (the **Expansion** parameter). Since this raises the (max) amplitude, you will probably want to rescale the output using the **Amplification** parameter. It is better to use this method than to amplify the resulting file in a second step, since there will be no precision loss (i.e. noise) induced by an intermediate truncation to integer resolution (in WAV files, samples are stored as integers).

Gate

The **Gate** command silences sections of a WAV falling below a given amplitude **Threshold**. Abrupt clipping is avoided by allowing non-zero **Attack duration** (fade in) and **Decay duration** (fade out) times. Great for removing background noise in low/silent sections!

Keep in mind that the max amplitude of an 8 bit file is 128; that of a 16 bit file is 32768.

Gated Cut

The **Gated Cut** command removes samples below the specified **Threshold** level (amplitude) from the input WAV.

Used as an effect, it produces a distinctive "ripping" sound (not unlike what can be obtained by moving a pick-up across the grooves of a vinyl record, if you can still remember those!).

When the **Prune** box is checked, WAVmaker will perform semi-automatic removal of leading/trailing silence/noise from the file, by cutting only the first and the last sub-threshold section. This can be very useful when editing new sample files.

Keep in mind that the max amplitude of an 8 bit file is 128; that of a 16 bit file is 32768.

Echo

The **Echo** command does just what its name implies: given a WAV, it adds echo(es) to it.

The **Level** parameter controls the effect vs. dry sound output mix. At 100%, the first echo has the same volume as the original signal.

The **Feedback** parameter controls the percentage of the output signal to be led back to the input. A larger value yields more (and louder) echoes. Negative feedback values cause echoes to be phase inverted relative to the original signal. Setting **Feedback** to zero yields no echoes at all.

The **Delay** parameter determines the time between echoes.

The **Patch** box allows you to save and retrieve arbitrarily named Echo settings. In order to add a new patch to the list, click the **Add** button, then edit the new name. You can copy the current settings to an existing patch by selecting it and clicking **Update**. The **Remove** button discards the currently selected patch (be careful!).

Note: The destination file size is always equal to the source file size. [In order to avoid abrupt clipping of the echo effect, use the Extend command to add a suitable silent section to the source before applying Echo.](#)

Reverb

The **Reverb** command simulates the effect of multiple, random reflections (echoes) of the original sound. This is a fairly messy process, witnessed in part by the large amount of controllable parameters. As of version 2.1, you can choose between four different reverb algorithms (in the **Algorithm** box), tuned to different needs.

- The **Chamber** algorithm is optimized for small room simulation (400-600 ms decay).
- The **Chapel** algorithm is optimized for medium size room simulation (700-1300 ms decay).
- The **Hall** algorithm is optimized for large room simulation (1300-3000 ms decays).
- The **Custom** algorithm is a development of the classical Schroder/Moorer-style algorithms implemented by CSound, CMix etc. It is most useful for short decay times (80-500 ms) and adds a set of programmable **Low pass filters** simulating the absorption of high frequency sound components by the medium of propagation. There are six parallel effect lines for each sound channel, and you can control an independent low pass filter in each one of them. Generally speaking, a higher setting results in more attenuation (and therefore also in a shorter perceived reverberation time, which you may want to compensate for using the **Decay** setting) and a more muffled sound. Conversely, low settings give you longer, "tinnier" reverberations.

[The low pass filter settings only affect the Custom algorithm.](#)

When using the Custom algorithm, the **Decay** duration parameter should not be taken too literally, since the actually perceived length of the effect is strongly affected by the low pass filter settings.

The **Delay** parameter, on the other hand, can always be taken verbatim: it is the time (if any) before the effect onset. Most often, you'll want to leave it set to zero.

The **Effect level** settings determine the reverberation level relative to the original signal. The additive nature of the algorithm renders it prone to saturation, and speed considerations make it impractical to perform amplitude clipping at each step. [If you get strange sounding \(and looking; see WAVView\) results, lowering these levels is the first thing to try.](#)

The **Patch** box allows you to save and retrieve arbitrarily named Reverb settings. In order to add a new patch to the list, click the **Add** button, then edit the new name. You can copy the current settings to an existing patch by selecting it and clicking **Update**. The **Remove** button discards the currently selected patch (be careful!).

Note: The destination file size is always equal to the source file size. [In order to avoid abrupt clipping of the reverb effect, use the Extend command to add a suitable silent section to the source before applying Reverb.](#)

Extend

The **Extend** command inserts silent sections into WAV files. It is particularly useful for extending the trailing silence at the end of a WAV before applying Echo and/or Reverb (hence the name). This allows a natural decay rather than a sudden clipping.

Checking the **Append** box forces WAVamker to put the silent section at the exact end of the original sound data.

Cut

Given a WAV and a time range, the **Cut** command removes the latter from the former. **The Channels selection has no effect on this operation.**

Cut, Maximize and Reverse are the only WAV editing commands requiring no other parameters than those given on the WAV Edit window. They are executed immediately upon selection.

Reverse

The **Reverse** command is simplicity itself: given a WAV, it time-reverses it. The popular Reverse Cymbal sound (General MIDI program number 120) is nothing but an ordinary cymbal treated this way.

Cut, Maximize and Reverse are the only WAV editing commands requiring no other parameters than those given on the WAV Edit window. They are executed immediately upon selection.

Chorus

The **Chorus** effect is obtained by mixing a WAV with a delayed version of itself, with the delay going through a periodical variation. The **Variation** and the **Effect rate** should be kept fairly small (1 ms or less and no more than a few Hz, respectively).

The **Original level** and the **Effect level** are simple amplitude factors (you can use negative values for phase inversion). You can also introduce a constant **Delay offset**, although you'll probably want to leave this parameter set to zero most of the time.

The **Patch** box allows you to save and retrieve arbitrarily named Chorus settings. In order to add a new patch to the list, click the **Add** button, then edit the new name. You can copy the current settings to an existing patch by selecting it and clicking **Update**. The **Remove** button discards the currently selected patch (be careful!).

Flange

Flange has a lot in common with Chorus. Like the latter, it is obtained by mixing a WAV with a delayed version of itself, with the delay going through a periodical variation. However, there is an additional complication in the form of a **Feedback** loop. Large feedback values can yield truly warped sounds. In more moderate amounts, you get the effect which made Nirvana sound like Nirvana.

As with Chorus, the **Variation** and the **Effect rate** should be kept fairly small (1 ms or less and no more than a few Hz, respectively).

The **Original level** and the **Effect level** are simple amplitude factors (you can use negative values for phase inversion). You can also introduce a constant **Delay offset**, although you'll probably want to leave this parameter set to zero most of the time.

The **Patch** box allows you to save and retrieve arbitrarily named Flange settings. In order to add a new patch to the list, click the **Add** button, then edit the new name. You can copy the current settings to an existing patch by selecting it and clicking **Update**. The **Remove** button discards the currently selected patch (be careful!).

Converting files

In order to apply the file conversion commands, select the source file in the file box on the main panel and click the **Convert** button. Valid filename extensions are MID, LST, WAV and DFT.

When the Convert panel is first opened, WAVmaker will try to guess which conversion you wish to perform and suggest a destination filename in the appropriate directory. Note that you can edit both filenames, and that the destination directory and filename extension are changed automatically if you select a different conversion function.

Start the selected conversion by clicking the **OK** button.

The following conversions can be performed:

- MID 1 to 0 Standard MIDI Level 1 to Standard MIDI Level 0
- MID to LST Standard MIDI Level 0/1 to LST (WAVmaker playlist)
- LST to WAV LST (WAVmaker playlist) to WAV
- MID to WAV Standard MIDI Level 0/1 directly to WAV
- DFT to WAV DFT (compressed sample file) to WAV
- WAV to DFT WAV to DFT (compressed sample file)

See also the Make WAVs command (expands all sample files referenced by a PRG file, from DFT to WAV, in one shot).

The **MID 1 to 0** command merges the up to 65535 tracks in a Standard MIDI Level 1 file to the single track in a Standard MIDI Level 0 file. After all, while WAVmaker can handle both Level 0 and Level 1 files, you may want to use it along with some other MIDI software which can't.

The default destination path is the same as the source path, with a "0" appended to the filename (or substituted for the last character, if the source filename is eight characters long - watch out for name conflicts if the source filename is something like "myfile00.mid").

Unlike most other WAVmaker commands, MID 1 to 0 does not display a progress window, since the conversion is pretty fast on most machines.

MID to LST

The **MID to LST** command parses a Standard MIDI file (Level 0 or Level 1) and creates an LST (WAVmaker playlist) file.

In order to do this, WAVmaker needs the PRG files corresponding to the MIDI programs (instruments) used in the MID: if the MID uses programs 3 and 5, for instance, the PRG files 3.PRG and 5.PRG must be present in the current PRG directory.

WAVmaker also needs to know which MIDI channel(s) in which track(s) to use for drums. The GM (General MIDI) standard reserves **channel 10 on all tracks** for drums, and this is also the default setting suggested by WAVmaker when you click **OK** in the Convert panel. You can change this setting by editing the **Drum Track:Channel** list. The keyword **ALL** is accepted for both tracks and channels (internally, it's represented by the value **-1**, which you can also enter directly if you so prefer).

WAVmaker does not allow you to remove all drum track:channel entries in the list: there is always at least one entry. If you don't want a drum track, set this entry to an unused and/or out-of-range track:channel combination, such as 0:0.

Note that the MID to LST panel allows you to edit both the source and the destination file path. The Song edit box is empty and can not be edited, since no song WAV is created by this command.

Unlike most other WAVmaker commands, MID to LST does not display a progress window, since the conversion is pretty fast on most machines.

LST files are not playable; they must be rendered to WAV by the LST to WAV command before you can listen to the song. Most often, you'll probably want to perform the whole chain from MID to LST to WAV in one shot, using the MID to WAV command. The MID to LST command allows you to edit the intermediate LST file manually (using any text editor) for greater control over the rendering process.

LST to WAV

The **LST to WAV** command renders an LST (WAVmaker playlist) file to a WAV file.

In order to do this, WAVmaker needs the sample files referenced by the LST file. If the LST file refers to instruments GM0A4 and CLAVES, for instance, then sample files GM0A4.WAV and CLAVES.WAV must be present in the current WAV directory. If the necessary WAVs are not present, WAVmaker will try to locate the corresponding DFTs in the DFT directory and expand them prior to rendering.

Note that you can select an output format in any combination of 8/16 bits, 11025/22050/44100 Hz, mono/stereo, independently of the input sample format(s). You can also scale the volume by a constant factor (especially useful if you notice that saturation of the available dynamic range is causing distortion in the output).

It is usually not a good idea to let the LST to WAV converter create 8 bit output using 16 bit sample files. Sample mixing will then be performed in 8 bit precision, after discarding the low end byte of the sample. As this will typically leave you with just 3-4 significant bits, the result will be excessively noisy. A much better way is to create a 16 bit file and then convert it to 8 bit format using e.g. the Mix command.

Click the **OK** button on the Convert panel in order to start the rendering process. WAVmaker will display a current status & progress window throughout the operation. You can abort at any time by clicking the **Cancel** button on this status window (allow for some reaction time - WAVmaker will not check the buttons in the midst of heavy operations). Close the status window by clicking the **OK** button. The **Help** button calls up a list of explanations to error messages and of possible causes/solutions.

Most often, you'll probably want to perform the whole chain from MID to LST to WAV in one shot, using the MID to WAV command. The MID to LST command allows you to edit the intermediate LST file manually (using any text editor) for greater control over the rendering process.

LST to WAV Errors

- **Out of memory!**

Hopefully self-explanatory. Try closing any other open applications; get that RAM upgrade you've been considering...

- **Illegal duration!**

WAVmaker can't make any sense out of the duration value on the first line of the LST file (which should be in milliseconds, an **integer**, and of course **greater than zero**).

- **Disk full!**

Time to clean up your hard disk - or to buy a larger one.

- **"{" expected!**

Been editing that LST, have we? New Sample Blocks should be opened by a {.

- **Unexpected end of file!**

The LST ends in an unexpected way (with a note still playing; with a Note Block still open; with a Sample Block still open).

- **"[" expected!**

New Note Blocks in LST files should be opened by a [.

- **Illegal note value!**

The legal note range in an LST file is **-127 to 255**.

- **Illegal base note!**

The legal base note range in an LST file is **0 to 127**.

- **Illegal base volume!**

The base volume in an LST file must be **1 or greater**.

- **Illegal base panpot!**

The legal base panpot range in an LST file is **0 to 127**.

- **Illegal velocity sensitivity!**

The legal velocity sensitivity range in an LST file is **-100 to 100**.

- **Illegal pressure sensitivity!**

The legal pressure sensitivity range in an LST file is **-100 to 100**.

- **Illegal expression sensitivity!**

The legal expression sensitivity range in an LST file is -100 to 100.

- **Illegal release sensitivity!**

The legal release sensitivity range in an LST file is -100 to 100.

- **Illegal bend range!**

The legal bend range in an LST file is 0 to 127 (semitones).

- **Illegal modulation sensitivity!**

The legal modulation sensitivity range in an LST file is 0 to 100 (%).

- **Illegal modulation rate!**

The legal modulation rate range in an LST file is 0 to 100 (Hz).

- **Illegal attack end factor!**

The attack end factor must be 0 or greater in an LST file.

- **Illegal decay end factor!**

The decay end factor must be 0 or greater in an LST file.

- **Illegal attack length!**

The attack length in an LST file must be an integer, 0 or greater (milliseconds).

- **Illegal decay length!**

The decay length in an LST file must be an integer, 0 or greater (milliseconds).

- **Illegal start sample position!**

The start sample position value in an LST file must be 0 or greater.

- **Illegal end sample position!**

The end sample position value in an LST file must be equal to or greater than the start sample position value.

- **Illegal loop start position!**

The loop start sample position value in an LST file must be EITHER

- equal to or greater than the start sample position value AND smaller than or equal to the end sample position value; OR
- smaller than zero (no loop).

- **Illegal loop end position!**

The loop end sample position value in an LST file must be EITHER

- equal to or greater than the loop start position value AND smaller than or equal to the end sample position value; OR
- smaller than zero (no loop).

- **Sample file read failure!**

One of the WAV sample files referenced by the LST can't be found (check that it really is present in the current WAV directory); or there is something wrong with its header; or it's on a format not supported by WAVmaker (try running a WAV Scan on it).

- **Illegal event code!**

The LST file contains an unrecognized event code.

- **Illegal release slope!**

If there is a loop (i.e. the loop start position is 0 or greater), the release slope must be **negative** and sufficiently **large** to remain non-zero when translated to a delta per sample in the 8 decimal bit, fixed point format used internally by WAVmaker. **If you get this error message, try using a steeper, negative slope.**

- **Internal error!**

Uh oh... try to narrow down the offending LST file section and the conditions under which the error occurs; then contact me!

MID to WAV

The **MID to WAV** command performs the MID to LST to WAV conversion process in one shot.

When executing this command, WAVmaker first parses the specified Standard MIDI file (Level 0 or Level 1) and creates an LST (WAVmaker playlist) file; if this process succeeds, it then renders the LST to a WAV file.

Note that you can select an output format in any combination of 8/16 bits, 11025/22050/44100 Hz, mono/stereo, independently of the input sample format(s). You can also [scale the volume](#) by a constant factor (especially useful if you notice that saturation of the available dynamic range is causing distortion). These operations must be performed from the Convert panel, before the MID to WAV command is launched by clicking the **OK** button.

It is usually not a good idea to let the MID to WAV converter create 8 bit output using 16 bit sample files. Sample mixing will then be performed in 8 bit precision, after discarding the low end byte of the sample. As this will typically leave you with just 3-4 significant bits, the result will be excessively noisy. A much better way is to [create a 16 bit file and then convert it to 8 bit format using e.g. the Mix command](#).

- In order to create the LST, WAVmaker needs the PRG files corresponding to the MIDI programs (instruments) used in the MID: if the MID uses programs 3 and 5, for instance, the PRG files 3.PRG and 5.PRG must be present in the current PRG directory.

WAVmaker also needs to know which MIDI channel(s) in which track(s) to use for drums. The GM (General MIDI) standard reserves [channel 10 on all tracks](#) for drums, and this is also the default setting suggested by WAVmaker when you click **OK** in the Convert panel. You can change this setting by editing the **Drum Track:Channel** list. The keyword [ALL](#) is accepted for both tracks and channels (internally, it's represented by the value **-1**, which you can also enter directly if you so prefer).

[WAVmaker does not allow you to remove all drum track:channel entries in the list: there is always at least one entry. If you don't want a drum track, set this entry to an unused and/or out-of-range track:channel combination, such as 0:0.](#)

Note that the MID to WAV panel allows you to edit the full paths of all three (MID, LST and Song WAV) files involved.

No progress window is displayed while the LST is being created, since the conversion is pretty fast on most machines.

- In order to create the Song WAV, WAVmaker needs the sample files referenced by the LST file. If the LST file refers to instruments GM0A4 and CLAVES, for instance, then sample files GM0A4.WAV and CLAVES.WAV must be present in the current WAV directory. If the necessary WAVs are not present, WAVmaker will try to locate the corresponding DFTs in the DFT directory and expand them prior to rendering.

WAVmaker will display a current status & progress window throughout the rendering process. You can abort at any time by clicking the **Cancel** button on this status window (allow for some reaction time - WAVmaker will not check the buttons in the midst of heavy operations). Close the status window by clicking the **OK** button. [The Help button calls up a list of explanations to error messages and of possible causes/solutions.](#)

The **DFT to WAV** command expands compressed DFT files, created by WAV to DFT, back to playable WAV files.

Note: this command expands one DFT file at a time. In order to expand all sample files referenced by a PRG file in one shot, use the Make WAVs command from the main panel.

Unfortunately, the algorithm employed is not suitable for the disk-based approach used by most other WAVmaker commands. The full WAV file must fit in the available memory. Needless to say, larger sample files require more memory. The word size employed in the WAV file makes no difference as far as memory requirements are concerned, but it is reflected by the size of the compressed file.

Unlike most WAVmaker commands, **DFT to WAV** does not display a progress gauge, since execution is pretty fast on most machines.

Important note for users of WAVmaker 1.0: The 8 bit DFT format has changed. WAVmaker 2.x will therefore not be able to expand 8 bit DFTs created with WAVmaker 1.0. The 16 bit format is unchanged.

The **WAV to DFT** command performs lossy data compression on sound files conforming to the WAV subset used by WAVmaker. The compression ratio depends on the characteristics of the sound. For isolated instrumental notes, you can get good quality from DFT files ZIPped (or otherwise archived with standard lossless compressors) [to less than 1/50 of the original WAV filesize!](#)

Unfortunately, the algorithm employed is not suitable for the disk-based approach used by most other WAVmaker commands. The full WAV file must fit in the available memory. Needless to say, larger sample files require more memory. The word size employed in the WAV file makes no difference as far as memory requirements are concerned, but it is reflected by the size of the compressed file.

Unlike most WAVmaker commands, **WAV to DFT** does not display a progress gauge, since execution is pretty fast on most machines.

[Important note for users of WAVmaker 1.0:](#) The 8 bit DFT format has changed. WAVmaker 1.0 will therefore not be able to expand 8 bit DFTs created with WAVmaker 2.x. The 16 bit format is unchanged.

Make WAVs

In order to expand all sample files referenced by a PRG file, from DFT to WAV, select the PRG file from the main panel. The label on the **Convert** button will change to **Make WAVs**, and the button will be activated. Unlike most other WAVmaker commands, **Make WAVs** will act on all selected files (the other commands supporting multiple file selections are **Copy**, **Move** and **Delete**).

When you click the **Make WAVs** button, you are presented with a list of all sample files referenced by the PRG(s). Filenames already in use in the current WAV directory (with the implied WAV filename extension) are preceded by the word **No** (in the column labeled **Make**); unused filenames are preceded by the word **Yes**. Only the latter filenames will be acted upon.

You can toggle between **Yes** and **No** simply by clicking on the appropriate cell in the **Make** column. When you're done, click the **OK** button to execute the command.

WAVmaker will scroll down the list as the DFTs are expanded. [You can abort execution at any time by clicking the **Cancel** button.](#) Allow for some reaction time: WAVmaker will complete the expansion of the current sample file before terminating the **Make WAVs** command.



At the time of writing, the best way to reach me is by **e-mail** to

Tommy.Anderberg@abc.se

The second best choice is by **fax** to

+46 8 591 162 23

The slowest way is by **snail mail** to

Tommy Anderberg
Almvagen 4
195 44 Marsta
Sweden

Acknowledgments

I am grateful to all those who volunteered as Beta testers for WAVmaker 2.0. Special thanks go to

- [Steve Monett](mailto:steve.monett@cho.ge.com) (steve.monett@cho.ge.com)
Charlottesville, Virginia, USA
"I am willing to answer questions about using WAVmaker with Windows NT."

(With immense patience, Steve also reviewed the WAVmaker 2.1 and 2.2 pre-releases).
- [Jerry Fielden](mailto:fielden@odyssee.net), hard rocker & composer (fielden@odyssee.net)
"Rock on and never let go of your goals!"
- [Alex Prooper](mailto:alexp@xs4all.nl) (alexp@xs4all.nl)
"Information about the CD Regular Music, which is produced completely with a sound synthesis program, can be obtained from Alex Prooper, Koningstraat 42, 1941 BE Beverwijk, The Netherlands."
- [Joost Nieuwenhuijse](mailto:J.Nieuwenhuijse@et.tudelft.nl) (J.Nieuwenhuijse@et.tudelft.nl)
Home Page: <http://elektron.et.tudelft.nl/~joostn>
- [David Eriksson](mailto:oden.eriksson@mailbox.swipnet.se) (oden.eriksson@mailbox.swipnet.se)
Norrköping, Sweden
Home Page: <http://www.geocities.com/SiliconValley/1535/>
"Internet Addiction is a serious matter. I know."

All bugs still present in WAVmaker 2.x after these gentlemen were done with version 2.0 are of course mine and mine alone!

MIDI archives on the Net

This is a slightly edited version of [Piet van Oostrum's excellent list of Internet MIDI archives](#).

Version 1.144 1995/12/12 14:49:55

Note: the latest version of this file is available from the [ftp.cs.ruu.nl](ftp://ftp.cs.ruu.nl) archive as [MIDI/DOC/archives](#) (see below how to access the archive) and on the various [news.answers archives](#). This archive also contains a lot of midi music and programs. Mirrors of the [MIDI/SONGS](#) directory or the whole archive can be found on the following sites:

- [ftp.cis.nctu.edu.tw](ftp://ftp.cis.nctu.edu.tw) /MIDI/SONGS/
- [ftp.pu-toyama.ac.jp](ftp://ftp.pu-toyama.ac.jp) /pub/ftpmail/ftp.cs.ruu.nl/pub/MIDI/SONGS
- [ftp.monash.edu.au](ftp://ftp.monash.edu.au) /pub/midi/SONGS
- [ftp.funet.fi](ftp://ftp.funet.fi) /pub/sounds/midi
- [ftp.ibp.fr](ftp://ftp.ibp.fr) /pub/midi

A large collection of **MIDI files** can be found on wuarhive.wustl.edu, in [/systems/ibmpc/ultrasound/sound/midi/files](#). Some of these files are specific for the Gravis Ultrasound card but most are also usable for other synthesizers.

Another archive with many MIDI files is dbp.caltech.edu/pub/songs. There is an FTP archive at [ftp.ucsd.edu](ftp://ftp.ucsd.edu) [128.54.16.7]. It contains a.o. MIDI files, patches and a few programs. See the directory [midi](#). This archive looks rather out of date, by the way.

There is a demo of **Symbolic Composer** at

- media-lab.media.mit.edu [18.85.0.2] [music/midi](#)
- sumex.stanford.edu [171.65.4.3] [/info-mac/Application/\(symbolic-composer-*\)](#)

It can probably also be found on other info-mac archives. More Symbolic Composer stuff can be obtained from Fokke de Boer by e-mailing to Fokke.de.Boer@rivm.nl a message with **Subject: retrieve directory** and following the instructions that you will receive.

atari.archive.umich.edu [141.211.120.11] contains Atari stuff in directory [/atari/Music](#).

scam.berkeley.edu [128.32.138.1] contains netjam submissions on midifiles in directory [/pub/misc/netjam/submissions/](#).

Sites for **MIDI Sample Dump Standard** programs and samples:

- alf.uib.no [129.177.30.3] [/pub/sds](#)
- sweaty.palm.cri.nz [161.66.1.11] [sds](#)

wagner.musicnet.ua.edu [130.160.156.75] is a site that just started to collect music stuff. It has patches for **Roland JV-80/880/1000** in directory [/pub/music/patches/jv80](#). Another FTP site for the JV's (and Proteus) is [ftp.pacificrim.net:Pub/Synth/](ftp://ftp.pacificrim.net:Pub/Synth/). WWW page for the JV synths: <http://www.damp.apana.org.au/jv80/>.

oak.oakland.edu [141.210.10.117] has **EPS/EPS-16+/ASR** info in [pub/eps/docs/](#). There are some programs for the EPS in [/pub/eps/utills](#) (Atari ST, Amiga, MAC, MS-DOS). See [pub/eps/docs/utills](#) for info. Feel free to access oak.oakland.edu using Gopher (gopher.acs.oakland.edu, port 70) or WWW (URL: <http://www.acs.oakland.edu>).

FTP sites for the **Kurzweil K 2000**:

- <ftp.uwp.edu:/pub/music/lists/kurzweil> (maintained by jbuckman@aas.org)
- <bach.nevada.edu:/pub/k2000> (maintained by johann@nevada.edu)

WWW site for the Kurzweil K2000: <http://metro.turnpike.net/kcowgill/index.html>.
 WWW site for the Kurzweil company: <http://www.musicpro.com/kurzweil/>.

bach.nevada.edu also contains **christian music MIDI files** in </pub/midi>.
qiclab.scn.rain.com in </pub/music> contains various **conversion programs**.

Roland Samplers:

There is an FTP archive that includes introductory information, useful utilities (read Roland disks on your computer), archives of the mailing list, and some samples. The archive is on lotus.UWaterloo.ca in the [pub/sgroup](/pub/sgroup) directory.

There also is an FTP site with **TX16W** samples, maintained by one of the TX16W list-members. FTP to <ftp.informatik.uni-freiburg.de> [132.230.150.1], the samples are in the directory [pub/sounds/tx16w/samples](/pub/sounds/tx16w/samples). There are also back issues of the TX16W newsletter in [pub/sounds/tx16w/mail](/pub/sounds/tx16w/mail). A WWW entry is <http://ls7-www.informatik.uni-dortmund.de/html/tx16w.html>.

Archives of the **Music-Research Digest** can be found at cattell.psych.upenn.edu [12128.91.2.173] in directory [pub/Music.Research](/pub/Music.Research).

impaqt1.mem.drexel.edu [129.25.10.1] has a couple of MS-DOS files in [pub/files/ibm/midi](/pub/files/ibm/midi).

There is an **EMAX** (I and II) FTP site at sweaty.palm.cri.nz (161.66.1.11) some samples, FAQ, archives of mailing list and some software to allow sample interchange for PC compatibles.

Cakewalk files can be found at <ftp.funet.fi> [128.214.6.100] in [pub/msdos/sound/cakewalk](/pub/msdos/sound/cakewalk) and also on ucsd.edu (see above). More Cakewalk stuff (including archives of the Cakewalk mailing list) can be found at <ftp.cs.colorado.edu:/users/mccreary/archive/cakewalk> or on WWW as <http://www.cs.colorado.edu/~mccreary/archive/cakewalk>. There is a **VFX** archive in the parallel directory [mccreary/archive/vfx](/mccreary/archive/vfx). Another site for Cakewalk is mort.isvr.soton.ac.uk:/pub/pc/cakewalk (FTP) or <http://www.isvr.soton.ac.uk/People/ccb/Cakewalk> (WWW).

There are several **MIDI programs for MS Windows** in <ftp.cica.indiana.edu:/pub/pc/win3/sounds>.

OS/2 programs can be found on <ftp-os2.cdrom.com> [192.153.46.2]. Directories /pub/os2/2_x/mmedia and /pub/os2/2_1/mmedia contain the bulk of the MIDI-related files for OS/2. (There is a specific FTP site for OS/2 multimedia in the formative stages right now, but it is not yet in operation.) For a full index of OS/2 files at that site, retrieve </pub/os2/00index.txt>. Submissions may be placed in </pub/os2/incoming>.

mac.archive.umich.edu [141.211.165.41] has a **Macintosh** midi archive in [mac/sound/midi](/mac/sound/midi). Other Macintosh sites: See the file Mac-archives in this directory.

An FTP site for predominantly **D-70**-related files: </pub/D70> on kilroy.jpl.nasa.gov.

<ftp.neuroinformatik.ruhr-uni-bochum.de>; in directory </pub/outgoing/heja/sy-list> there are several infos/sounds/programs of interest for **Yamaha SY** users (this is the "inofficial" ftp-site of the SY list). **IMPORTANT NOTE:** go directly to the above mentioned directory! Access via WWW: <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/heja/sy-l>

[ist.html](#). The site contains mostly original material and is not a mirror of the big music archives.

Yamaha VL1 info and patches are available by WWW on <http://www.sky.net/~kennyb/vl1sounds.html>.

<ftp.ecn.nl:/pub/RP-1>: **DIGITECH RP-1** MIDI guitar effect pedals and some other MIDI stuff.

Apple 2 MIDI Synth archives can be found on cco.caltech.edu [[131.215.6.10](#)] in the pub/apple2/music/synthlab/ directory. They are mirrored on grind.isca.uiowa.edu [[128.255.21.233](#)].

Amiga users can get some things by mail from mrcserv@janus.mtroyal.ab.ca (send a **HELP** message first). For Amiga see also the aminet sites:

- litamiga.epfl.ch 128.178.151.32
- ftp.luth.se 130.240.18.2
- ftp.uni-kl.de 131.246.9.95
- ftp.uni-erlangen.de 131.188.1.43
- ftp.cs.tu-berlin.de 130.149.17.7
- ftp.uni-paderborn.de 131.234.2.32
- ftp.wustl.edu 128.252.135.4
- [ftp.cdr om.com](http://ftp.cdr.om.com) 192.153.46.2
- merlin.etsu.edu 192.43.199.20

There is a listserv containing **e-music** related stuff at American University. Send a message to listserv@auvm.bitnet containing the line: [get emusic filelist](#).

IRCAM: IRCAM operates an anonymous FTP site, [ftp.ircam.fr](ftp://ircam.fr), with programs and information on music, DSP, multimedia, infosystems, networking, etc., for Unix platforms, Macintoshes and PC; (amongst others:)

[/pub/](#)

music/	Computer music software and info.
databases/	Music databases.
doc/	Documentation.
FAQ/	FAQ about audio, dsp, etc...
MIDI/	MIDI standard definition.
programs/	Repository of public domain computer music programs.
snd/	

It also has a WWW server on similar resources, <http://www.ircam.fr>. This includes a MIDI archive (programs, sounds, patches, docs).

Computer Music Journal Internet Archive: mitpress.mit.edu:/pub/Computer-Music-Journal and ccrma-ftp.stanford.edu:/pub/Publications/cmj. Tables of contents, abstracts, and editor 's notes of several volumes of CMJ. World-Wide Web (WWW) home page in the file [CMJ.html](#) (e.g. URL <ftp://mitpress.mit.edu/pub/Computer-Music-Journal/CMJ.html>).

[ftp.mcc.ac.uk /pub/cubase](ftp://mcc.ac.uk:/pub/cubase) **Cubase** archive. Contains Atari, Mac and MS Windows demos.

[ftp.waldorf-gmbh.de](ftp://waldorf-gmbh.de) A site with information, patches etc. of **Waldorf products**, contains also a lot of other MIDI stuff. Also reachable by email to archive-server@waldorf-gmbh.de (send a **HELP**

message) . See also mailing list section below.

You'll find some **NoiseTracker/SoundTracker MOD files**, and **Sound Blaster** things on the following FTP sites:

- [Garbo.Uwasa.fi](#) /pc/sb and /pc/music
- [Wuarchive.Wustl.Edu](#)
- [Simtel20.Amy.Mil](#)
- [ftp.funet.fi](#) /pub/amiga/audio
 /pub/atari/sound
 /pub/unix/sound
 /pub/msdos/sound
- [ux1.cso.uiuc.edu](#) /amiga/mirror/mods
- [ftp.brad.ac.uk](#) /misc/mods /incoming/mods
- [ftp.uni-kl.de](#) /pub/amiga/mods
 /pub/amiga/ianet/mods

[/ftp.eng.ufl.edu:/pub/msdos/demos/music/](#) contains several music files, demos etc. (hardly any MIDI, but mostly MODs etc).

The demo version of **QSEQ**, a complete sequencer for MS-DOS/Sound Blaster, is available by email automatic server. Just send a message to qseqmail@h2o.cad.cea.fr with the following line in the body of the message: [sendqseqto <your-complete-email-address>](#) or [sendhelpto <your-complete-email-address>](#) , if you just want an help file about the mailer and QSEQ. QSEQ is also available on SimTel mirrors ([oak.oakland.edu](#) etc...). And from [ftp.cs.ruu.nl:/pub/MIDI/PROGRAMS/MSDOS](#).

[ella.mills.edu](#) in [/ccm/tuning](#) contains files about **alternative tunings**, a.o. back issues of the Alternate Tuning mailing list, software to put synths in alternate tunings and a bibliography on the subject. You can also access this archive by sending a message to listproc@earth.mills.edu. See the file [/ccm/README](#) for more info about this archive.

[ftp.und.ac.za](#) (University of Natal, Durban) contains MIDI files and programs for the **Sound Canvas** and **MT-32** in directory: [/pub/pc/midi/](#).

The official **Sound Canvas Users Group** FTP archive is <ftp://ftp.tue.nl/pub/midi>. WWW access at <http://www.umn.edu/nlhome/g119/evan0102/scgroup/scgroup.html>.

The **EMUSIC-L** mailing list (see below) archives can be found on the WWW <http://sunsite.unc.edu/mcmahon/emusic-l/welcome.html>. FTP access will be available soon.

[ftp.mcc.ac.uk](#) contains the archives of the **Emagic Logic mailing list** in [pub/emagic/ListArchive](#) and other MIDI stuff in [pub/MIDIMISC](#). There is also a WWW page on http://www.mcc.ac.uk/~emagic/emagic_page.html.

The WWW archive of the **Korg Wavestation Mailing List** can be found at: <http://anjovis.tky.hut.fi/~ws/>.

There is a large collection of patches for the **Korg DW/EX8000** at loose.apana.org.au in the directory [pub/dw8000](#).

Korg X3-FTP - [ftp.netcom.com](#) in [/pub/jo/jonin/midi/x3](#) is the **Korg X3** FTP site for patches and various programs relating to and supporting the Korg X3 (editors, utilities, cakewalk ini settings, documents, etc.).

[/musie.phlab.missouri.edu/pub/korg/](http://musie.phlab.missouri.edu/pub/korg/) contains info related to **Korg electronic music products**, particularly the "0" (**01/W, 03, 05**) and **X** synthesizers. Anonymous FTP upload is available in subdirectory subm issions. WWW access is via <http://musie.phlab.missouri.edu/pub/korg/>.

SHARC Timbre Database: [ftp.ep.susx.ac.uk/pub/sandell](ftp://ftp.ep.susx.ac.uk/pub/sandell) (Web users: <ftp://ftp.ep.susx.ac.uk/pub/sandell/README.html>). This is a database with **timbre information** of musical instruments.

heinous.isca.uiowa.edu//kahless/ftp/pub/ **MAX programming language** (for the Macintosh): subdir **max**. Algorithmic composition: subdir **algo-comp**.

WWW Archives of **Classical Midi Sequences:** <http://www.ima.net/~prs/midi.html>.

Csound, a program to generate computer music:

- Macintosh and Unix versions: cecelia.media.mit.edu/pub/Csound/.
- IBM PC and Atari versions: [ftp.maths.bath.ac.uk/pub/dream/](ftp://maths.bath.ac.uk/pub/dream/).

Csound add-ons can be found at ft.hmc.edu/pub/csound and indigo.pac.utexas.edu/pub/csound.

The **Humdrum Toolkit** is a (free) toolkit of programs to assist in music research. Humdrum runs under UNIX, LINUX, NeXTstep, DOS, and OS/2 systems. FTP from archive.uwaterloo.ca/uw-data/humdrum.

MAILING LISTS:

NOTE: The addresses mentioned are those where you can subscribe or get info about the list. The lists themselves usually have the name without the **-request** if that part is present, or **LISTNAME@HOST** if the command to subscribe is **SUBSCRIBE LISTNAME** sent to **LISTSERV@HOST**. If **<Your Full Name>** appears in the following it has to be replaced by your real name (like Piet van Oostrum for me). Similarly **<your-mail-address>** must be replaced by your email address.

Algorithmic composition majordom@heinous.isca.uiowa.edu
message: [subscribe ALGO-COMP](#).

Alternate Tunings listproc@eartha.mills.edu
message: A blank line followed by **SUBSCRIBE TUNING <Your Full Name>** (David Madole, madole@mills.edu).

Analogue Heaven analogue-request@magnus.acs.ohio-state.edu.

Cakewalk LISTSERV@LISTS.COLORADO.EDU
message: **SUBS CAKEWALK <Your Full Name>**

Casio synths casio-request@cads.com

Csound Majordomo@noether.ex.ac.uk
message: [subscribe csound <your-email-address>](#)

Cubase cubase-users-request@mcc.ac.uk

DX7 djb@byrd.math.uga.edu (David J Benson)

DIGITECH RP-1/RP-10 MAILSERV@ECN.NL (Marcel Bernards, bernards@ecn.nl)
Guitar effect users
message: [subscribe RP-1-L](#)

Finale LISTSERV@SHSU.edu
normal: message: [SUBSCRIBE Finale "Your Full Name in Quotes"](#)
digest: message: [SUBSCRIBE Finale-Digest "Your Full Name in Quotes"](#)

EMU Emax emax-request@flobalob.hpl.hp.com

K2000 LISTSERV@american.edu
message: [SUBSCRIBE K2000 <Your Full Name>](#)

Korg 01/W majordomo@mapeaston.com
message: [subscribe korg01w](#)

Korg X3 majordomo@io.rdc.puc-ri.o.br
message: [subscribe x3-list <your-email-address>](#)

MAX Interactive music/
multimedia environment listserv@vm1.mcgill.ca
message: [SUBSCRIBE MAX <Your Full Name>](#)

Netjam netjam-request@xcf.berkeley.edu
with [Subject: request for info](#)

EMagic's Logic majordomo@mcc.ac.uk
(formerly **Notator**)
For its digest: message: [subscribe logic-users <your-email-address>](#)
message: [subscribe logic-users-digest <your-email-address>](#)
human: logic-users-request@mcc.ac.uk
message: [subscribe](#)

Ensoniq VFX vfx-request@digibd.com

EPS eps-request@oak.oakland.edu
message: [subscribe](#)

Quadrasynt MACJORDOMO@POLICC.UNEX.ES
message: [SUB Quadrasynt <Your Full Name>](#)
no subject, no signature.

Roland Samplers sgroup-request@lotus.UWaterloo.ca

Roland D-70 cyamamot@kilroy.jpl.nasa.gov (Clifford Yamamoto)

Roland JV-80/880 jv80-request@burner.com
message: [#SUBSCRIBE <your-email-address>](#)

Roland JV-1080/Xp-50 etogek@etn.ericsson.se

Roland U20/U220 now a Usenet newsgroup
alt.music.synth.roland.u20

TX16W majordomo@lists.eunet.fi
message: [subscribe tx16w <Your Full Name>](#)

Yamaha SY sy-request@chorus.fr

EMUSIC-L LISTSERV@AUVM.AMERICAN.EDU
(LISTSERV@AUVM.BITNET)
(Discussions on Music)
message: [SUBS EMUSIC-L <Your Full Name>](#)
Digest form: [SUBS EMUSIC-D <Your Full Name>](#)

Sound Canvas User's Group Majordomo@damp.apana.org.au
message: [SUBSCRIBE SCUG-LIST](#)

SYNTH-L LISTSERV@AUVM.AMERICAN.EDU
(LISTSERV@AUVM.BITNET)
message: [SUBS SYNTH-L <Your Full Name>](#)

SQ-x/KS-32 ks32-request@cygnus.com

Waldorf (+other) users user-forum-request@waldorf-gmbh.de
Subject: [subscribe](#)

Korg Wavestation [<No longer operational>](#)

Women 's issues in music LISTSERV@IUBVM.UCS.INDIANA.EDU
(LISTSERV@IUBVM.BITNET)
message: [SUBSCRIBE WIML-L <Your Full Name>](#)

Note: I am setting up a MIDI archive on our machine, so if you have info about PD stuff (programs and MIDI files) please share with me. And if you want to share some of your own things that will be appreciated. The archive is available with FTP from [ftp.cs.ruu.nl](ftp://ftp.cs.ruu.nl) [131.211.80.17], directory MIDI. Also by a mail-server - send mail to muu.nlwith with the following contents:

```
BEGIN
PATH <a valid mail address to you>
HELP
send MIDI/INDEX
END
```

Note: specify a correct address (e.g. user@host.univ.edu or user@host.BITNET).

Piet van Oostrum

Dept. of Computer Science
Utrecht University
Padualaan 14
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Telephone: +31 30 2531806
Telefax: +31 30 2513791
Internet: piet@cs.ruu.nl

PGP public key by finger or WWW <http://www.cs.ruu.nl/~piet>

Pronounced as 'Pete'!

How to REGISTER - and WHY!

The WAVmaker Evaluation Package is being distributed as shareware. This allows you to try it out before you decide to buy the WAVmaker Registered Package.

Why bother to register? Here are six good reasons:

- You will [receive the full General MIDI sound set!](#)
- The opening screen and ten second delay will go away!
- You will [receive the latest version!](#)
- You will [obtain the right to distribute and commercially exploit WAVmaker's output!](#)
- You will [further WAVmaker's development!](#)
- You will [feel good!](#)

In order to register, you need to do two things:

- **Decide** whether you want to settle for a [\\$40 US Standard registration](#), enough to get all of the above, or if you want to boost WAVmaker's development by going for the [\\$60 US Advanced registration](#), which entitles you to [having the next major release sent to you at no additional cost when it becomes available](#).

Either way, add \$5 US for shipping and handling if you are in a member state of the European Union, or \$10 US if you are outside of the European Union.

For simplicity, amounts are given in US dollars. You need **not** make your payment in US dollars, though; if you are not in the US, just get hold of a newspaper, look up the exchange rate, and compute the corresponding amount in your own currency. (I could have used ECU, or DM... but most shareware buyers are in the US, so US dollars it is).

- **Pick the payment method most convenient to you:**

MasterCard/Visa	by PHONE	TOLL-FREE in the US!
	by FAX	
	by E-MAIL	
Money order	by SNAIL MAIL	
Bank transfer		
Postal giro		
Check	by SNAIL MAIL	
Cash	by SNAIL MAIL	

The shareware principle

Shareware is not "shared" or "free" software; it's a distribution concept. By allowing you to scrutinize the actual product (rather than a colorful ad and a fancy cardboard box claiming to contain the solution to all your problems) **before** you shell out your hard-earned cash, it forces the author to be at least as good as somebody using traditional sales channels. Who has not bought a commercial product only to be swearing at it or not using it at all just a few days later? As a shareware buyer you risk nothing but the time it will take you to reach an **enlightened** decision as to whether you really want to buy the software.

The author, on the other hand, enjoys substantial savings in distribution and marketing. Apart from the satisfaction of cutting out parasitic middlemen who add nothing but cost to the product, this gives the (good) shareware author a competitive edge in pricing. This is why WAVmaker can be sold at a fraction of the cost of comparable products being distributed along traditional channels: it's not feeding salesmen, only who's doing the actual work! Distribution is being taken care of by the magic of the Internet and the efforts of FTP site and BBS maintainers committed to offering the best possible service.

An increasing number of major players in the software field are realizing the advantages of this approach and turning to shareware for their own distribution (a recent example is Novell). With the ongoing growth and ever increasing proliferation of the Net into ordinary people's homes (witness M\$ Network) and the effort being put into the development of secure cybercash systems, shareware looks more and more like the wave of the future. For limited categories of software products, such as communication programs and archivers, it is already the distribution method of choice, and the recent successes of Doom and Descent have proved that the principle also works well for games (which suffer from particularly short lifecycles, ill suited to the try-before-you-buy approach) if combined with retail-only sequels.

You will receive the full General MIDI sound set!

Remember, what is being freely distributed is not the full WAVmaker, but the WAVmaker Evaluation Package, which only contains a limited set of demo sounds. [Upon registration, you will receive the WAVmaker Registered Package, with the full set of General MIDI compliant samples and patches \(and then some\)](#). You will then in effect have a superb-sounding, 16 bit stereo, General MIDI wavetable sound module at your disposal, allowing you to render virtually any MIDI file which you're likely to encounter on the Net or on a BBS.

You could of course buy a wavetable card or sound module to do the same job, but that would cost you ten or twenty times as much and still not offer all the features of the full WAVmaker package.

For starters, wavetable cards suffer from severe memory limitations. Ask yourself how much wavetable data can be packed into 1 MB or so of ROM. Since they are built for real-time playback, these cards can't use the sophisticated (but computationally heavy) data compression schemes implemented by WAVmaker's WAV to DFT and DFT to WAVconverters, so for them 1 MB is 1 MB, or at best 2-4 MB of actual wavetable data. Compare this to the size of WAVmaker's General MIDI sound library: 150 MB! The magic of DFT to WAV makes it possible to distribute this (truly!) great database on less than three ordinary, 1.44 MB diskettes. If you have not tried these programs out yet, go ahead and do so (I would have a hard time believing this myself without seeing it first hand!).

Also, try finding a wavetable card which will allow you to isolate MIDI tracks and channels, render them separately, run the various parts through different chains of filters and effects like echo, chorus, flanger, vocoder, ring modulator and more, then mix everything down to a finished, CD-quality result, ready to be included in your next game or burnt onto your next CD-ROM! WAVmaker lets you do all that - if you have the necessary sound data, that is :-).

The opening screen and ten second delay will go away!

Since you will no longer need to be reminded about how easy it is to register, you won't be!
[After installing the Registered Package, WAVmaker will start up without delays.](#)

You will receive the latest version!

If you have downloaded or have been handed an obsolete version, you will receive up-to-date files. As long as WAVmaker keeps receiving user support, it will keep developing and getting better, so [see to it that you have the latest and best version!](#)

You will obtain the right to distribute and commercially exploit WAVmaker's output!

Let's face it, there are some really weird people out there, and some may be weird enough to spend a month or so of their time creating their own version of the WAVmaker database rather than shelling out the 40 bucks required for Standard registration. Having done that, they would probably realize that their library can be sold to others. Unfortunately, this would create a threat to WAVmaker's continued development, by competing directly with the Registered Package, so - much as I hate this kind of thing - I have been forced to take some elementary legal precautions.

I **could** of course have left out WAV to DFT from the program; as there is no comparable compression system on the market today, the weirdos would have to either burn a CD-ROM or limit themselves to personal use of their library (which is quite OK with me). But that would punish the vast majority of users whom I sincerely expect to support WAVmaker's continued development, by denying them a valuable compression utility. So, I have included the WAV to DFT converter, but I am explicitly prohibiting two things:

- You may not distribute **any** kind of output produced with WAVmaker unless you have purchased your own personal copy of the WAVmaker Registered Package. What you do with the WAVmaker EP for your personal use is none of my business - if you are a weirdo valuing your time at less than 40 bucks a month, go ahead - but if you want to share the results with others, you must register, as spelled out in the Limited User License Agreement.
- You may not distribute WAVmaker output if that output is part of a competing product, like a sample database meant to be used with WAVmaker itself, and hence in direct competition with the WAVmaker Registered Package, without prior written permission from me. This is true even if you are a registered user (that would make you a double weirdo, or maybe a weirdo squared: first pay 40 bucks to get the real thing, **then** spend a month creating your own database... naah, can't happen...). This is also spelled out in the Limited User License Agreement {License}, which you are accepting to honor by using WAVmaker.

Please note that similar clauses can be found in most software packages capable of creating valuable output (have a look at the license agreement which came with your latest Borland compiler, for instance; it prohibits you from using the compiler to create applications competing directly with the Borland product range!). I don't really like doing this, but I want to ensure that WAVmaker keeps developing and isn't killed right off by parasites.

If you have an interesting sound/patch library which you'd like to make publicly available, [contact me](#) and we'll hammer out an agreement; my objective is protection, not repression (I'm a nice guy, really!), and I **do** want to see WAVmaker grow.

Anyway, the bottom line is: [if you want to distribute or otherwise exploit WAVmaker's output, register A.S.A.P.!](#)

You will further WAVmaker's development!

Already as WAVmaker 2.2 is being released, I'm working on the next round of upgrades. As usual, I won't disclose any details ahead of time, but native Windows 95/NT support opens up many new, exciting possibilities. [Help me explore them!](#)

You will feel good!

Designing, implementing and documenting WAVmaker has been **lots** of hard work. By using WAVmaker without contributing to its continued development, you would not only be acting against your own best interests (wouldn't you like a better, faster version, loaded with lots of new exciting features?); you would also be doing something which you really know to be wrong, deep down inside...

[Do the right thing! Get that good, warm feeling! Register your copy of WAVmaker today!](#)

- **If you are in the US**, call NorthStar Solutions **toll-free** at **1-800-699-6395** (10:00 a.m. - 10:00 p.m. Eastern Standard Time).
- **If you are NOT in the US**, call NorthStar Solutions at **1-803-699-6395** (10:00 a.m. - 10:00 p.m. Eastern Standard Time).

Have the following information ready:

- Program name and version number (**WAVmaker 2.2**).
- **Where** the Registered Package should be mailed.
- Your Visa or MasterCard **number** and **expiration date**.

IMPORTANT

NorthStar processes registrations only, please [contact me](#) directly for any **product/technical support!**

Send the completed registration form to NorthStar Solutions' fax: [1-803-699-5465](tel:1-803-699-5465).
(Available 24 hours. International and business orders encouraged.)

Be sure to include the following information:

- Program name and version number ([WAVmaker 2.2](#)).
- [Where](#) the Registered Package should be mailed.
- Your Visa or MasterCard [number](#) and [expiration date](#).

IMPORTANT

NorthStar processes registrations only, please contact me directly for any **product/technical support!**

Send the completed registration form to NorthStar solutions, using one of the following addresses:

- **America OnLine:** STARMAIL
- **CompuServe:** 71561,2751
- **Internet:** STARMAIL@AOL.COM

Be sure to include the following information:

- Program name and version number ([WAVmaker 2.2](#)).
- [Where](#) the Registered Package should be mailed.
- Your Visa or MasterCard [number](#) and [expiration date](#).

IMPORTANT

NorthStar processes registrations only, please contact me directly for any **product/technical support!**

Send the completed registration form and your money order to

NorthStar Solutions

P.O. Box 25262
Columbia, SC 29224
USA

Make the money order payable to NorthStar Solutions.

Be sure to include the following information:

- Program name and version number ([WAVmaker 2.2](#)).
- [Where](#) the Registered Package should be mailed.
- Your Visa or MasterCard [number](#) and [expiration date](#).

IMPORTANT

NorthStar processes registrations only, please contact me directly for any **product/technical support!**

Instruct your bank to transfer the amount to

SwedBank (SWIFT code: SWEDSESS)
Clearing Number: 8327-9
Account Number: 23.403.510-3
Recipient: **Tommy Anderberg**
Almvagen 4
195 44 Marsta
Sweden

Do not forget to state your full name and address!

If possible, contact me and/or send the completed registration form in connection with your payment.

Instruct your post office to transfer the amount to

PostGiro Bank Sweden (SWIFT code: PGSISESS)
S-105 06 Stockholm

Account Number: 617 37 30-0
Recipient: **Tommy Anderberg**
Almvagen 4
195 44 Marsta
Sweden

This should work in Austria, Belgium, Denmark, Finland, France, Germany, Great Britain, Italy, Japan, Luxembourg, the Netherlands, Norway, Switzerland, Spain, and probably elsewhere too (check with your post office).

If you are in Sweden, use ordinary PostGiro with the above Account Number and Recipient.

Do not forget to state your full name and address!

If possible, contact me and/or send the completed registration form in connection with your payment.

Send the completed registration form and an international check
(you'll have to ask your bank to issue one) directly to

Tommy Anderberg

Almvagen 4
195 44 Marsta
Sweden

Send the completed registration form and cash directly to

Tommy Anderberg

Almvagen 4
195 44 Marsta
Sweden

Needless to say, sending cash is not as safe as going through a bank or the postal giro system; checks are preferable, too.

License information

Last updated: January 1, 1996.

This file contains important license information regarding the distribution and use of the WAVmaker Evaluation Package (henceforth, "WAVmaker EP") and of the WAVmaker Registered Package. If neither Evaluation Package (EP) nor Registered Package are specified, the term "WAVmaker" refers to both.

WAVmaker is NOT in the public domain, nor is it freeware. It is [Copyright \(C\) Tommy Anderberg, 1995-1996](#). All rights reserved worldwide. This software and all the accompanying data and documentation files are protected by national and European laws and by International Treaty provisions. You must treat WAVmaker like any other copyrighted material, except that you may copy the WAVmaker EP and give to others (subject to the limitations outlined below). Any use or distribution of this software, data and documentation in violation of Copyright law or of the terms outlined in this document will be prosecuted to the best of our ability.

The conditions under which you may copy and distribute the WAVmaker EP are described below under [Limited Distribution License Agreement](#).

The conditions under which you may use WAVmaker are described below under [Limited User License Agreement](#).

All rights not expressly granted here are reserved to Tommy Anderberg.

Limited Distribution License Agreement

This Limited Distribution License Agreement (throughout this section, the "Agreement") is a legal agreement between you, the distributor, and Tommy Anderberg. By copying any part of WAVmaker, you are agreeing to be bound by the terms of this Agreement.

As the copyright and trademark holder for WAVmaker, Tommy Anderberg authorizes individuals and organizations to copy the files contained in the WAVmaker EP for their own use or for giving to others, ONLY when the following conditions are met:

- The WAVmaker EP - including all related program, data and documentation files - may not be modified in any way and must be distributed together, as a complete package, without exception. You may not distribute corrupted or otherwise damaged files.
- No price or other compensation may be charged for the WAVmaker EP. A reasonable distribution cost may be charged for the distribution media, shipping and handling. BBS systems may not charge beyond their normal rates for downloads of the WAVmaker EP.
- WAVmaker is designed to be easily extended by the addition of new program and data modules. You may not, however, distribute an extended version of the WAVmaker EP without prior written permission from Tommy Anderberg.
- You may not distribute any file contained in the WAVmaker Registered Package, either with the WAVmaker EP, by itself, or as part of another public domain, freeware, shareware, or commercial packaging offer, without prior written permission from Tommy Anderberg.

- Neither this Agreement nor any part or portion hereof shall be assigned or sublicensed, except as described herein. Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby. If any provision is determined to be unenforceable, you agree to a modification of such provision to provide for enforcement of the provision's intent, to the extent permitted by applicable law. Failure of a party to enforce any provision of this Agreement shall not constitute or be construed as a waiver of such provision or of the right to enforce such provision. If you fail to comply with any terms of this Agreement, your license is automatically terminated, and you may be prosecuted at any time under national and European laws and under International Treaty provisions.
- You acknowledge that you have read this Agreement, you understand this agreement, and understand that by copying any part of WAVmaker, you agree to be bound by this Agreement's terms and conditions. You further agree that, except for written separate agreements between Tommy Anderberg and you, this Agreement is a complete and exclusive statement of your distribution rights. This agreement supersedes all prior oral agreements, proposals or understandings, and any other communications between Tommy Anderberg and you relating to the subject matter of this Agreement.

Limited User License Agreement

This Limited User License Agreement (throughout this section, the "Agreement") is a legal agreement between you, the user, and Tommy Anderberg. By installing WAVmaker, by loading or running any part of it, or by placing or copying any part of it onto your computer hard drive, you are agreeing to be bound by the terms of this Agreement.

For the purposes of this section, "use" means loading the software into RAM, as well as installation on a hard disk or other storage device.

As the copyright and trademark holder for WAVmaker, Tommy Anderberg authorizes individuals or organizations to use WAVmaker, ONLY when the following conditions are met:

- You shall not: modify, translate, disassemble, decompile or reverse engineer the compiled executables contained in WAVmaker.
- You may not exploit running copies of WAVmaker commercially, e.g. by renting, leasing or selling access to a computer running WAVmaker.
- The WAVmaker EP is intended for evaluation purposes. You may not distribute or otherwise exploit the output produced by WAVmaker if you have not purchased your own personal copy of the WAVmaker Registered Package.
- You are free to distribute and exploit the output produced by WAVmaker, provided that you have purchased a personal copy of the WAVmaker Registered Package, and that the output in question is not part of a competing product (e.g. a sound data base for use with WAVmaker itself, and hence in direct competition with the WAVmaker Registered Package), in which case you must have prior written permission from Tommy Anderberg.
- The user accepts sole and full responsibility for any damage or loss arising from the use or the inability to use WAVmaker. While this software has been extensively tested and is believed to be safe, Tommy Anderberg disclaims any warranties, either expressed or implied, including but not limited to warranties of fitness for ANY particular purpose and of uninterrupted, error free operation. Under no circumstances shall Tommy Anderberg

be liable for loss of data, loss of profits, lost savings, special, incidental, consequential, indirect or other similar damages arising from the use or the inability to use WAVmaker.

- Neither this Agreement nor any part or portion hereof shall be assigned or sublicensed, except as described herein. Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby. If any provision is determined to be unenforceable, you agree to a modification of such provision to provide for enforcement of the provision's intent, to the extent permitted by applicable law. Failure of a party to enforce any provision of this Agreement shall not constitute or be construed as a waiver of such provision or of the right to enforce such provision. If you fail to comply with any terms of this Agreement, your license is automatically terminated, and you may be prosecuted at any time under national and European laws and under International Treaty provisions.
- You acknowledge that you have read this Agreement, you understand this Agreement, and understand that by installing any part of WAVmaker, by loading or running any part of WAVmaker, or by placing or copying any part of WAVmaker onto your computer hard drive, you agree to be bound by this Agreement's terms and conditions. You further agree that, except for written separate agreements between Tommy Anderberg and you, this Agreement is a complete and exclusive statement of the rights and liabilities of the parties involved in the use of WAVmaker. This agreement supersedes all prior oral agreements, proposals or understandings, and any other communications between Tommy Anderberg and you relating to the subject matter of this Agreement.

