**Contents**

## Introduction

### What Wave Repair Is

Wave Repair is a special purpose WAV file editor specifically aimed at the restoration of recordings sourced from vinyl LPs. It provides a number of facilities to help in removing the clicks and pops inherent in such recordings. A number of its features may also be generally useful in other situations.

It is a 32 bit program that runs under Microsoft Windows 95/98 and NT. It has not been tested on Windows Me or Windows 2000, but there is no reason why it should not work on those operating systems. It operates only on 16bit stereo WAV format files sampled at 44.1kHz or 48kHz.

I developed Wave Repair for my own use while restoring music signals retrieved from vinyl records. It is specifically targeted at identifying and repairing the clicks, pops and ticks that inevitably plague vinyl replay. I wrote it because I was unable to find an affordable WAV file editor which performed this task in an acceptable manner.

Other editors which claim to perform declicking do so by algorithmic means, usually by performing some kind of spectral analysis. One major barrier to using these tools productively is that they tend to have a number of configuration parameters that can be tried, but the actual declicking is a slow process. The consequence of this is that the typical procedure is: set parameters; declick; listen to results; repeat ad nausiam. Wave Repair includes a real-time declicking function, which allows the user to interactively adjust the click detection parameters while listening to the effects. In this way, the best settings for a particular section of waveform can be found quickly. Despite this facility, my own experience is that any kind of automatic approach often fails to produce acceptable results. This is probably due to the unpredictable nature of vinyl damage.

There are of course professional hardware and hardware/software systems that perform this task (such as CEDAR), but they are very expensive. I am unable to verify their performance since I do not have the funds to try them for myself, but assume that since the professional recording industry is enthusiastic about them, then they probably do a very good job. Therefore, if you are a professional with the funds to use these kinds of tools, Wave Repair is probably not worth your consideration. (In any case, if you're a professional, you're probably working at rather higher resolutions than 16bit 44.1/48kHz).

Despite the fact that Wave Repair does have a real-time declicking function, for good results it remains primarily a manually operated system. I have come to the conclusion that the only reliable way to do high quality declicking on a budget is to manually repair the damage on an individual click basis. The repair method for any specific click might be: interpolation, replacement with a similar block from nearby, copying over from the other channel, smoothing, muting, complete removal, or perhaps redrawing the waveform using the mouse as a "pen". Wave Repair is specially designed to make these types of manual repair operations straightforward to carry out. General purpose audio editors tend to be less usable in this respect.

A typical method of cleaning up an LP is: the user scans through the waveforms listening and looking for damage, and fixes any found. (The main use of the automatic declicking is for cleaning up very quiet sections (eg. fade-outs) which can contain thousands of tiny ticks and would be unbearably tedious to fix on an individual tick-by-tick basis). Consequently, using Wave Repair is a time-consuming business: listen to the music; home in on a click by scanning through the waveform; repair the damage; repeat for the next click. Performing a thorough clean up of an entire LP typically requires many hours of work.

Wave Repair does not read in the whole file or convert it to some internal format. It only reads as many samples as are required to paint the display, so the maximum number of samples read from the WAV file is the width in pixels of your display. The only significant delays are due to positioning within the WAV file to read an appropriate subset of samples. For example, initial reading of a 200Mb WAV file on a system with a typical EIDE hard disk takes less than 10 seconds.

When you make alterations, Wave Repair stores them in main memory as differences. When you save your changes, it simply rewrites those parts of the WAV file that are affected by the differences. The consequences of this are:

1. Saving of changes is very rapid. (There is one situation where the saving of changes is not so rapid: where there is an extensive change (such as a Smooth or Fade) on only one channel. This is because the data in a WAV file has the channels interleaved, which means that if every sample in a range must be updated on one channel only, then it is necessary for Wave Repair to separately position and write each individual updated sample within the file).
2. There is no long delay up front when working with long WAV files.
3. You don't need any extra workspace on disk for temporary files (this is less of an issue these days, now that hard disks are so cheap).
4. On the downside, you can't make changes which alter the number of samples in the WAV file, since this would require rewriting the entire file. Other shareware editors are better suited to this task. Wave Repair does have the ability to mark a section for removal (or mark that extra silence be inserted), and then to write a new WAV file with the marked sections removed/inserted, but this is not a generalised cut-and-paste facility.

<u>What Wave Repair Isn't</u>

Wave Repair is not a general purpose WAV file editor. I am generally delighted with the quality of other shareware editors, and saw no reason to re-implement anything which was available in them that is not specifically useful for restoring recordings sourced from vinyl LPs.

Typical operations that are available in general purpose digital audio editors that you won't find in Wave Repair are:

1. Have many wave files open at once.
2. Cutting and pasting sections of waveform.
3. Effects such as distortion, echo, reversal, pitch shift, playback rate change.
4. Resampling to different sample rates and/or resolutions.
5. Mixing of multiple waveforms into one.

6. Removal of broadband noise.

I am aware of the fact that the last of these operations (broadband noise reduction) is sometimes useful when restoring vinyl LP recordings, and may investigate the possibility of adding it in a future release. For now, though, if you need this feature, other tools will have to be used. To this end, two inexpensive general purpose shareware editors that I can highly recommend are GoldWave and CoolEdit.

**System Requirements**

Wave Repair is not especially demanding of resources, and will run successfully on quite low-end machines. The minimum requirements are:

1. PC running Windows 95, 98, NT or 2000. It should work on Windows Me, but this has not been verified.
2. Soundcard capable of 44.1/48kHz 16bit stereo playback/recording.
3. Reasonably fast hard disk (eg. EIDE or SCSI) in order perform playback/recording without hiccups.
4. 486 or higher CPU. However, note that the real-time preview functions are CPU hungry, and the minimum CPU requirement to operate these functions reliably is a 200MHz Pentium.
5. Wave Repair works with an ISA video card, but a local bus card (eg. AGP, PCI, etc) is preferred.

**Registration**

If you have registered, here are instructions to install your unlock key.

The registration fee for Wave Repair is US$30. When you register, you will be sent a personal unlock key which will work with all future releases of Wave Repair. (In other words, you get free upgrades for life).

The simplest and most convenient way to register is with a credit card transaction on the Internet at this web site:

**http://www.shareit.com/programs/100727.htm**

This method is better for you because you will receive your unlock key by return email, usually within minutes. It is better for me because the transaction is completely automatic and I do not have to do any manual processing. ShareIt is a third party credit card merchant. Their web site is secure (transactions are encrypted), so this payment method is just as safe as mail-ordering something on your credit card.

Alternative Payment Options

If for some reason you would prefer not to register via ShareIt with a credit card, or do not have a credit card, then there are alternative options. However, please bear in mind that all these other payment methods require me to send your unlock key by hand, which means that if I am away there could be a delay:

PayPal
Cash
Personal Cheque (from a UK bank or building society)
Electronic Bank Transfer

**Interface**

**Menus**

Please note that menu items are only enabled when appropriate. Many operations operate on the selected region, so if there is no currently selected region, those menu items are disabled. A few operations directly update the WAV file on disk; they can be recognised by the suffix "(Direct Overwrite)" in the menu item. These operations can only be performed when there are no unsaved edits, so if there are any outstanding edits those menu items are disabled.

**File Menu**

Open WAV File
Opens a WAV file. The currently open file (if any) is closed.

Save
Saves changes that have been made directly back over the original file.

Save As
Saves the entire file to a different place, including any outstanding changes. Note that this can take a long time, since the entire file must be written.

Save Selected Samples
Saves the selected region of the file as a new WAV file. This is useful for copying out small regions of the file which you would like to work on with another WAV editor.

Close WAV File
Closes the currently open WAV file. This is actually never really necessary, since Wave Repair does not hold the file open, but is included for completeness.

Truncate Extra Data
Some other editors add extra administrative data to the end of the WAV file. Although this does not bother Wave Repair at all, it might cause problems with other programs, such as CD-R burning software. In these cases, you may want to remove this extra data via this option.

Options
Invokes the Options dialog, through which the user can configure a number of items pertaining to the interface and operation of Wave Repair.

Macros
Invokes the User Macros Setup dialog, through which the user maintains macro definitions.

Exit
Shuts down Wave Repair.

Recently Opened Files
A list of up to nine recently opened files appears at the end of the menu. Any of these files can be opened by selecting from the menu, thus avoiding the need to use an open file dialog.

## Edit Menu

Undo
Removes the most recent change. The preceding change becomes the most recent, which can then be removed. You can keep undoing changes until there are none remaining.

Redo
Re-applies the most recently undone change. You can keep redoing undone changes until there are none remaining. Note however that as soon you make any update, or save the file, the ability to redo any previously undone changes is lost. (In other words, if you undo a change then make another change or save the file, you can't redo the previous undo).

Interpolate Left
Interpolates the selected region of the left channel using a straight line.

Interpolate Right
Interpolates the selected region of the right channel using a straight line.

Interpolate Both
Interpolates the selected region of both channels using straight lines.

Bezier Interpolate Left
Interpolates the selected region of the left channel using a Bezier curve; when the number of samples being interpolated is small (less than a few dozen), such a curve usually gives a better fit with the surrounding waveform than a straight line does.

Bezier Interpolate Right
Interpolates the selected region of the left channel using a Bezier curve.

Bezier Interpolate Both
Interpolates the selected region of both channels using Bezier curves.

Silence
Replaces the selected samples with silence (digital zeros).

Fade In
Progressively fades in the selected region from total silence to full volume.

Fade Out
Progressively fades out the selected region from full volume to total silence.

Partial Fade In
Progressively fades in the selected region from a defined attenuation level to full volume.

Partial Fade Out
Progressively fades out the selected region from full volume to a defined attenuation level.

Mark Selection for Deletion
Marks the selected region for later removal.

Clear Deletion Mark(s)
Removes any deletion marks within the selected region.

Add Silence Insertion Mark
Places a marker at the start of selection indicating that a period of silence is to be added to the file at a later stage. The amount of silence to be added can be specified as a time (mins:secs.millisecs) or as an absolute number of samples.

Add Surrounding Silence
Places silence insertion markers at the start and end of the selected region. This is just a quicker way of achieving what could be done with two separate calls to Add Silence Insertion Mark (the second of which, for the end of selection, would require careful redefinition of the start of selection). It is primarily intended to be used when the whole WAV file is selected, thus allowing standard amounts of silence to be added at the beginning and end of a file. See the Add Surrounding Silence dialog.

Remove Silence Insertion Mark(s)
Removes any insert-silence markers within the selected region.

Execute Deletions/Insertions
Rewrites the WAV file, with all regions marked for deletion removed, and all marked silence insertions added. Note that the WAV file is not directly overwritten (since a power failure during this process would corrupt the file). Instead, a temporary file is written, and if this is successful, the original WAV file is then replaced. The temporary file is written to the specified temporary directory (see Options dialog). If this temporary directory is on a different hard disk partition to the original WAV file, the file will need to be written twice (once to the temporary file, and once to move it back over the WAV file). If the temporary directory is on the same disk as the WAV file, the second phase is a simple rename, so the operation takes half the time.

Append Other File at End
Adds the contents of another WAV file at the end of the currently loaded file. This provides a convenient way of concatenating WAV files together.

Smooth Abrasion
Smoothes out the selected samples with a cleaner waveform. Primarily intended for the reduction of short bursts of distortion caused by scuffs or wide scratches. By default, a fairly subtle smoothing is applied, but a setting in the Options dialog can be used to apply more drastic smoothing.

Edit Left
Determines whether changes to the left channel are possible. Any changes you attempt to make to the left channel (eg. interpolations, redrawing of the waveform) when this option is switched

off will be ignored. If a channel is currently not editable, its colour is changed on the main window's display to make the fact visually obvious.

Edit Right
Determines whether changes to the right channel are possible.

Redraw Wave Mode
Switches between normal mode and redraw wave mode.

Replace From File (With Undo)
Overwrites samples, starting at the beginning of the current selection, with samples from another WAV file. See Select Replacement Data dialog for details of how to specify the replacement data. This particular menu item performs the replacement as an update stored in memory or a temporary file so that its effect can be removed with the Undo option. Note that this could consume a vast amount of memory. If your computer has the resources to do this, then it will work but may run extremely slowly.

Replace From File (Direct Overwrite)
As Replace From File (With Undo), but directly overwrites the original file on hard disk, and therefore cannot be undone. This will run considerably faster than Replace From File (With Undo), but should only be used if you are absolutely sure it is what you want.

## Blocks Menu

Copy Preceding Block (Left)
Replaces the selected samples on the left channel with the samples immediately preceding them. If a segment of damage cannot be repaired using interpolation or wave-redrawing, it may sometimes be possible to substitute a copy of what came before. (It will probably then be necessary to redraw the lead-in and lead-out to the copied section to get a seamless join).

Copy Preceding Block (Right)
The same as **Copy Preceding Block (Left)**, except that the right channel is updated.

Spectral Replacement (Left)
An average of the immediately preceding and following blocks is constructed via Fourier analysis, and the resulting waveform is replaced over the selected region. This menu item operates on the left channel. For fairly short regions of between a few hundred and a few thousand samples, this can often give a better repair than simply copying a block over from elsewhere. Spectral replacement is limited to blocks with a maximum length of 16,384 samples.

Spectral Replacement (Right)
The same as **Spectral Replacement (Left)**, except that the right channel is updated.

Copy Left to Right
Replaces the selected samples on the right channel with the corresponding samples of the left channel. Sometimes (especially where the stereo differences are small) a good repair of one channel can be effected by substituting the samples from the other channel. (It will probably then be necessary to redraw the lead-in and lead-out to the copied section to get a seamless join).

Copy Right to Left
The same as **Copy Left to Right**, except that the left channel is updated.

Nudge Up (Left)
Moves the selected samples of the left channel upwards on the display. The most common use for this would be after copying an overlay block that fits well but is offset vertically. It can also be useful to correct a sudden jump in the waveform. The amount by which the samples are nudged is just that which causes them to move by one pixel on the display. Therefore, the actual offset applied depends on the current vertical scaling factor; you can control this by adjusting the scaling factor using the scroll bars along the right hand edge.

Nudge Down (Left)
Like **Nudge Up (Left)**, but moves the selected samples of the left channel downwards on the display.

Nudge Up (Right)
The same as Nudge Up (Left), but operates on the right channel.

Nudge Down (Right)

The same as Nudge Down (Left), but operates on the right channel.

Set Compatible Block Params
Invokes the Compatible Block Parameters dialog to configure the detailed behaviour of the Find Compatible Block option.

Reset Compatible Block
Discards the possible replacement block that was found (if any).

Find Compatible Block
Searches for a possible replacement portion of waveform for the selected region. If there is no current potential replacement, the search starts right next to the selected region, otherwise it starts from the location of the current block.

Copy Compatible Block (Left)
Replaces the selected region of the left channel with the current possible replacement.

Copy Compatible Block (Right)
Replaces the selected region of the right channel with the current possible replacement.

Set Overlay Block
Establishes the currently selected region as a candidate to be overwritten by another block of the same length in the WAV file. This causes an image of the candidate region to be displayed in blue on top of wherever it has been placed in the WAV file.

Discard Overlay Block
Removes a candidate block that was previously set up by Setup Overlay Block.

Back by Block Width
Moves the blue image of the candidate block backwards in the WAV file by its own length.

Forward by Block Width
Moves the blue image forwards by its own length.

Back One Sample
Moves the blue image backwards by one sample.

Forward One Sample
Moves the blue image forwards by one sample.

Copy Overlay (Left)
Copies the samples underneath the blue image over the candidate block, for the left channel only.

Copy Overlay (Right)
Copies the samples underneath the blue image over the candidate block, for the right channel only.

Reselect Original Block
Re-selects the region which was the selected region when the overlay block was established. This is useful if, having used region selection as a means of conveniently adjusting zoom factors, you wish to easily return to the original block of interest.

Smooth Edges After Copying
If switched on, the transition points at the start and end of copied blocks are smoothed into the surrounding samples with Bezier interpolations.

## View Menu

Standard Detail Scale
Sets the time-axis zoom factor to be the number of samples set in the Options dialog. The default scale is one pixel per sample.

Set Detail Scale
This is a quick way to set the number of samples to be used in Standard Detail Scale to be the number currently displayed. Thus, you can zoom to a desired level and then invoke this option rather than figure out how many samples to specify in the Options dialog.

Selected Samples
Displays the selected region such that it occupies the whole screen.

Select Displayed Samples
Makes the currently displayed page of samples the selected region.

Select Entire File
Makes the entire WAV file the selected region (but does not display it).

Zoom In
Sets the time-axis zoom factor to be twice the previous one (ie. displays half as many samples on the screen).

Zoom Out
Sets the time-axis zoom factor to be half the previous one (ie. displays twice as many samples on the screen).

Entire File
Displays the entire WAV file such that it occupies the whole screen.

Maximise Amplitude
Determines whether each screenful of samples is displayed such that the amplitude-axis zoom factor is made as great as possible.

Ignore Left in Maximise
If checked, only the right channel is considered when determining the maximisation factor. Note that if Ignore Right in Maximise is already checked, an attempt to check this will be ignored.

Ignore Right in Maximise
If checked, only the left channel is considered when determining the maximisation factor. Note that if Ignore Left in Maximise is already checked, an attempt to check this will be ignored.

Reset Vertical Adjusters
Returns the scroll bars at the right hand edge of the screen to display the waveform at 100%, centered.

Previous View
Displays the previous page in the list of displayed pages.

Next View
Displays the next page in the list of displayed pages.

Recent Views
A list of up to nine recently displayed pages appears at the end of the menu. You can return to any of these pages by selecting from the menu.

**Position Menu**

Beginning of File
Displays the first page of samples in the file.

End of File
Displays the last page of samples in the file.

Next Page
Moves the display to the next page of samples.

Previous Page
Moves the display to the previous page of samples.

Next Fragment
Moves the display forward by that fraction of a page as defined in the Options dialog. The default is a sixth of a page.

Previous Fragment
Moves the display backwards by that fraction of a page as defined by the Options dialog.

Start of Selection
Moves the display such that the start of the selected region is at the start of the screen.

End of Selection
Moves the display such that the end of the selected region is at the end of the screen.

Next Revolution
Moves the display onwards by an amount corresponding to one revolution of the record (as defined in the Options dialog); the intention here is to make it easy to move on to the next click caused by a scratch across the record.

Previous Revolution
Moves the display backwards by an amount corresponding to one revolution of the record.

Goto Position
Moves to a specific elapsed time within the file, and moves the page such that the chosen time is one fragment (default one sixth of a page) from the left of the screen; optionally allows the start and/or end of the selected region to be set simultaneously. See Goto Time dialog.

Mark Region
Selects a region of the file by elapsed time or absolute sample number, and displays that region such as to occupy the whole screen. See Mark Region dialog.

Remember Selection
Records the current selected region so that it can be returned to later. This is particularly useful if

you have selected a particular section of music to be worked on and wish to be able easily to return to it after performing other operations that require changes to the selected region.

Reinstate Selection
Returns the selected region to be that most recently recorded with Remember Selection.

Selection Details
Displays the exact sample numbers of the current selection. These may be useful when used in conjunction with other WAV editors or during a subsequent use of Replace From File.

Selection Statistics
Displays the peak and average (RMS) values of samples within the current selection. This can be helpful when used in conjunction with amplification and compression to achieve similar perceived loudness across many WAV files. (Files with similar RMS values tend to have similar perceived loudness).

**Declicking Menu**

Set Declick Parameters
Invokes the Click Detection/Repair Parameters dialog to configure the automatic click detection facility.

Analyse Click (Left)
Attempts to set up the click detection parameters by analysing the waveform on the left channel at the start of selection. The analysis finds the least aggressive parameters which detect the click at that place, while not detecting other clicks immediately to either side. (Note it is possible that there may be no detection settings which will work). The click detection parameters are left set to the results of the analysis.

Analyse Click (Right)
As for Analyse Click (Left), but the waveform on the right channel is analysed instead of the left.

Find All Clicks
Scans the selected region for possible clicks.

Clear Stored Clicks
Discards the list of possible clicks found by the last Find All Clicks.

Clear False Clicks
Discards the list of positions that have been explicitly declared not to be clicks. Thus, next time a Find All Clicks is invoked, these positions may be included.

Goto Next Click
Moves the display on to the next click in the list found by the last Find All Clicks. If the display has already reached the end of the list, this option wraps round to the first click.

Goto Previous Click
Moves the display back to the previous click in the list found by the last Find All Clicks. If the display has already reached the beginning of the list, this option wraps round to the last click.

Goto Specific Click
Moves the display to a particular click in the list by specifying its number.

Goto Click After Start of Selection
Moves the display to the first click in the list which is after the current start of selection. This can be useful if a large batch of potential clicks were found close together (typically during a quiet section) and you can hear there is no audible click within that section; place the start of selection at the end of the section and invoke this option to skip over this large batch of false clicks.

Remove Current Click
Interpolates around the currently selected click in an attempt to remove it.

Set Current Click as False
Adds the position of the currently selected click to a list of declared non-clicks. The click is also removed from the list of found clicks. Once a click has been set as false, subsequent invocations of Find All Clicks will not include clicks at the same position.

Remove All Clicks
Performs Remove Current Click on all clicks in the list.

Find and Remove All Clicks
Scans the selected region for clicks and removes them all in one pass. This is a rapid operation that updates the original WAV file directly, and it cannot be undone.

Save Clicks in File
If a long list of potential clicks were found by Find All Clicks, you may not have time to deal with them all in the current session. This option stores the list of potential click positions in a file, so that you can end your session and return to them later. It also stores the list of declared non-clicks.

Load Clicks from File
Retrieves a list of potential clicks and declared non-clicks that were saved earlier. Note that when this option is executed, the current list of potential clicks (if there is one) is lost.

Declick Preview
Repeatedly plays the selected samples via the selected soundcard whilst removing clicks according to factors set in the Declick Preview dialog.

**Cue Points Menu**

Add Cue Point
Adds a track or index marker at the first CD frame boundary after the current start of selection position.

Delete Cue Point(s)
Deletes all track and index markers within the currently selected region.

Goto Cue Point
Invokes the Goto Cue Point dialog in order to move to a specific cue point.

Find Tracks
Scans the selected region looking for possible track boundaries. The characteristics of a track boundary, and how they are marked, can be specified in the Options dialog.

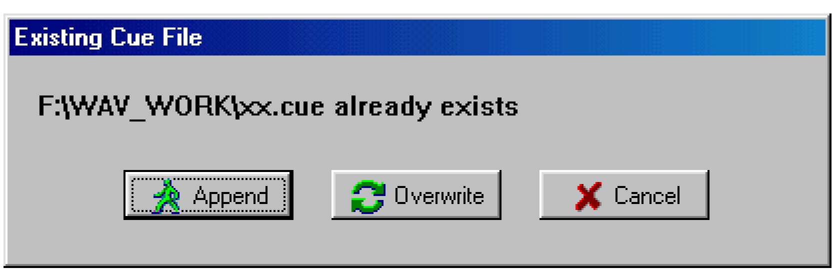
Convert Markers to Cue Points
Converts every marker in the selected region into a cue point defining the start of a new track. This can be useful if you have placed markers manually at the start of each track, or if you have run Find Tracks, asking it to place markers.

Display Track
Presents the Display Track dialog from which a specific track may be picked. The display is then updated so as to show the entire track selected.

Split Tracks
Writes individual tracks (as defined by the cue points that are set) to separate new WAV files.

Setup CD Text Data
Allows you to prepare limited CD Text information that, if a cue sheet is written, will be added to that cue sheet. See CD Text Setup dialog for further details.

Write Cue Sheet
Creates a CDRWin-style cue sheet file from the markers that have been set. If an existing cue file is selected, the following dialog appears:




The **Append** button causes the cue points to be added to the end of the existing cue sheet file, starting with the next track number after the last one in the file; this provides a convenient way to build a single cue sheet for many WAV files. The **Overwrite** button simply replaces the existing cue sheet file with one for just the current WAV file. The **Cancel** button abandons the operation.

Read Cue Sheet
Reads a cue sheet that was previously created by Write Cue Sheet and recreates the cue points.

**Help Menu**

Contents
Invokes online help for Wave Repair.

Registration
Describes registration options for Wave Repair.

About
The usual box about the program.

**Dialogs**

**User Configurability**

A number of interface attributes may be configured by the user to suit their preferences. These are:

Options when opening WAV files.
Options when playing and recording WAV files.
Detailed behaviour of some operations.
Configuration of the soundcard.
Various matters concerning the main window.
The colours used in the display.
The shortcuts assigned to the menu items.

All the above items are configured via the Options dialog.

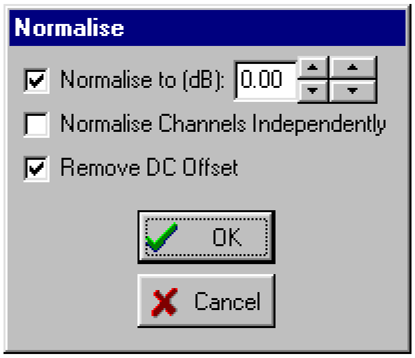
Additionally, Wave Repair includes a facility whereby you can set up macros that execute a sequence of operations. Up to 24 such macros can be created; they are set up via the User Macros Setup dialog.

**The Restoration Process**

The process of restoring recordings made from vinyl LPs involves the following steps:

## Normalise Dialog

This dialog appears when **Other Effects | Normalise** is invoked:




If **Normalise to** is checked, then normalisation will be performed, and in this case the maximum level that will be achieved is given by the dB level set in the box alongside. The level can be reduced to -3dB if desired, but as a general rule it should be left at 0dB.

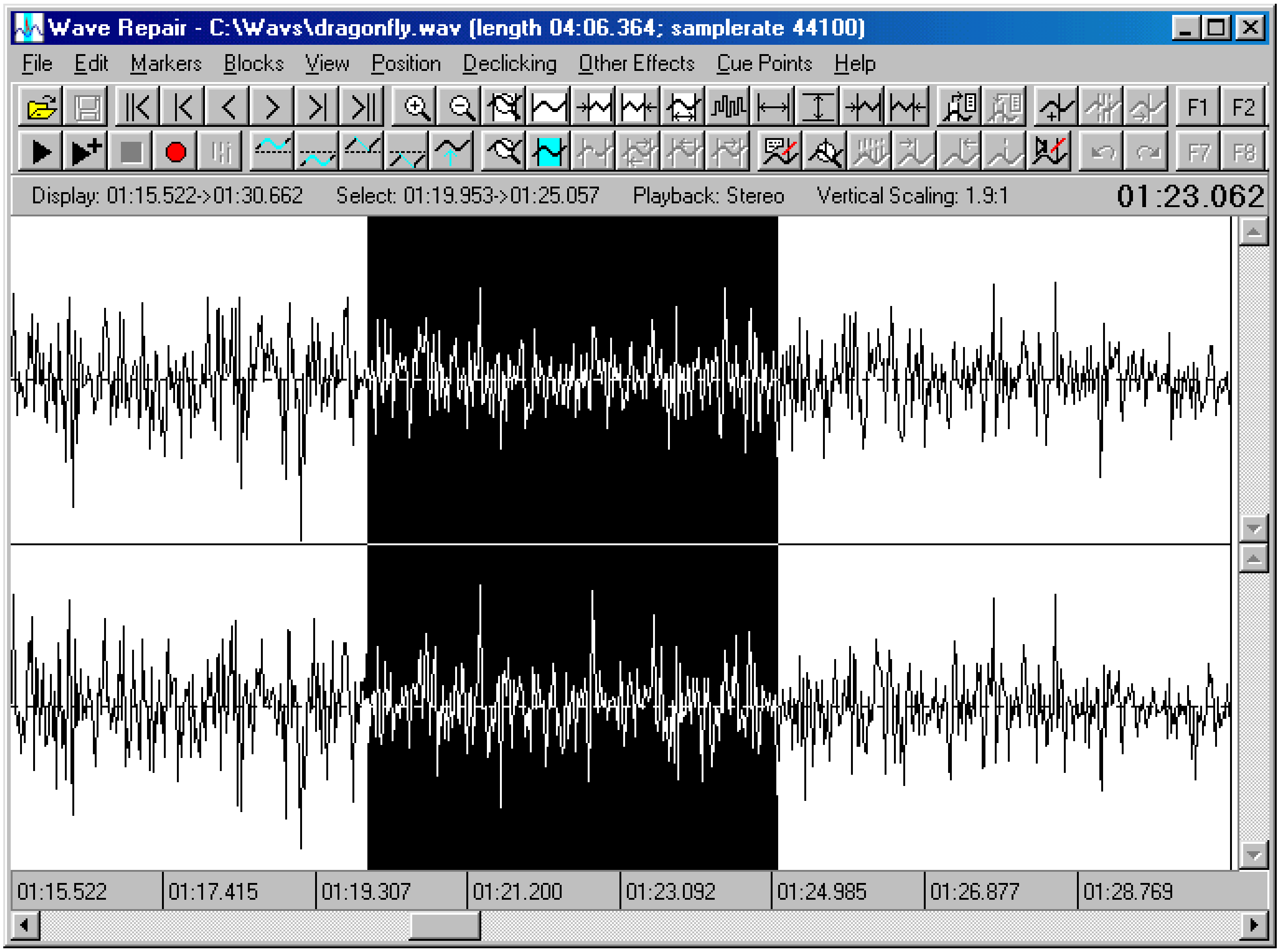
If **Normalise Channels Independently** is checked, then both channels will be normalised to the same maximum level. This can alter the balance of the two channels if they were significantly different in level previously. By leaving this option unchecked, the relative loudness of the two channels is retained (and the louder of the two will be maximised to the specified level).

If **Remove DC Offset** is checked, then any DC offset that exists will be corrected during normalisation. This means that if the waveform was off-centre, it will be centred before normalisation.

To remove a DC offset, without performing any normalisation, simply check the **Remove DC Offset** option, and leave **Normalise to** unchecked.

## The Main Window

Like most other sound editors, working in Wave Repair is largely carried out in its main window:




The caption gives the name of the WAV file being processed, plus its overall length and sample rate. Beneath the menubar is a toolbar containing buttons to execute common functions; each button has a tooltip describing its use which pops up if the mouse is held over the button. The toolbar can be switched off if the user prefers not to use it. Underneath the toolbar is a status bar showing various attributes of the WAV file:

**Display** shows the start and end times of the currently displayed portion of the file.

**Select** shows the start and end times of the portion of the file which is currently selected (if any).

**Playback** indicates which channel(s) will be played during playback.

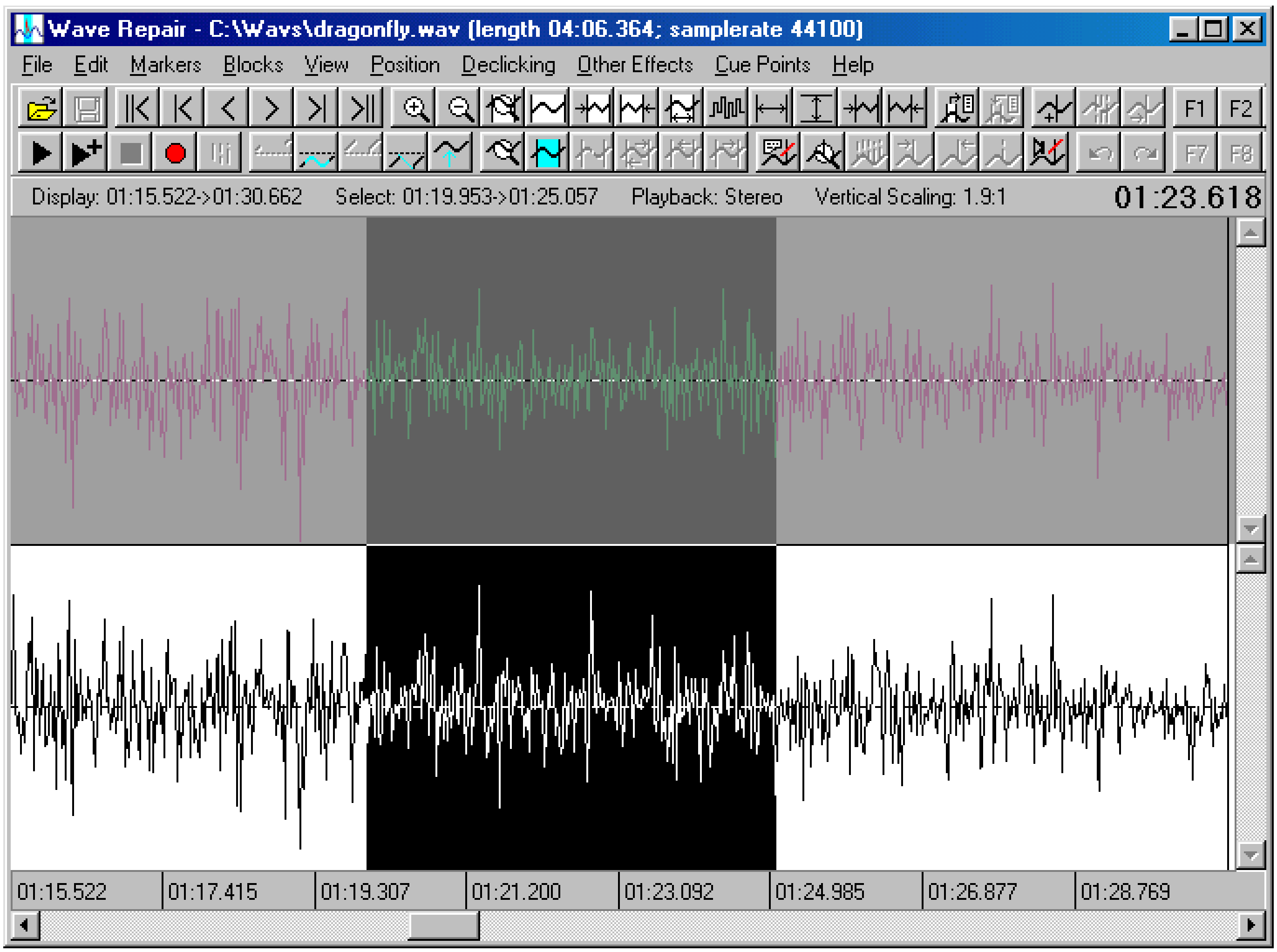
**Vertical Scaling** indicates the current amplification on the vertical scale.

The time at the far right hand end shows the current position of the mouse, (or the position into the file during playback).

By default Wave Repair shows times in the form **min:sec.millisecs**, but the user can change this to be **min:sec.frames** (where a frame is 1/75th sec; the standard block size used on CDs).

The two waveforms are the left and right channels (left on the top, right on the bottom). The waveform is normally displayed in black-on-white, but the part of the WAV file which is selected is shown reversed (ie. white-on-black). This selected region is important, since most of the functions within Wave Repair operate on the selected region. To the right of the waveforms are a pair of vertical scroll bars which control the zoom factor of the vertical axis (ie. the amplitude domain). The top scroll bar adjusts the zoom factor, while the bottom scroll bar adjusts the centre-line offset. Since in most cases it is convenient to work with Maximise Amplitude enabled (a mode whereby the displayed page is automatically scaled so that the amplitude shown is as large as possible), these scroll bars can be switched off if desired.

It is possible to switch off the ability to edit one or other channel (so that an effect can be applied to one channel only). In this case, the channel is shown in a different colour to make it visually obvious that it is currently not editable. Here is the appearance when the left channel is not editable:




Beneath the waveforms is a time bar which shows elapsed time into the WAV file for the displayed screen.

Finally, there is a horizontal scroll bar. The entire scroll bar represents the complete WAV file, the size of the scroll bar's thumb indicates what proportion of the file is currently displayed, and the position of the thumb indicates where in the file the currently displayed page is located. This scroll bar can be used to navigate through the WAV file. Alternatively menu items and/or keyboard shortcuts can be used, so the scroll bar can be switched off if it is not required (which frees up a few more pixels for the waveform display). If you prefer, a user option allows you to position the scroll bar across the top of the display rather than at the bottom.

**Use of the Mouse and Keyboard**

As well as the standard uses of the mouse (eg. for selecting menu items, pressing buttons, etc), the mouse is used in Wave Repair for these purposes:

While Wave Repair is in **redraw wave mode** (identifiable due to the up-arrow or cross cursor shape and the change in background colour of the display):

1. Left click and right click don't really do anything, although they may result in a minuscule redraw if you happen to drag the mouse accidentally. Basically, if you can't see that a redraw has occurred, it probably won't be audible; if you can see an accidental redraw, simply undo it.
2. Drag from left-to-right (either button, either direction): replaces the current waveform with whatever shape you draw with the mouse.
3. Left click while holding down the control key establishes a sub-mode to allow you to redraw the waveform by simply moving the mouse. There is no need to hold down the mouse button as you draw, which may make accurate movement of the mouse a little easier. To finish the redraw, just left click again.

While Wave Repair is not in **redraw wave mode** (normal cursor shape):

1. Left click without holding down the control key: sets the start-of-selection position.
2. Left click while holding down the control key: sets a marker at the position clicked.
3. Right click without holding down the control key: sets the end-of-selection position.
4. Right click while holding down the control key: sets the audible marker at the position clicked.
5. Right click while holding down the Alt key: begins playback at the position where the mouse click was made. Playback continues for as long as the mouse button is held down. When the button is released, playback ends and a marker is placed at the position where playback stopped.
6. Drag (either button, either direction): sets the selection to be the dragged region. In the special case where an overlay block has been established (overlay blocks are discussed in the section on Repairing Damage), if a mouse drag is initiated within the displayed overlay block, then the overlay block is moved with the mouse.

In either mode, the mouse scroll wheel (if present and activated) navigates through the WAV file: by a page at a time normally, or a fragment at a time if the control key is also held down.

In normal working, Wave Repair doesn't need any keyboard input, so all keystrokes are available as shortcuts for various menu items. Therefore, rather than requiring you to press key combinations or sequences, many commonly used accelerators are single letter keystrokes (eg. the L key interpolates the left channel). This strategy is very convenient if you need to use the mouse and keyboard together: use the mouse for waveform editing, and your other hand to execute common editing functions without having to constantly move the mouse up to a menu item. All the shortcuts are shown on the corresponding menu items. In addition to those, the following additional shortcuts can be used to move through the WAV file:

1. space bar: next fragment

2. escape: previous fragment
3. return/enter: next page
4. '+' above enter: previous page

While Wave Repair is playing back the selected region:

1. space bar: places a marker
2. escape: pauses or resumes playback

While WaveRepair is recording a new WAV file:

1. space bar: places a marker.

**Navigating in WAV Files**

The slice of the file displayed in the main window is known as a page. You can move forward or back by a page using the **Position | Next Page** and **Position | Previous Page** options. If you wish to move forward or back by a smaller amount, this can be done with the **Position | Next Fragment** and **Position | Previous Fragment** options. A fragment in this context is a fraction of a page; unless you have re-defined it with the Options dialog, a fragment is 16% (about 1/6th) of a page. Moving forward or back by a fragment is useful if you see something interesting at the very start or end of the page and would like to see it in its surrounding context. All these menu options have convenient keyboard shortcuts, so you don't have to keep selecting from menus. There is an option to switch on a scroll bar along the bottom of the window that provides yet another mechanism to move around within the WAV file; which method you choose is entirely down to personal preference.

The number of samples displayed in a page is controlled by a variety of options:

1. **View | Standard Detail Scale** sets the number of samples to that defined in **Detail Scale Samples/Page** in the Options dialog. If you have not set this (the usual state of affairs), it defaults to a few samples less than the number of pixels across the main window; this means that it shows one sample per pixel, which is the best scale for visual identification of most clicks and pops.
2. **View | Selected Samples** sets the current page to be exactly the current selected region, and the number of samples per page is set accordingly.
3. **View | Zoom In** and **View | Zoom Out** simply halves or doubles the number of samples per page.
4. **View | Entire File** sets the number of samples per page to the number of samples in the file; this means that the whole file is displayed.
5. **Position | Mark Region** sets the current selected region to a start and end time, and also sets the current page to be that region.

Other options do not alter the number of samples per page, but move the display to a different part of the file:

1. **Position | Goto Position** sets the current page such that the time specified appears near the start of the page.
2. **Position | Start of File** shows the page starting at the first sample in the file.
3. **Position | End of File** shows the page ending at the last sample in the file.
4. **Position | Start of Selection** shows the page starting at the first sample in the currently selected region.
5. **Position | End of Selection** shows the page ending at the last sample in the currently selected region.

## User-Definable Macros

Wave Repair includes a facility whereby you can set up macros that execute a sequence of operations. Up to 24 such macros can be created. Macro numbers 1 to 12 are invoked by pressing one of the keyboard's 12 function keys or a corresponding button on the toolbar. Macros 13 to 24 can only be invoked by pressing one of the function keys while holding down the Shift key, in which case the macro invoked is the function key number plus 12 (eg. Shift+F3 invokes macro number 15).

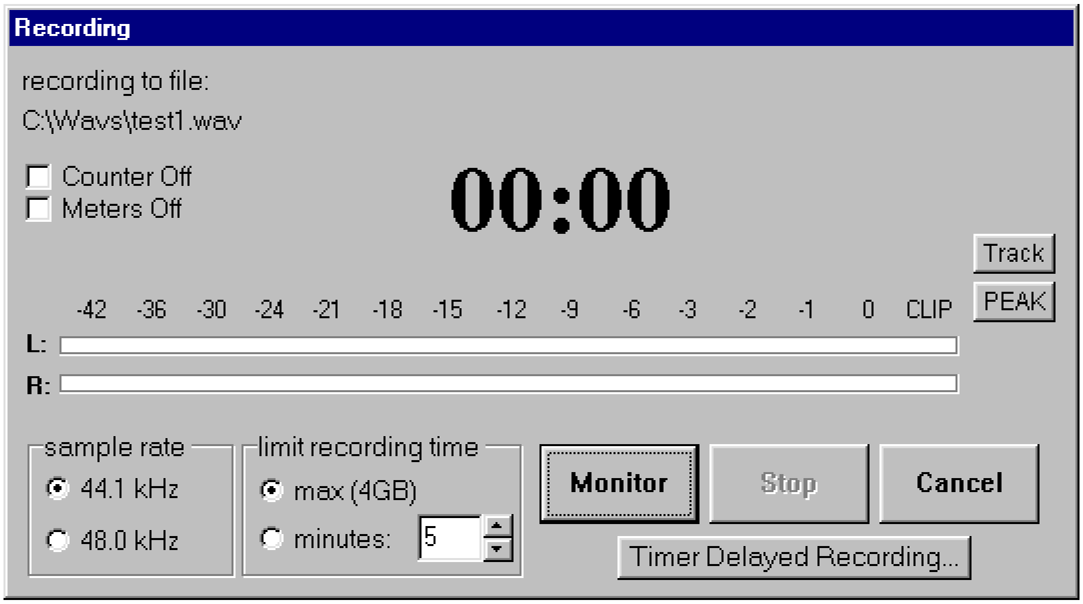
Macros may call other macros, but recursion is not supported, and will be detected and trapped while the macro is running.

During macro execution, if an operation involves the display of a dialog box which has a Cancel button, then pressing the Cancel button will abandon the remainder of the macro execution.

The macros are set up via the **File | Macros** option, which displays the <u>User Macros Setup</u> dialog.

**Recording Dialog**

When you choose to record a new WAV file, this dialog should appear:




(In some cases, you may first see a warning message. Here is a discussion of that message).

At the top left, the WAV file that will be created by the recording process is reported. Note that if this file already exists, it will be overwritten.

On the right a drop-down list of recording inputs available on the current soundcard (set to the currently selected recording input) is normally shown (unless you have instructed Wave Repair not to do so using the Options dialog). You should select the appropriate input (which is normally the **Line** input - sometimes called the **Aux** input - when recording from a stereo amplifier or tape deck). This feature has been added to Wave Repair in an attempt to overcome the confusion caused by the interface to the Windows Volume Control utility, whereby many people don't realise they are not recording from the expected input. (You can, of course, use the Windows Volume Control utility to switch record inputs if you wish). In rare cases this drop-down list will not appear. This is because the soundcard's driver does not control its inputs in the common ways that Wave Repair knows about. In this case there is no option but to use the Windows Volume Control utility (or the soundcard's own control program) to select the desired input.

The large figures display the elapsed time (in minutes and seconds) during recording.

The **sample rate** radio buttons allow you to select the rate that will be used for the recording. It defaults to the sampling rate set in the Options dialog. Note that recordings are always 16 bit stereo, and only two sampling rates are supported (44.1 and 48 kHz).

The **limit recording time** radio buttons allow you to set up a time limit for the recording. If a specified number of minutes is selected, the recording will automatically stop after that time.

This allows you to start a recording and leave it to finish unattended.

Underneath the elapsed time display are a pair of record level meters (left channel on top, right channel below). For the most part they behave like the LED meters on a normal cassette deck, and figures above give the recording level in dB below full scale.

Beyond the "LED" meters, the peak recording level reached during the current recording is displayed for the two channels. The figures are in dB below full scale. Since the figures are held for the entire recording, you can leave a recording in progress and check the peak level attained when you return; there is no need to constantly watch the meters. The button labeled **PEAK** can be pressed to reset this peak display.

The button labelled **Track** can be pressed during a recording to place cue points to mark new tracks (in preparation for later track splitting). This may be a more convenient (if slightly less accurate) way of setting up the cue points than adding them after the recording has finished.

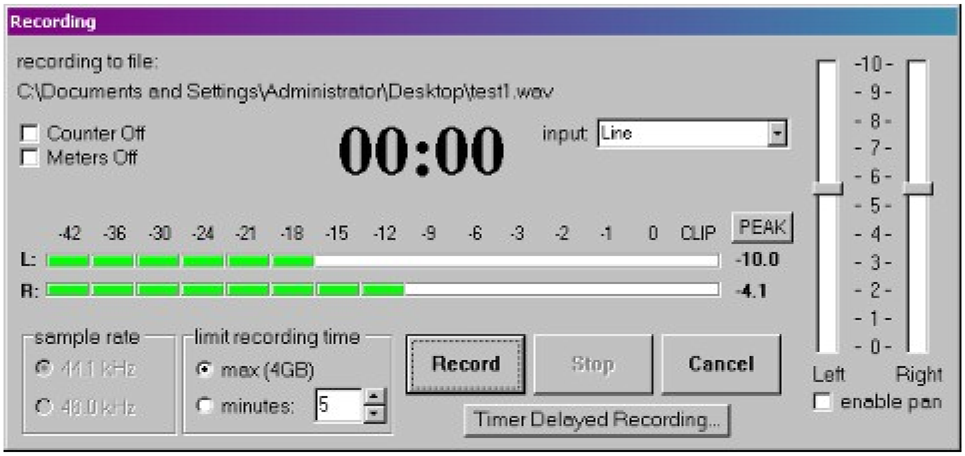
Some systems have video cards which can consume large amounts of CPU and/or hog the system bus. If you experience dropouts during recording, this could be the reason. The two checkboxes labeled **Counter Off** and **Meters Off** enable you to switch off the elapsed time counter and record level meters in order to reduce the amount of video card activity, which may eliminate such dropouts. Here is a more detailed discussion of dropouts.

When making digital recordings, it is important to avoid clipping. The last "LED" with the CLIP label comes on if clipping might have occurred. The definition of when possible clipping has occurred is controlled by the values set for **Clipping Level** and **Clipping Samples** in the Options dialog. Note that since Wave Repair only ever sees the digital samples emerging from the soundcard, it cannot know for certain that clipping has definitely occurred; the clip indicator is only a warning that it might have happened. Once this "LED" comes on, it doesn't go out for the remainder of the recording: this means that you can leave a recording in progress and if any clipping occurs during your absence it is still shown. If any such clipping is detected during the recording, then a marker is placed at each suspected clipping point. These markers will help you to quickly find the possible clips.

The **Timer Delayed Recording** button allows you to instruct Wave Repair to begin recording at some time in the future. Here are further details about this facility.

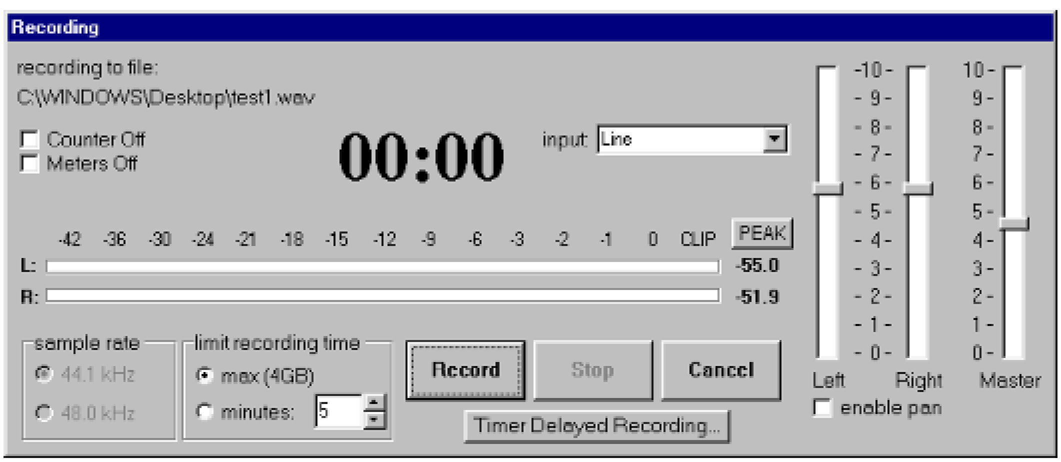
The **Monitor** button switches on the meters and monitors the soundcard input. This is a little like putting a tape deck into "record-pause" mode, and allows you to check the recording levels before starting the actual recording. At this point, the currently selected input is switched on for recording, and the record level meters begin displaying the signal level arriving there.

In addition, a pair of record level controls may appear at the right hand side:




These controls works like the sliders on a tape recorder that are used to adjust the recording level. Please note that the 0 to 10 scale is completely uncalibrated, as different soundcards exhibit different characteristics when their record level is changed. It is of course possible to adjust the record level using the standard Windows Volume Control utility (or any mixer application which came with your soundcard). The **enable pan** option determines whether the recording level of the two channels may be adjusted independently: when this option is set, they can be independently adjusted, otherwise they are locked together.

Some soundcards have an additional "Master" record level control. If your soundcard has one, then a third slider should appear that allows you to adjust this as well:




There are two reasons why these controls are provided in Wave Repair:

1. Since Wave Repair's meters are so accurate, you'll want to set the record level using them, so it is more convenient to have the level adjustment right next to the meters.
2. The granularity of movement of Wave Repair's record level slider is much finer than the one in the Windows Volume Control utility, so it allows you to set the record level much more accurately. (Note however that some soundcards may not offer such a fine degree of adjustment as the sliders might imply: you may find that moving the sliders a small amount has no effect on the record level, then the level suddenly changes as the sliders move a little

bit more).

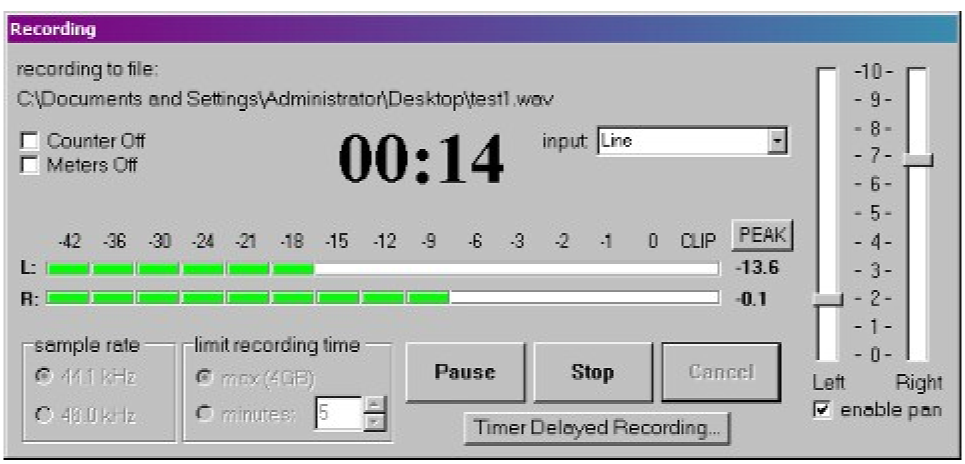
There are two main reasons why the record level sliders might not appear after pressing the **Monitor** button:

1. The currently selected input does not have a level adjustment capability (for example when using an SPDIF input).
2. The soundcard's driver returned an unexpected response when Wave Repair asked it for details of its record level control. In this case, it is necessary to use the Windows Volume Control utility (or the soundcard's mixer program) to set the record level.

In some cases, you may find that the enable pan option is disabled. This means one of two things:

1. Although the input is stereo, the recording levels of the two channels cannot be adjusted separately. (You might point out that the Windows Volume Control utility allows the balance slider of such an input to be moved. This is true, but it is due to a bug in Volume Control; adjusting the balance slider in this case does not affect the stereo balance of the input).
2. The selected input is genuinely mono. In this case, the recorded WAV file will still have two channels, but they will be identical (ie. you will end up with a "double mono" rather than stereo recording).

Once the **Monitor** button has been pressed, its label changes to **Record**: pressing the button at this stage actually begins recording to hard disk and the elapsed time counter starts. This means that to actually initiate a recording requires that you press the button twice: hopefully this is no great hardship if you don't need to check the record levels before starting. Here we see the state of the screen a short while after the recording has started, and also showing how the channels can be independently adjusted:




During recording, the label on the **Record** button changes to **Pause**: pressing it now will pause the recording (but leave the meters working). The label changes to **Resume**, and pressing it now will continue recording.

Note: do not attempt to press the **Record** button at precisely the right moment to capture the exact start of the desired recording, since there will be a few milliseconds delay while buffers are prepared, and you could miss the opening transient of the recording. It is much safer to start recording a little before the programme material begins; removing excess leading blank space is easy after the recording is finished.

While recording, you can press the space bar to place a marker in exactly the same way that you can during playback. For example, you might wish to listen and mark possible clicks as you record.

The **Cancel** button is only available before the **Record** button has been pressed, and dismisses the dialog without making a recording.

The **Stop** button ends the recording; the dialog remains visible so that you can check the final peak levels. After pressing **Stop** the label on the **Cancel** button changes to **Close**; pressing it dismisses the dialog and Wave Repair reads the recorded file into its main window.

**Options Dialog**

The Options dialog is a tabbed folder with seven tabs that allows the user to configure various behaviours of Wave Repair:

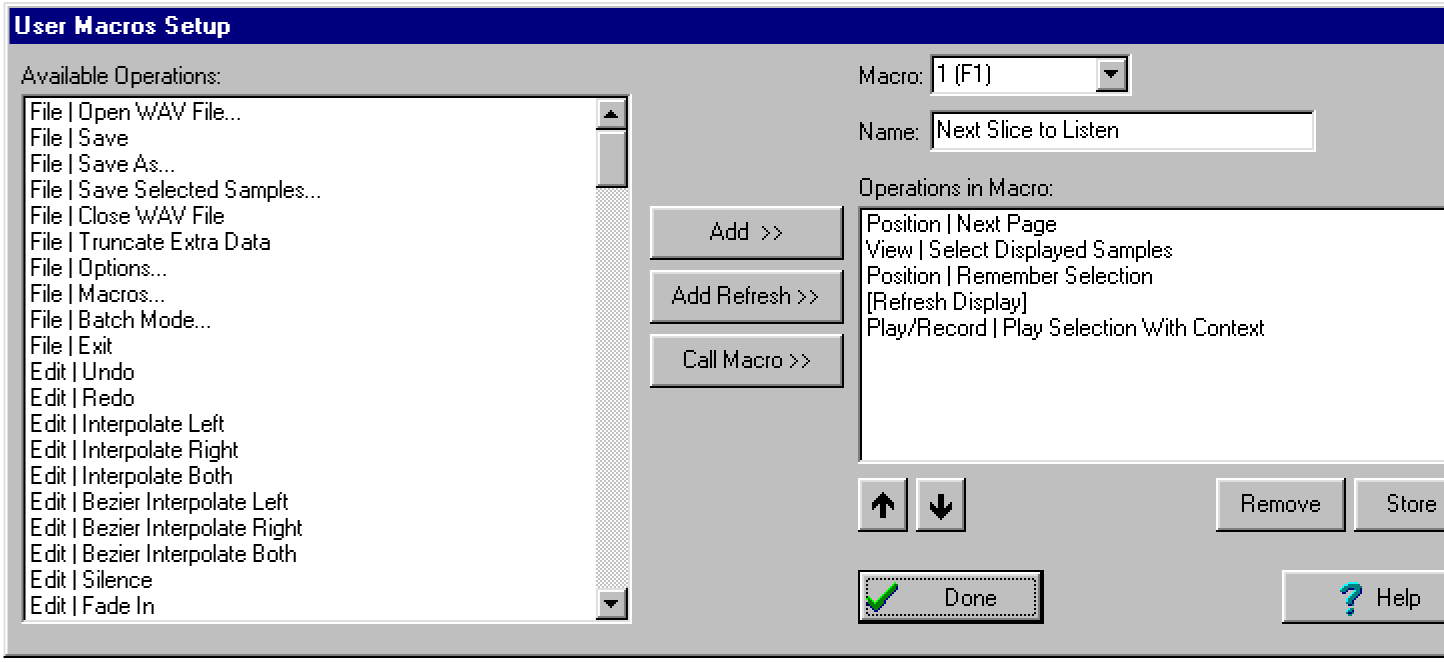
Files Tab
Interface Tab
Operations Tab
Track Splitting Tab
Playback/Recording Tab
Colours Tab
Menu Shortcuts Tab

Below the tabbed folder are three buttons:

**OK** saves any changes that have been made to the options and dismisses the dialog.
**Cancel** discards any changes and dismisses the dialog.
**Help** displays a help page appropriate to the currently selected tab.

**User Macros Setup Dialog**

This controls the user-definable macros:

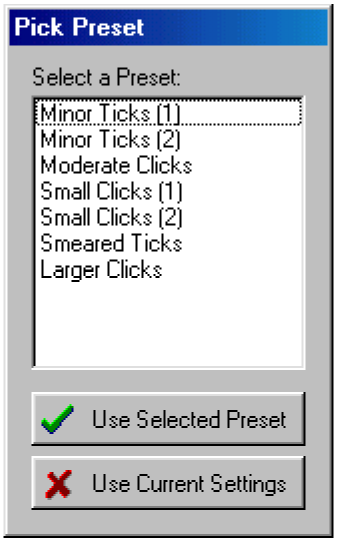



The **Macro** drop-down list selects the macro being defined. It shows the macro's number (1 to 24) and the keypress required to invoke it in parentheses. Note that the first 12 macros may also be invoked via buttons on the toolbar.

**Name** is a name that may optionally be given to the macro.

**Available Operations** is a list of the options that are found on Wave Repair’s menu bar. Any operation can be selected and included in the macro being defined using the **Add** button.

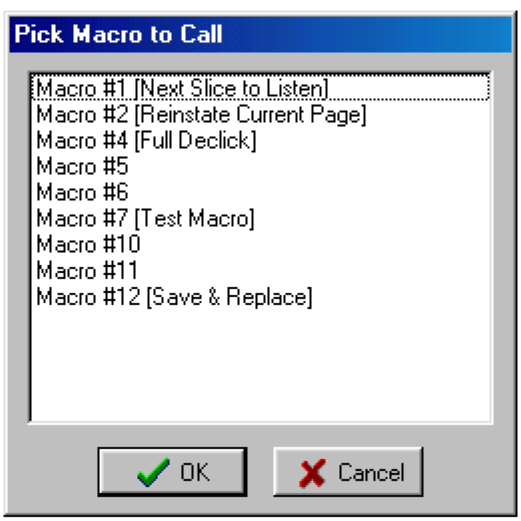
If an option that has presets available is selected (eg. find clicks, filter, equalise) and the **Add** button pressed, a dialog appears to allow you to select the desired preset:




To pick a specific preset to be used when the macro is executed, highlight the preset and press the **Use Selected Preset** button. If you press the **Use Current Settings** button, then when the macro is executed, the settings that happen to be in force will be used.

Under normal circumstances Wave Repair does not refresh the display during the course of executing a macro. This is to avoid unnecessary repaints. There are occasions when you need the display refreshed at particular points within a macro, and an instruction to do so can be added with the **Add Refresh** button.

It is possible to call another macro from within a macro; such a nested call can be added with the **Call Macro** button.
When pressed, this dialog appears:



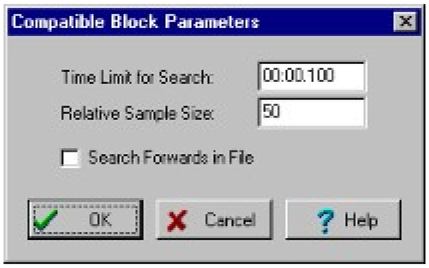

It lists all the currently defined macros along with their names (if any), and allows you to select the nested macro to be called. Note that recursive macro calls are not possible, and will be trapped when you execute the macro.

**Operations in Macro** shows the list of operations that are currently included in the macro, which will be executed in the order they appear in the list. Note that the same operation may be performed multiple times within a macro; this is why the operation is not removed from **Available Operations** when it is added to the macro.

The four buttons underneath the **Operations in Macro** list are used to move the sequence of operations within the macro (up or down), remove an operation from the macro, and store the macro definition. Note that a macro definition does not become available for use until it has been stored.

**Compatible Block Parameters Dialog**

This dialog controls the manner in which possible replacement portions of waveform are sought:




**Time Limit for Search** defines how far away from the selected region should be checked before giving up trying to find a possible replacement. It is unlikely that a compatible replacement will be found more than a few hundredths of a second away.

**Relative Sample Size** defines how closely the start and end sample values must match those of the start and end of the selected region in order for a block to be considered as possibly compatible. This value is a percentage (ie. if set to 50, this means that the start and end sample values of a possible match must be less than 50% bigger or smaller than the start and end values of the selected region).

**Search Forwards in File**, when checked, causes the search to go forwards rather than backwards. A useable replacement portion of waveform is usually more likely to be found before the selected region rather than after.

**Redraw Wave Mode**

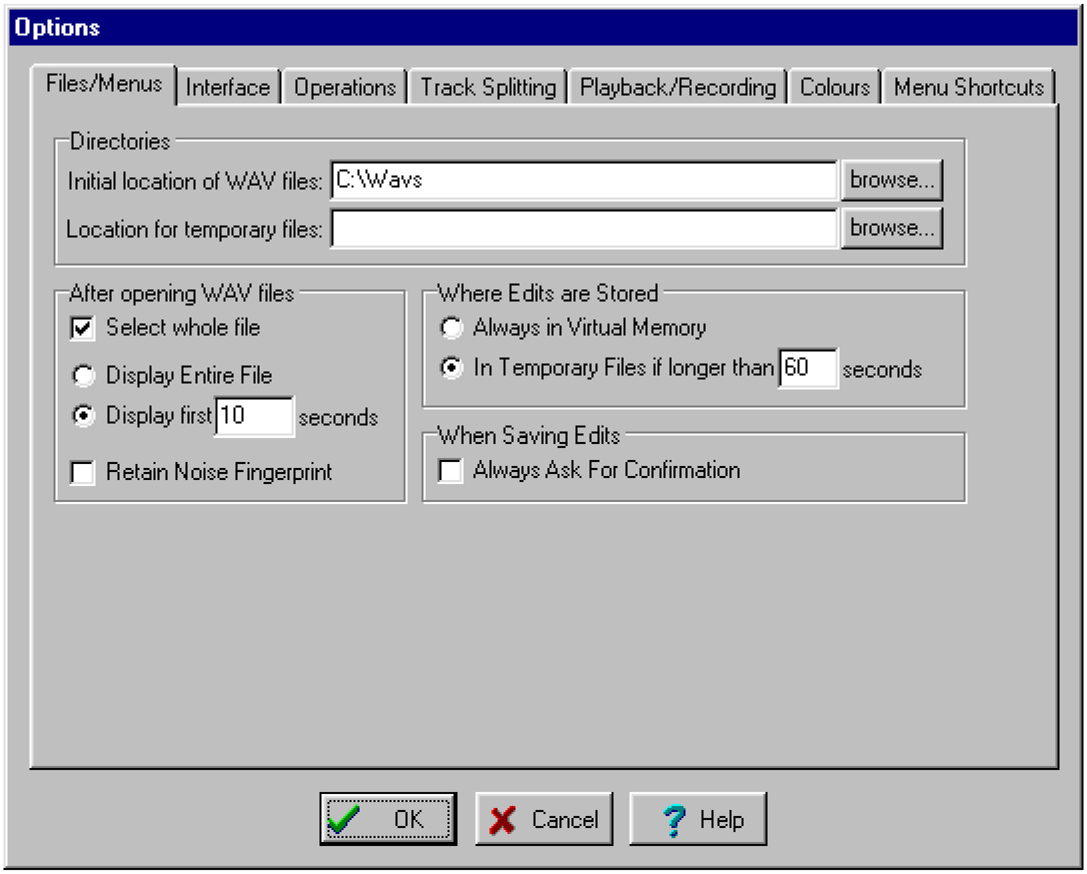
You can switch Wave Repair in and out of a special mode called **redraw wave mode**. This allows you to use the mouse to touch up the waveform manually. The menu item **Edit | Redraw Wave Mode** toggles this mode on and off. You can easily tell when the mode is on since the background colour of the display changes (unless you have deliberately set it to be the same colour as normal mode), and the mouse cursor changes from the normal one to an up-arrow or a cross, depending on which option you have chosen.

While in redraw wave mode, if you click and drag the mouse, then a new waveform is drawn where you drag the mouse, and this waveform replaces what was originally in the WAV file.

**Common Areas of Confusion**

Some aspects of the way that Wave Repair operates are distinctly different to other audio editors, and they have caused some confusion to a number of users. They are discussed here.

**Options Dialog: Files/Menus Tab**




**Initial location of WAV files** is the starting directory for any file open dialogs. If you change directory within a file open dialog, the new directory remains in effect for the remainder of the session.

**Location for temporary files** is the directory which will be used if it is necessary to create any temporary files. If it is left blank, then temporary files are created in the same directory as the currently loaded WAV file, which can result in improved performance in those cases where a temporary file must be written and then moved back over the WAV file. It is therefore a good idea to leave this blank unless disk space is limited and you need to have temporary files placed on a particular disk.

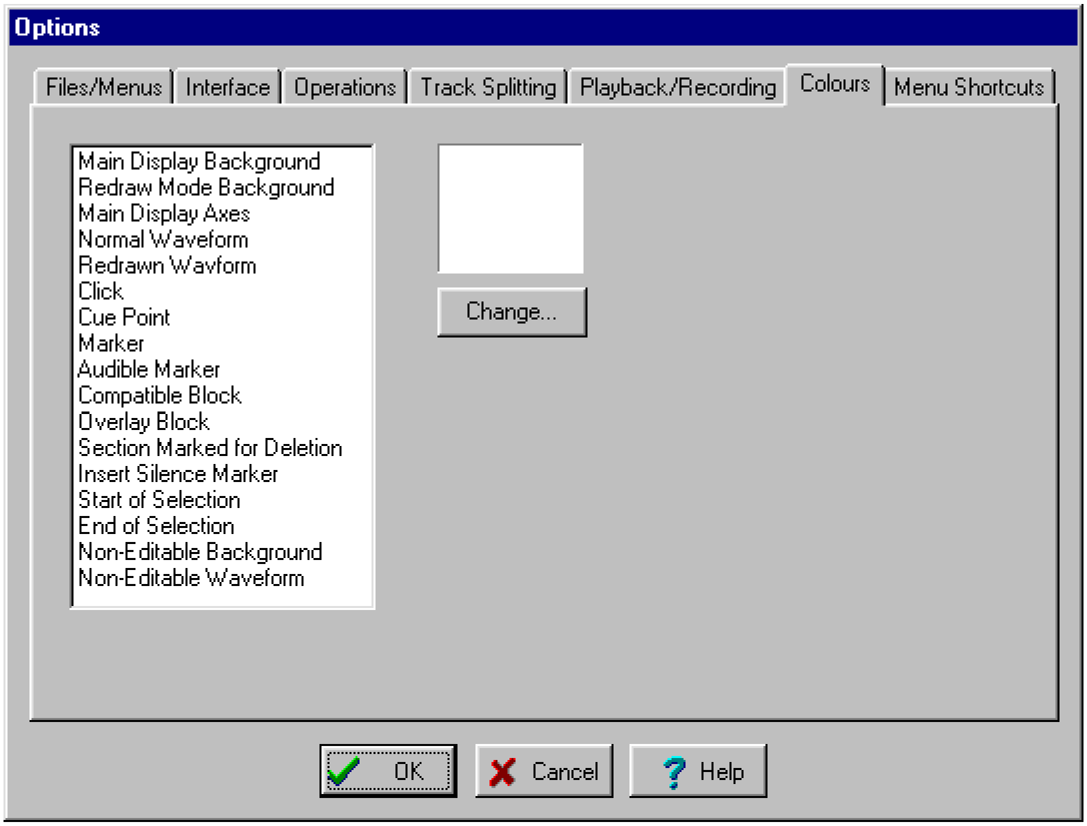
(The temporary files created by Wave Repair have names of the format **WR*nnnnnn***, where ***nnnnnn*** is a number. For example, the first temporary file in a folder will be called **WR000000**, the next one **WR000001**, etc. Therefore, if you see any file with this format of name when Wave Repair is not running, it is probably safe to delete it. Of course, it is not possible to guarantee that no other program might have created a file with such a name, but it seems extremely unlikely).

**After opening WAV files** determines what happens immediately after a file is loaded. **Select whole file** specifies that the entire WAV file should become the selected region after it has been opened. **Display Entire File** specifies that the whole WAV file should be displayed; alternatively the **Display first N seconds** option can be used to indicate what part of the start of the file should be shown. **Retain Noise Fingerprint** specifies that if there was a noise fingerprint from a previous file, it will not be discarded; this allows a fingerprint obtained from a previous file to be used in noise reduction operations on the new file.

**Where Edits are Stored** specifies what happens when a lengthy edit is made. If **Always in Virtual Memory** is selected, then edits of any length are stored in the PC's main memory, or virtual memory if the physical memory is full. For machines with a great deal of physical memory, this can be a useful option, but long edits consume a great deal of memory. For example, a 30 second edit will consume about 5MB of memory. An attempt to normalise an entire 20 minute LP side would require about 200MB of memory, and on some machines you may run out of swapfile space. Therefore it is probably better to select the **In Temporary Files** option, setting the corresponding value to a suitable number of seconds. The default of 20 seconds is suitable for most PCs, but if you have a lot of physical memory (eg. 256MB), then you could probably safely set this value higher (eg. 60 seconds). Long edits in temporary files take rather longer to read and display, but the response time remains consistent even after many huge edits, whereas if the **Always in Virtual Memory** option is chosen and the swapfile space becomes exhausted, the PC will grind to a halt. One possible drawback to using the **In Temporary Files** option is that, if your hard disk is not particularly fast, it may be unable to respond quickly enough during playback of sections that incorporate long edits, resulting in a "stuttering" sound. In this case, you would be advised to use the **Always in Virtual Memory** option.

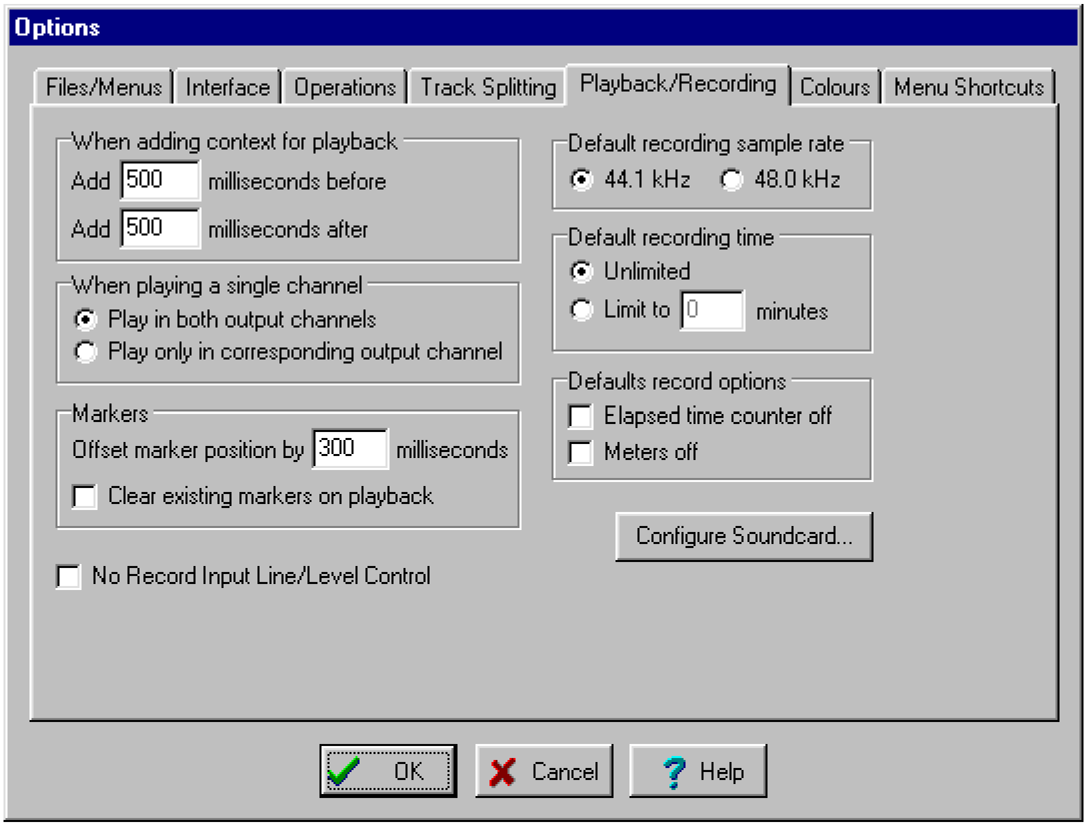
**When Saving Edits** specifies what happens when edits that have been made are saved back to the WAV file on hard disk. If **Always Ask For Confirmation** is selected, then a prompt requesting that the save should really happen is output. This can help to prevent inadvertantly saving (eg. by accidentally clicking the **Save Updates** button instead of the adjacent **Play Samples Plus Context** button).

## Options Dialog: Colours Tab




The list on the left gives the items whose colours can be changed. The box on the right shows the currently selected colour for the item. The **Change** button displays a standard colour selection dialog to allow a different colour to be chosen.

**Options Dialog: Play/Record Tab**




**When adding context for playback** gives the amount of the file which is played before and after the selected region when any playback option is used that requires the context to be included.

**When playing a single channel** is used when playing back only one of the channels. It determines whether that single channel will be played back through both channels of the soundcard.

**Offset marker position** gives a time in milliseconds. During playback or record, the space bar can be pressed to place a marker at an interesting point. Reaction time upon hearing the interesting point means that the space bar will be pressed some time after the actual position desired, so this time in milliseconds is subtracted from the time at which the space bar is pressed in an attempt to place the marker at a position that is more likely to be correct. Since each listener's reaction time may be different, the offset value can be changed.

**Clear existing markers on playback** specifies whether all playback markers should be deleted every time playback starts. It is rare that setting this option on is useful.

Under normal circumstances Wave Repair attempts to present controls to select the recording input and adjust the recording level on its Recording Screen. However, some soundcard drivers either do not support the necessary operations, or provide these facilities in an unusual way that

Wave Repair isn't familiar with, and in these cases Wave Repair fails to find the required controls and outputs a warning. If you know that your soundcard isn't supported, set the option **No Record Input Line/Level Control**; this tells Wave Repair not to even bother trying to present these controls. Here is a more detailed discussion of the situation.

**Default recording sample rate** selects 44.1 or 48 kHz as the default sampling rate for recording. (Remember that Wave Repair always records in 16 bit stereo, so there is no option to change these).

**Default recording time** specifies a number of minutes. When the Recording dialog appears, the **limit recording time** option is initially set to the number of minutes specified here; if the **Unlimited** option is selected, **limit recording time** is initially set to **max (4GB)**. This initial setting of the recording limit can of course then be changed in the normal way. Thus the **Default recording time** option is effectively a "default default". For example, if you normally record LP sides, you might find it useful to set this to, say, 30 minutes.

**Configure Soundcard** is not relevant in most cases. It is only necessary to bother with this if your computer has more than one soundcard, or your soundcard supplies multiple "virtual recording devices", or if you are having trouble with dropouts or other anomalies during recording or playback.

## Splitting Tracks

To split a large WAV file into separate WAV files for each track, special markers called **cue points** must be placed at the positions within the WAV file where each track begins.

To add a cue point, click the left mouse button at the place where the track or index point is required, and then execute Cue Points | Add Cue Point. A dialog appears which asks which type of cue point you require; note that when splitting tracks only new tracks starting at index 1 are relevant, so do not select the **Track, index 0** or **Index** options. Note that you should not attempt to place a cue point at the start of the WAV file to specify the first track; Wave Repair assumes that track 1 begins at the start of the WAV file.

There are some subtle issues concerning the exact positions where cue points are best placed. If you wish to consider these, they are discussed here.

Wave Repair also has a facility, invoked via **Cue Points | Find Tracks** which will attempt to find track boundaries for you, but you should bear in mind that automatic track detection will fail in some cases, and manual placement is more accurate in any case.

Once all the required cue points have been placed, the tracks can be written out to separate WAV files by executing **Cue Points | Split Tracks**. This displays a standard file save dialog so that you can select the name and place where the separate WAV files will be written. The file name you give will be used as a "stub" name, and the actual files created will have the suffix "01", "02", "03", etc added. You will notice that the filename suggested by default is the same as that of the WAV file to be split. so if you accept this, the individual tracks will be in files with the same name as the original large file, but with track number suffixes. For example, if you are splitting the file "ABC.WAV", the created files would be "ABC01.WAV", "ABC02.WAV", and so on.

### Padding the Last Track

If a WAV file being split is not an exact multiple of the CD block size (1/75th sec), then unless some action is taken the last track written out will itself not be a multiple of the CD block size. In some cases this may be acceptable, but in general it is a good idea to make every track an exact number of blocks. Therefore, when **Cue Points | Split Tracks** is invoked, you may be asked whether you wish the last track to be padded with zeros. In most cases it is a good idea to agree to the padding, which adds enough zero-valued samples to the end of the last track in order to make it an exact number of blocks.

You can invoke **Cue Points | Split Tracks** even if there are no cue points set. This allows you to pad a WAV file which contains a single track to a CD block boundary.

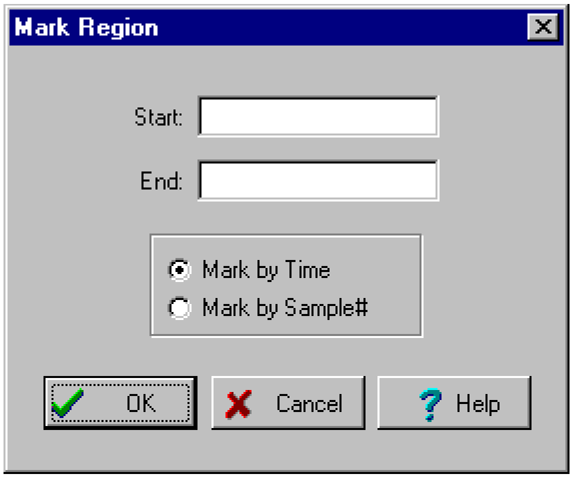
**Goto Time Dialog**


This dialog moves the display to a particular time offset within the WAV file. Once there, the display is positioned so that the requested time is one third of the way from the left hand edge.

Times should be given in the format MM:SS.FFF where MM is minutes, SS is seconds, and FFF is fractions of a second. If times are currently being displayed in CD-frames, FFF should be given as a frame number (1 or 2 digits); otherwise it is tenths, hundredths or thousandths of a second, depending on whether you give 1, 2 or 3 digits.

The radio buttons allow you to optionally set whether the start or end of selection markers should be set to the specified time.

**Mark Region Dialog**

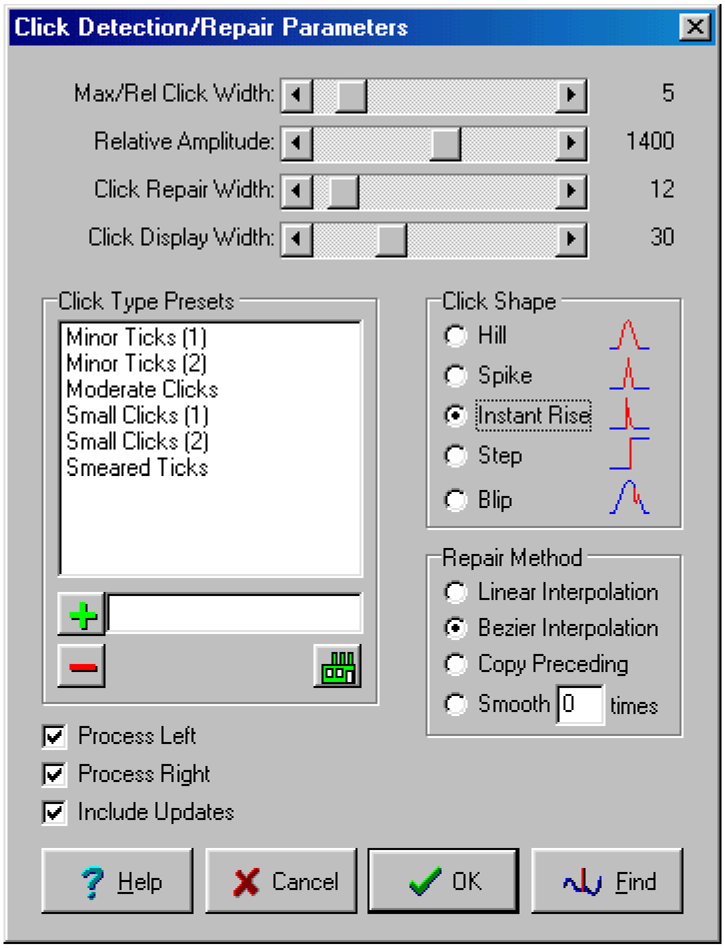



This is a quick way to set the start and end positions of the selection to exact time offsets or sample numbers. After setting the selection boundaries, it also sets the display to show the entire selection.

The radio buttons allow you to choose between setting by elapsed time into the WAV file (which is the more common method), or by exact sample number (samples in the WAV file start at 1).

If giving the positions as times, they should be given in the format MM:SS.FFF where MM is minutes, SS is seconds, and FFF is fractions of a second. If times are currently being displayed in CD-frames, FFF should be given as a frame number (1 or 2 digits); otherwise it is tenths, hundredths or thousandths of a second, depending on whether you give 1, 2 or 3 digits.

**Click Detection/Repair Parameters Dialog**




This dialog adjusts the factors which the automatic click detection process uses to recognise potential clicks. Note that two adjustable parameters (**Search Window Size** and **Relative Offset**) have been removed as from release 4.3. This is because they were of extremely marginal value, and served only to confuse most users.

**Max/Rel Click Width** differs depending on the type of click shape being sought. For "hill" and "spike" click shapes it is the maximum number of samples during which a potential click should resolve itself. For the "blip" click shape it is the maximum width of a wave peak (expressed as a percentage of the average width of peaks within the window) that will be considered a possible click. The larger the value, the more clicks will be found. Note that this parameter has no effect on the detection of "instant rise" and "step" click shapes.

**Click Repair Width** sets the number of samples around a click which will be amended if it is automatically (rather than manually) removed. If this value is too small, the edges of the repair might themselves create small clicks; if it is too large, the repair is more likely to remove nearby music transients.

(Note that the above two parameters are inter-related, and adjusting one may cause another to change).

**Click Display Width** sets the number of samples which will be selected before and after the detected click when it is displayed in response to a **Find Next Click**. The rationale behind this option is that when studying a potential click, it is often useful to toggle between displaying a page at one pixel per sample (to view the context around the click) and displaying a smaller range around the sample in order to accurately review and/or repaint the waveshape. By setting an appropriate width of selection, this toggling can easily be done with the D and S keys, which means that it is not necessary to use the mouse to select an appropriate region, thus avoiding the need to keep switching in & out of **redraw wave mode**.

**Relative Amplitude** sets the size of a sudden change that should be considered to be a potential click. This is expressed as a percentage of the average amplitude change within the search window; only changes in amplitude greater than this will be considered as potential clicks. This parameter is used to eliminate large changes when the whole window is effectively filled with large changes; thus where there is a high level of high frequency energy, a change has to be correspondingly greater to be considered as a click. The smaller this parameter, the more clicks will be found.

**Click Shape** sets the kind of clicks to look for:
1. **Hill** looks for clicks which rise and fall (or fall and rise) by at least the set relative amplitude within the space of the set maximum click width.
2. **Spike** is similar to **Hill**, except that there must be at least one point within the click width where the difference between two adjacent samples is greater than the set relative amplitude. This is characteristic of many (but not all) LP clicks which are caused by dust particles, since they tend to shoot up very rapidly and then fall back more slowly.
3. **Instant Rise** looks only for clicks which shoot up (or down) by the set relative amplitude in a single sample, **and** where the next sample moves in the other direction; this is a subset of **Spike**, and is useful when looking for subtle clicks at a very low level.
4. **Step** simply looks for clicks that rise (or fall) by at least the set relative amplitude in a single sample, whether or not the next sample goes in the other direction. I have seen a few examples of vinyl damage which appear as steps, but they are quite rare.
5. **Blip** is rather different. Here, the average width of waveform peaks is measured within the search window, and then narrow peaks are sought. This click shape is sometimes useful when small clicks imposed on top of smoother waveforms are missed by the other click shapes.

**Repair Method** defines how a click is to be repaired (if it is repaired automatically):
1. **Linear Interpolation** means that clicks are repaired with simple straight line interpolations. In general, Bezier curve repairs are preferable, but if the repair width is significant (say greater than 20 samples) then a linear repair can be better.
2. **Bezier Interpolation** means that clicks are repaired with smooth curves (called "Bezier curves"). For short duration clicks this usually gives a good repair.
3. **Copy Preceding** means that the click is replaced with the immediately preceding section of waveform. This is usually only worth trying if the repair width is large and linear interpolation hasn't worked.
4. **Smooth** replaces the click with a smoothed version of the same waveform. This repair method

is like manually selecting the click and executing **Edit | Smooth Abrasion**. The associated parameter defines how many smoothings are applied.

The **Click Type Presets** section of the dialog allows you to store and retrieve useful combinations of settings. Each stored preset has a name, shown in alphabetical order in the list box. Note that the presets encompass only those attributes controlled by the adjustable parameters, click shapes, and repair methods; the check box options are not included in the presets. Three small buttons control the stored presets:

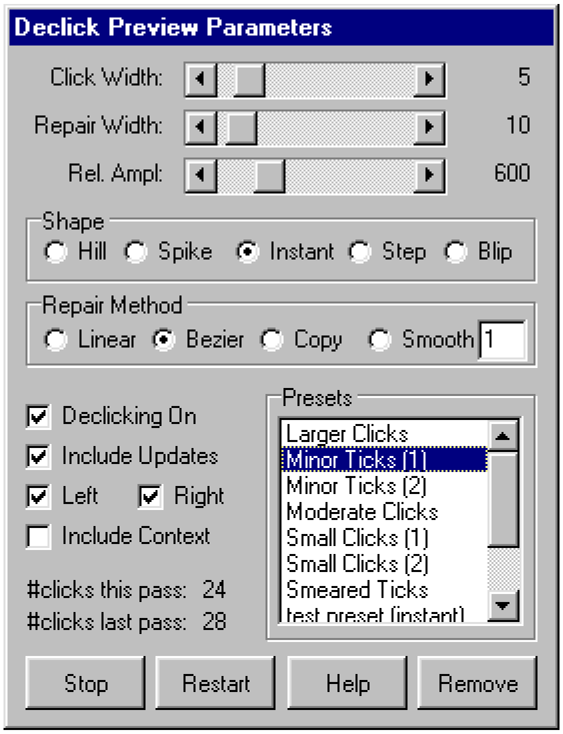
1. The button with the red minus sign deletes the currently highlighted preset.
2. The button with the green plus sign adds the current settings as a new preset. A name for the new preset must be given in the edit box next to the button. Note that if you type the name of an existing preset, it will be overwritten by the new settings. (Note also that when you select an existing preset, its name is placed into the edit box, which is convenient if you wish to adjust the parameters for a preset and store the updates).
3. The button with the picture of a building (it's supposed to look like a factory) installs/restores a set of "factory default" presets that I have found to be fairly useful for typical LP damage. Any presets you may have created are left untouched; only the factory default presets are affected. A warning is given before any preset is overwritten.

**Process Left Channel** and **Process Right Channel** do exactly what they imply: each channel is included in the click detection and repair procedure only if the corresponding option is checked.

**Include Updates in Detection** determines whether any changes that you have made to the WAV file, but not yet saved, should be included in looking for clicks. If unchecked, the click detection process runs against the WAV file on disk; if checked, any unsaved changes are also considered. If you know that there are no unsaved changes in the region to be scanned, leaving this option unchecked will result in a slightly faster scan.

The **Find** button is simply a quick way of saving the settings and immediately running **Find All Clicks** without the need to separately invoke it from the **Declicking** menu.

**Declick Preview Dialog**




This is a modeless dialog which appears while **Declicking | Declick Preview** is in operation. It allows the declicking parameters to be adjusted while the selected samples are playing. The initial settings are inherited from whatever was last set in the Click Detection/Repair Parameters dialog. When Declick Preview ends, the values which have been set here are forwarded back to the main automatic click detection system.

**Click Width, Repair Width**, and **Rel Ampl** are the same parameters as **Max/Rel Click Width**, **Click Repair Width**, and **Relative Amplitude** in the **Click Detection/Repair Parameters** dialog.

**Shape** is the same parameter as **Click Shape** in the **Click Detection/Repair Parameters** dialog; note that the **Hill** shape requires more CPU power to detect, and may not operate properly on slower computers.

**Repair Method** is the same parameter as in the **Click Detection/Repair Parameters** dialog. However, note that the **Copy** method may not work perfectly during declick preview if the click happens to be at the start of a playback buffer. This is because the replacement section of waveform required is not available in this case.

If the **Declicking On** option is not checked, Declick Preview operates in a pass-through mode without repairing any clicks. This allows a simple before-and-after comparison to be made. The **Include Updates** option is the same as in the **Click Detection/Repair Parameters** dialog. The **Left** and **Right** options allow declicking to be restricted to specific channels.

**#clicks this pass** and **#clicks last pass** are simply counts of the number of clicks which have been detected in the current and previous passes through the selected region. This can help you to judge the effect of making changes.

Any presets which may exist are available for use during declick preview, but note that new presets cannot be saved here (they can only be saved in the Click Detection/Repair Parameters dialog).

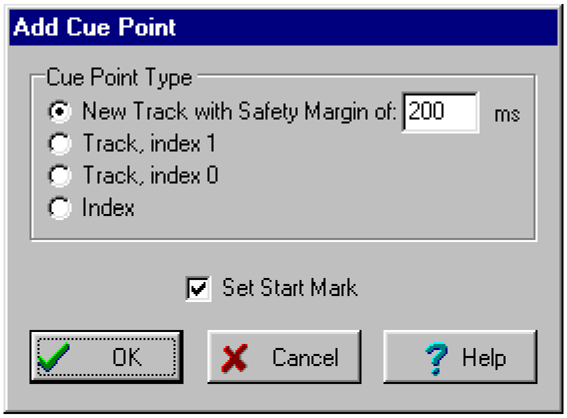
If **Include Context** is checked, an additional amount before and after the selected region is included in the playback.

The **Stop** button ends Declick Preview, which can also be ended via the **File | Kill (Stop) Output** menu option or by pressing the **Stop Playback** button on the toolbar.

The **Restart** button interrupts the current pass through, and begins playback again at the start.

The **Remove** button ends the declick preview and immediately removes the clicks found without having to separately invoke **Find All Clicks** and **Remove All Clicks** from the Declicking menu.

## Add Cue Point Dialog




When **Cue Points | Add Cue Point** is invoked, a cue point is set at a CD frame boundary near the start of the selected region. This dialog determines what type, and exactly where, the cue point is added.

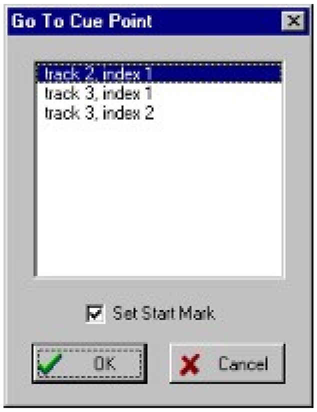
The default is **New Track with Safety Margin**, and this requires some explanation. The most common type of cue point you might want is for a new track at index point 1. In such a situation, it is common practice to place the cue point a short time (usually about 200 milliseconds) prior to the actual start of the music. The purpose of this is to act as a "safety margin" so that CD players which mute their output while moving to a new track will have time to unmute before the music begins; it also caters for older CD players which can be a little inaccurate in positioning to a new track. Rather than having to find the start of the track and then work out the safety margin yourself, this option makes it easy to just place the start of the selected region exactly on the beginning of the music and have Wave Repair incorporate the safety margin for you.

The other options do not incorporate a safety margin, allowing you to place cue points exactly where you wish, although you should be aware that since a cue point must be on a CD block boundary, the resulting cue point is placed at the first block boundary after the start of the selected region.

A new track at index point 0 is useful if you wish to have a "count down" during the intro to a track on a CD burned from the WAV file. New index marks are generally used to mark sections within tracks.

The **Set Start Mark** option allows you to reset the start of selection to the block boundary where the cue point is actually placed, which provides a simple way for you to listen to how the track will start.

**Goto Cue Point Dialog**



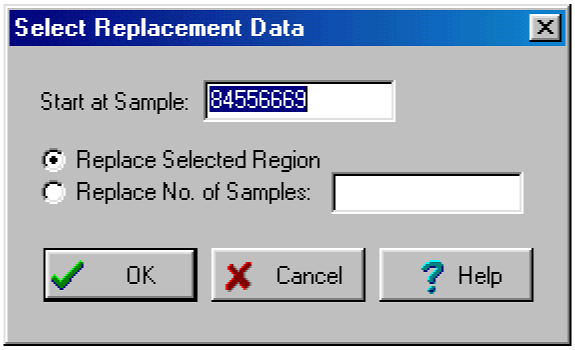

This dialog allows you to select any of the cue points currently set in the WAV file and move the display to that position.

The list gives the cue points available.

If **Set Start Mark** is checked, then the start of selection is moved to the chosen cue point, which may be useful if you wish to listen from precisely the location of a cue point.

**Select Replacement Data Dialog**

When **Edit | Replace From File** is invoked, a standard file open dialog appears with which the user selects a file from which replacement data will be read; we will refer to this selected file as the **source file**. After this, the following dialog appears:




**Start at Sample** is the sample number in the source file where the replacement data starts. (Samples are numbered starting at 1). The default is the sample number of the current start of selection. The rationale behind this is that you may have various versions of the WAV file that have been passed through different processors; if you wish to replace a section with the corresponding section from one of the other versions, you will want to start at the same sample number. (The other common starting sample number from the replacement file would be 1, which of course is easy to specify).

If **Replace Selected Region** is checked, then sufficient samples are copied from the source file to replace the selected region in the current WAV file. This is the most likely requirement, where you will have selected the precise region to be replaced.

If **Replace No. of Samples** is checked, then the required number of samples to be replaced should be entered into the box following this option, and that number of samples will be copied.

Note that if the required number of samples to be copied is greater than is available in the source file, or if the replacement would extend beyond the end of the current WAV file, the number of samples actually copied is reduced accordingly.

**OK** executes the replacement and dismisses the dialog.

**Cancel** dismisses the dialog without performing the replacement.

**Resetting Shortcuts**

Here is what to do if you manage to get your menu shortcuts all messed up:

1. Open the file WAVREP.DAT in any text editor (eg. Notepad). You will find this file in the same directory as WAVREP32.EXE. You will see that WAVREP.DAT looks just like a normal INI file.

2. Find the section headed [Shortcut]. Delete this entire section (ie. up to, but not including, the next section header enclosed in square brackets).

3. Save WAVREP.DAT.

You should now find that the next time you start up Wave Repair, the menu shortcuts will have returned to their default settings.

**Markers**

While processing a file, you may notice places of interest to which you would like to return at some later time. Rather than trying to remember where all these places are, you can mark them. The marks are displayed as green lines, which provide a visual clue to the places you wish to revisit.

There are two situations in which you may wish to add markers:

1. During the course of playing the selected region (or recording a new WAV file), you may hear many instances of potential damage to which you would like to return for further investigation. These places can be marked by pressing the space bar as you hear them, which causes the current position in the playback or recording to be marked.

   Clearly, there will be some delay between you hearing something to be marked and pressing the space bar, primarily due to simple reaction time. Since each user's reaction time may be different, the Options dialog provides a place for you to set up an appropriate delay (in milliseconds). Some experimentation may be necessary to determine your own reaction time, but around 300 milliseconds is a good place to start.

2. While you are processing the file manually (ie. not during playback), you may notice suspicious looking waveforms that may be damage of some sort, but you would rather do something else before returning to them. These places can be marked by holding down the control key while clicking on the position with the left mouse button.

There is one special marker, known as the "audible marker", and it is distinguished from other markers in being shown as a red (rather than green) line. The audible marker is placed or moved by right clicking with the mouse while holding down the control key. During playback, a distinct "click" is output at the audible marker's position. The purpose of this is to enable you to listen to the relative locations of a defect being sought and a known click (ie. the one at the audible marker). This can help to home in on a glitch which is difficult to track down.

The markers consume very little resource, and so there's no need to clear them unless you have so many that they cause confusion, in which case you can remove them with **Markers | Clear Markers in Selection** or **Markers | Clear All Markers**. These options remove the audible marker as well as normal markers.

**Recording the WAV File**

The first step is to record the WAV file onto hard disk. Wave Repair can do this for you (via the Recording dialog), or you might prefer to use another recording package: Wave Repair is quite able to read in a standard WAV file from hard disk. One thing to bear in mind is that Wave Repair only works on 44.1 or 48 kHz 16 bit stereo WAV files, so you should be sure to record in that format.

If your eventual aim is to write the results of the restoration to CDR, remember that CDR must be written at 44.1 kHz, so you should record at that sampling rate. If you already have your analogue sources digitised on some external format (eg. DAT or Minidisc) then it makes sense to transfer to hard disk digitally if you have a soundcard capable of digital I/O: this avoids the possible degradation involved in an extra D/A and A/D conversion stage. However, if you have DAT tapes recorded at 48 kHz and wish to write the results to CDR, then you will need to perform a sample rate conversion at some stage. Wave Repair does not support sample rate conversion; you will need to use another package (such as GoldWave or CoolEdit).

When processing an LP side or lengthy tape, do not be tempted at the recording stage to try and split the tracks into separate WAV files, since you are likely to lose the timing of the inter-track gaps. (You might be surprised at how unnatural the results sound if the inter-track gaps change on an album with which you are familiar). Moreover, if tracks segue into one another, it will be very difficult to arrange for the segues to remain correct if you split the tracks at record time. When the time comes to write a CDR, you may need to split the tracks into separate WAV files: this is simple using Wave Repair, and is described in Preparing to Write a CDR.

Don't try to cue up the recording to start and stop precisely. Instead, allow a little time before and after the music to be included in the WAV file. It is easy to remove this later by using Deletion Marks, and removal by this method is far more accurate than trying to start and stop a recording at the right moment.

While recording is in progress, you can press the space bar to place markers to which you can return when the recording has finished. You might, for example, like to listen and mark possible damage as the WAV file is being recorded. Wave Repair also automatically places markers wherever it detects that clipping may have occurred during recording.

What if you experience dropouts, "stuttering", or other anomalies while recording?

In the unlikely event you find that recording to hard disk does not go smoothly, the most likely cause is a system configuration issue. Here are some notes which may help you to resolve the problem.

**Preparing to Write a CDR**

Once you're happy with the results of restoration, you might wish to write a CDR. The CD writing packages on the market fall into two main camps:

1. The majority require each track of an audio CDR to be in a separate WAV file. (Roxio's Easy CD Creator is the most widespread example). In order to use a package of this type, you will need to split the tracks in your long WAV file into separate WAV files.

2. Other packages can write multiple tracks from a single WAV file. (Goldenhawk's CDRWin is a common example). In order to use a package of this type, you need to write a text file (known as a "cue sheet") which defines where within the WAV file each track begins.

Wave Repair's cue point facility can help with both tasks. By placing cue points at the start of each track, you can either split the tracks into separate WAV files or write out a CDRWin-format cue sheet. Note that Wave Repair automatically assumes that track 1, index 1 begins at the start of the WAV file, so you should not place a cue point there. If you choose to write a cue sheet, Wave Repair can insert some limited CD Text information if you require.

You can use the **Cue Points | Find Tracks** option to attempt to find the track split points automatically. The characteristics of track boundaries used by this automatic process are defined in the Options dialog You should bear in mind that this automatic process cannot be perfect, but it might provide a useful starting point.

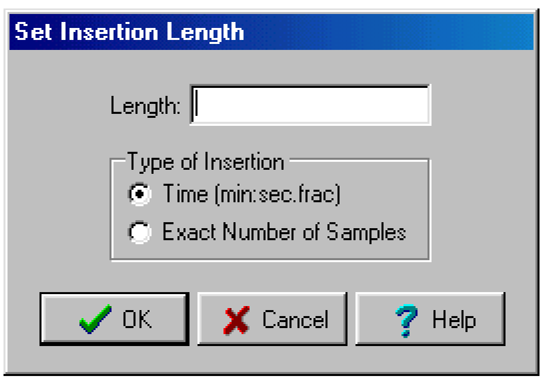
**Deletion and Silence Insertion Marks**

As has been stated elsewhere, Wave Repair is not a general purpose WAV editor, and does not include a copy/cut/paste facility of the kind found in many other WAV editors. However, one valid way of repairing some kinds of damage in recordings from analogue sources is simply to delete a few samples which include the damage. Another operation which is often useful is the insertion of a short section of extra silence. Therefore Wave Repair has the ability to perform these specific tasks.

Deleting from or inserting into a WAV file involves re-writing the entire file. To do so every time you wish to remove a short section or insert a period of silence would be unbearably tedious, so instead of doing that, Wave Repair allows you to mark a section as "to be deleted", and to place markers where silence is to be inserted. Many such sections can be marked in this way, and when you are ready to re-write the entire file it can be done just once, removing all the marked sections and inserting all the desired silences at the same time, by invoking **Edit | Execute Deletions/Insertions**.

To mark a section as "to be deleted", make it the selected region and invoke **Edit | Mark Selection for Deletion**. While a section is marked for deletion, it is displayed on the waveform display in red, and it is also skipped during playback to allow you to hear how the re-written WAV file will sound. Apart from these two behaviours, a section that is marked for deletion is treated exactly like any other section of waveform.

To indicate where a period of silence is to be added, place the start of the selected region at the position where the silence is to be placed, and execute **Edit | Add Silence Insertion Mark**. The following dialog will appear:




This allows you to specify the length of the silence to be inserted, which may be given as a time (in mins:secs:millisecs) or as an exact number of samples. Once placed, the mark is shown by a vertical line, together with the length of silence as a time.

Unfortunately the way that Wave Repair is structured makes it impossible for the added silence to be shown on the waveform display, or included during playback, until the WAV file is re-written and the silence physically added to the file. Therefore, if you are trying to add a suitable length of silence between tracks it is not possible to preview how it will sound. In this case it is best to overestimate, insert that amount, rewrite the file, and then place deletion marks (which can be previewed) to remove any excess silence.

Inserting silence into a file also allows extra samples to be inserted from another WAV file. To do so is a two-stage process: insert silence of a suitable length, re-write the WAV file to physically extend the file, then overwrite the silence with the desired samples from the other file using **Edit | Replace From File**.

To re-write the whole file so as to remove all marked sections and add all new silences, invoke **Edit | Execute Deletions/Insertions**. (Note that if a silence-insertion mark happens to be placed inside a region marked for deletion, the silence-insertion mark will be ignored). This operation writes the results to a temporary file, and once that is successful it replaces the original WAV file. This is done for safety reasons, since if you were to lose power to the PC during a re-write over the original file, the file would become corrupted. Note that this does mean there must be enough disk space to hold this second temporary copy of the file.

When samples are deleted/inserted, any existing markers, cue points and clicks will be automatically adjusted. Note, however, that since cue points must lie on CD block boundaries, they are likely to move slightly (to the next CD block) if the deletions and insertions leave them away from a block boundary. Be aware also that that by inserting and/or deleting samples from the WAV file, any associated files (ie. **.mkr** (markers), **.cue** (cue sheet) and **.clk** (clicks)) will be rendered inaccurate. Therefore if you wish to keep such files up to date, load them into Wave Repair before executing the deletions/insertions and then write them back out after executing the deletions/insertions.

**Broadband Noise Reduction**

Broadband noise reduction is the process whereby a constant noise (such as tape hiss or general "vinyl roar") which appears throughout the recording is reduced.

As a general rule broadband noise is not something which affects vinyl replay very much, although of course historical LPs made from early tape recordings may include some level of tape hiss. More modern records tend to have background noise which is of much lower frequency and not nearly as noticeable as tape hiss. Nevertheless, you may feel that some reduction is appropriate at specific places (eg. quiet sections of the music, and in the gaps between tracks).

If you decide to perform both decrackling and broadband noise reduction, you should do the decrackling first, and a fresh noise fingerprint should be taken for the broadband noise reduction after decrackling. Whether you do declicking before or after broadband noise removal is entirely up to you; it doesn't matter in which order they are performed.

Broadband noise reduction works by analysing a "fingerprint" of the noise on its own to determine the characteristics of the noise. This is then subtracted from the region where noise reduction is required. Therefore if you wish to perform broadband noise reduction, it is necessary to do so before trimming away the leading and trailing sections, as these sections are the best places to obtain the noise fingerprint.

So, before noise reduction can be performed, select a region that contains only background noise, without any large clicks or musical signal. The run-in groove (excluding the sound of stylus touching down), or a gap between tracks, is a good place to find a suitable section. Once such a region is highlighted, invoke **Noise Reduction | Get Fingerprint**. The Set Precision dialog will appear which is used to choose how accurate the fingerprint analysis (and subsequent noise reduction) will be. The greater the precision, the better will be the quality of noise reduction, at the expense of taking longer to process.

Once the fingerprint has been taken, select the region to have background noise reduced and invoke **Noise Reduction | Reduce Noise**. The Reduce Noise dialog will appear which allows you to select how aggressively to subtract the noise from the signal. Excessive amounts of noise reduction will result in objectionable artifacts, usually a "distant, phasey" sound. The aim is to find a level which produces a worthwhile reduction in noise without affecting the music.

Due to the complex nature of broadband noise reduction, there is no realtime preview available. It is therefore advisable to experiment on short sections to find a suitable level of noise reduction before processing a lengthy section. After noise reduction is complete, you should review the results to make sure they are satisfactory before saving.

**Trimming & Fading Leading and Trailing Blank Areas**

You will almost certainly wish to trim away unwanted leading and trailing sections of "dead space".

It is a good idea to try and get the music to start very close to the beginning of the WAV file; I recommend that you include no more than 1/5th second before the music starts, and bring the level up from zero to full volume (using **Edit | Fade In**) during this 1/5th second lead-in time.

At the end of the WAV file, don't chop the music off suddenly as it ends. It is usually better to leave a few seconds of trailing space after the music ends, and gradually bring the level down from full volume to zero (using **Edit | Fade Out**) during those few seconds.

Once you've decided which parts of the leading and trailing areas you wish to trim away, there are two basic approaches:

1. Select that part of the WAV file you wish to keep (ie. make the selected region start about 1/5th second before the music starts, and end about 3 seconds after it ends), and use **File | Save Selected Samples** to write the part you want to keep to a new WAV file. Once the new WAV file has been written, you can delete the original file.

2. Mark the leading and ending sections as areas to be removed (using **Edit | Mark Selection for Deletion**) and then use **Edit | Execute Deletions** to remove them.

If you have recorded an entire LP (ie. both sides) into a single WAV file (by using the pause facility of the recording screen while you turn the record over), then you will probably want to remove some "dead space" in the gap between the two sides. In this case, the second of the above two methods is clearly preferable, since you can also mark the dead space between the sides and preview what the transition will sound like.

**Markers Menu**

Goto First Marker
Moves the display to a page containing the first marker that has been set.

Goto Next Marker
Moves the display on to the next marker that has been set.

Clear Markers in Selection
Deletes all markers within the selected region.

Clear All Markers
Deletes all markers in the entire file.

Find Clipping in Selection
The selected region is scanned, looking for possible clipping. What constitutes clipping is defined by **Clipping Level** and **Clipping Samples** in the Options dialog. A marker is added at every place where clipping is suspected.

Save Markers in File
Saves all markers in a file.

Load Markers from File
Reloads the markers that were previously saved in a file.

**Locating Damage**

The first thing to do is listen to the WAV file (preferably on headphones, which are more revealing of damage than loudspeakers), and identify the general region where the damage lies. Home in on the damage by selecting smaller and smaller regions that contain it. There will come a point where you cannot home in any further because the playback is so short that nothing intelligible can be heard. My own technique for homing in is to start by selecting a region that includes the click, making sure that the region extends for about half a second to one second beyond the damage. Then, progressively move the start of selection forward, listening each time with **File | Output (Play) Selection**, until the selection starts as close to the damage as possible. If you move the start of selection beyond the damage, just move it back again until it reappears when you listen. This way you should find that you will have placed the start of selection within a few thousand samples of the damage, which now makes finding it visually must easier.

Other users have reported that they have success by progressively moving the end (rather than the start) of the selected region back towards the audible glitch, listening out for when it disappears.

Finally, it may be helpful in some cases to place the audible marker (by right clicking while holding down the control key) at a position near the defect so you can listen to the relative locations of a known click (the audible marker) and the one being sought.

Some regions of damage can be quite widespread (eg. during a quiet section there may be a continuous background of ticks you would like to remove). There is no problem in selecting this entire region; it can be treated in just the same way as a smaller one, but will just take more time.

Once you have zoomed in as far as possible, you have a portion of waveform which you know to contain the damage. What you do now depends on the kind of damage and the size of the region.

Clicks, Pops and Ticks

These are the most common form of vinyl damage and are thankfully the easiest to identify. The visual waveshape is a characteristic up-and-down (or down-and-up) spike, often rising (or falling) to the maximum deflection in just one or two samples.

Where you are trying to remove a single click, you should have been able to zoom in fairly tightly (to within perhaps 1/10th of a second). In this case, locating the damage is probably easiest by simply scanning through the region manually. Switch on **View | Maximise Amplitude** and **View | Standard Detail Scale** to get the best visual representation of the waveform. Now, starting at the beginning of the selected region (via **Position | Start of Selection**), begin scanning through the region page by page (the **enter** key goes forward a page at a time). You should see the characteristic up-and-down spike quite easily. Note that there may be many much smaller spikes that happen to be inaudible; what you're looking for is a fairly obvious one. If you get to the end of the selected region (easily noticeable because the display changes from white-on-black to black-on-white) without having spotted the spike, try another scan: you may simply have missed it first time round. Alternatively, the click might be caused by something other than a spike. I have seen clicks caused by discontinuities that look like steps rather than spikes. You might also be surprised at how audible some apparently innocent-looking glitches can be, while other more visually impressive damage turns out to be inaudible. Some more subtle ticks may not be visually obvious at the standard detail scale of one pixel per sample, so it may be necessary to zoom in further to 2 or even 4 pixels per sample to see the waveform in greater detail.

Longer regions are very tedious to scan manually. In these cases, it is usually best to start off using the automatic click detection facility. Use **Declicking | Set Declick Parameters** to pick the kind of clicks you want to find. Then run **Declicking | Find All Clicks**. If this finds more than a few clicks, this will be quite enough to be getting on with. Once the potential clicks are found, switch to **View | Standard Detail Scale** and then start stepping through the found clicks with **Declicking | Goto Next Click**. Each click can be considered individually for repair. It is probably a good idea to do an initial pass through the found clicks to get a feel for those that are significant; as you do this, take a note of the click numbers of those you wish to return to, and then do so via **Declicking | Goto Specific Click**. Actual repair of the clicks is described in the next section; at the moment we're just discussing how to find them.

If, having repaired the more significant clicks in a long quiet region, there are still a number of quieter (but still audible) ticks you wish to remove, then another run of the automatic click detection process (this time using more sensitive parameter settings) might be helpful. On the other hand, there may be just a few individual clicks that remain; in this case it is probably easiest to deal with each one separately in the manner described above for single clicks.

### Abrasions

Abrasions are much harder than clicks to identify. They are by nature more extended in time. It is really only feasible to try and repair abrasions that occur within sections without very much high frequency content. Wave Repair does not have a facility to find this kind of damage automatically; you will have to manually scan the area.

The characteristic waveform you are looking for is a section where there is a sudden burst of high frequency "mush" superimposed on a much smoother waveform. It's pretty futile to try and fix abrasions that last longer than a few hundredths of a second.

### What if You're Not Sure?

Sometimes you can hear the damage, but no matter how closely you look at the waveform you can't see it for sure. There may be a number of potential areas that look suspicious, but none of them are obvious.

In this case, to pin down where the damage is, just pick one of the possibilities, select it, and play the surrounding area, but skipping the selected region (using **File | Play Context Around Selection**). If the damage is gone (possibly being replaced with an obvious click), then you've probably found it. If instead the original damage remains, then that particular possibility wasn't the one, and you'll have to try another.

**Manual Declicking**

The bulk of the task of restoring recordings of vinyl LPs involves removing the short blemishes caused by dust and vinyl damage. What follows should not be thought of as a set of rules you must follow. They are merely a distillation of my experiences to date in using Wave Repair. You may well find other methods that you find easier to operate.

While Wave Repair does have an automatic declicking facility, it is (like all other automatic declicking packages, despite what they may claim) limited in what it can achieve. Some vinyl damage can only be repaired manually, and manual repair of most other vinyl damage gives better results than automatic repair.

There are three types of damage characteristic of vinyl LPs: (i) clicks, pops and ticks; (ii) abrasions, resulting in short bursts of "static-like" distortion; and (iii) general vinyl "hash".

Wave Repair is targeted primarily at repairing clicks, pops, ticks and very short abrasions (although these are somewhat more problematical). Since it does not provide any kind of broadband noise reduction, it cannot help with vinyl hash. A fourth kind of damage, resulting from groove-jumps, is not considered; such damage can be regarded as utterly fatal and any attempt to fix it is futile.

The task of repairing vinyl damage consists of two phases:

1. Locating Damage.
2. Repairing Damage.

**Automatic Declicking**

The automatic click detection process in Wave Repair comprises two aspects: finding clicks, and removing them.

The mechanism which finds clicks can be very helpful in locating damage that requires repair. However, before finding clicks, it is necessary to set up the detection parameters. These describe the characteristic shape and size of the clicks to be found via a number of options. The screen which allows these options to be set manually is the Click Detection/Repair Parameters dialog. Setting these parameters manually can be a daunting task owing to the number of options and the difficulty in visualising how they might relate to clicks to be found. Therefore there are two facilities which can assist you: the Declick Preview facility, and the Analyse Click facility.

The mechanism to find clicks is invoked via the **Declicking | Find All Clicks** option, which scans the currently selected region for possible clicks. It basically works as follows. The portion of the file to be declicked is scanned sequentially in chunks (typically each chunk is about 250 samples or less). Within each chunk, an average amplitude change is calculated; this gives a sort of "base level of activity". The aim is then to find amplitude changes within the chunk which exceed the average change by a certain degree. Any such excessive change is regarded as a potential click. The various parameters are definable via the **Click Detection/Repair Parameters** dialog.

**Find All Clicks** gathers the candidate clicks into a list which can be browsed later by hand using the **Goto Next Click** and/or **Goto Previous Click** options. Each such click can then be considered for manual or automatic repair. Manual repair involves either interpolation, or redrawing the waveform with the mouse, or perhaps smoothing, copying or even deletion. Automatic repair, invoked via **Declicking | Remove Current Click**, attempts to fix up a suitable area around the damage using either interpolation, replacement or smoothing (according to the repair method you choose), but in many cases it will of course provide inferior results to manual repair. In some cases, you may find that **Remove Current Click** manages a partial repair, but the tail end of the click remains.

In addition to automatic repair of individual clicks using **Remove Current Click**, the option **Declicking | Remove All Clicks** automatically fixes every click in the current list; the effect is identical to manually stepping through the list and invoking **Remove Current Click** on each one.

**Note**: If you invoke **Edit | Undo** after performing a **Remove All Clicks**, all the removals are undone; the clicks are not reinstated one at a time.

While **Remove All Clicks** can be effective in some circumstances, it should be viewed as the "lazy" option; manual repair of clicks nearly always yields better results.

Note that if the detection parameters are set aggressively (such that just about every real click is almost certain to be found), there will be a great many "phantom" clicks in the list. On the other hand, if the parameters are set so as to avoid these phantoms, a number of more subtle clicks are bound to be missed. It is best to try experimenting with a few settings to find the best compromise.

I have found through experience that it is all too easy to set aggressive parameters and find that they appear to work well on one section of music. Then, when applied to some other section they result in unpleasant clipping effects. For the very best results, the automatic declicking mechanism is probably best ignored. You are much better off simply listening to the waveform, noting the approximate times where clicks are heard, then going in and manually scanning through the relevant sections zoomed in to one pixel per sample, and with the amplitude scale maximised.

## Dealing with False Clicks

When you use the automatic click detection mechanism, it is inevitable that some pieces of waveform will be found and deemed to be possible clicks when in fact they are not. Of course, you can ignore these "phantom" false clicks while you deal with the list. However, if you do another pass (eg. using different parameters to try and find other types of click) it is quite possible that the same false clicks will be found again. It could become quite tedious to have to keep ignoring the same old set of false clicks many times.

Therefore, you can tell Wave Repair that a click it thinks it has found really isn't a click. This is done by invoking **Declicking | Set Current Click as False**. Once a click has been declared to be false in this way, it will not be included in any subsequent click detections. If you store the list of clicks in a file using **Declicking | Save Clicks in File**, the declared false clicks are also stored. Note that this ability to declare false clicks was new from release 4.3, and because of this the format of **.clk** files changed at that release. Therefore, old **.clk** files (written by release 4.2a or earlier) or not compatible with releases 4.3 and later (and vice-versa).

## Unattended Declicking

The optimum declicking parameters will nearly always vary within a long WAV file. Therefore for best results, you should declick a long WAV file in sections. This is of course a time consuming task. If you are prepared to accept less-than-ideal declicking in order to save time, the option **Declicking | Find and Remove All Clicks** will process the selected region in one go: finding the clicks and removing them according to the currently set parameters. The best way to use this option is to establish the best compromise of declicking parameters using the real-time declick preview feature.

Warning: Note that in order that this operation can run quickly, it directly amends the original WAV file on hard disk, and its effects cannot be undone. You are very strongly advised to make a backup copy of the original WAV file before proceeding in case the results are unacceptable.

Note that in an unregistered copy of Wave Repair, this operation works only on the left channel (this allows an evaluation of its effectiveness while preventing it being of any practical use).

**Repairing Damage**

Once you've located the damage, there are a number of options for fixing it: interpolation, redrawing, replacement, deletion, smoothing and muting. Having used one or more of these repair options in an attempt to repair the damage, you can listen to the results before saving the changes. If the results are unsatisfactory, you can discard them using the **Edit | Undo** option.

Note: only save the changes when you are happy with the results. Remember that in order to achieve good performance, Wave Repair writes the updates directly back over the original WAV file; once you save the changes, the original is overwritten.

<u>Interpolation</u>

If you have a simple spike type click that interrupts an otherwise smooth waveshape (eg. on the up- or down-slope of a clearly defined wave cycle), interpolation will almost certainly do the trick. Simply select the region immediately around the spike, and execute **Edit | Interpolate** or **Edit | Bezier Interpolate** for the appropriate channel. Where the damage covers only a few samples, a Bezier interpolation usually gives a good repair; if it extends over several dozen samples, a Bezier interpolation might create a large hump, and in these cases a linear interpolation is usually better.

<u>Redrawing</u>

Redrawing the waveform with the mouse is the most flexible mechanism available. In theory you can of course draw any waveform you like, but in practice it is virtually impossible to predict what a complex drawn waveform will sound like.

Redrawing tends to be useful when the damage is a little more extensive than a simple spike. Having identified the damage to be repaired, switch on **Edit | Redraw Wave Mode**, and by clicking and dragging the mouse, simply draw over the damage with the waveform you regard would have been there had the damage not been present. You will see this new waveform in red as you drag the mouse. When you release the mouse button, the redrawn waveform will replace the damage. You may find that zooming in to a finer level of detail in the time domain will make the redrawing easier. In this case, you'll find yourself switching in and out of redraw wave mode, and swapping between one pixel per sample and higher degrees of zoom. This is the reason behind many common options having single keystrokes as shortcuts: you can use the mouse in one hand to define the selected region you're interested in and to redraw waveforms, while using the other hand to press **W** (toggle redraw wave mode), **S** (show selected region) and **D** (show standard detail scale - one pixel per sample).

<u>Replacement</u>

Some damage extends for too long a time to be repaired by interpolation or redrawing. Such damage is typically a burst of distortion superimposed over a complex waveform with a lot of high frequency content. Even though it isn't a single click visually, it often sounds like one (although it may sound more like a smeared "splat" than a well-defined click). The problem with these kinds of waveforms is that it's virtually impossible to distinguish the damage from the music. You can try redrawing bits and pieces, but my experience is that you just end up with a different and equally objectionable splat. Alternatively it may be a very large click whose effects extend over a few hundred samples.

If the damage is to one channel only, and the stereo difference between the channels is small, replacement of the damaged section from the other (undamaged) channel may well produce a good repair. To try this, highlight the damaged section and use **Blocks | Copy Left to Right** or **Blocks | Copy Right to Left** as appropriate to copy over the other channel.

An option that sometimes works well on regions with lengths between a few dozen and a few thousand samples is Spectral Replacement. This analyses the blocks immediately preceding and following the region to be repaired, and computes an average of the two as the replacement. To try this, highlight the damaged section and invoke **Blocks | Spectral Replacement**.

Alternatively, you can try to replace the damaged region with a copy of some other portion of the same WAV file that looks pretty much the same. This is a little like the way that some of the old analogue de-clickers for record players worked in the 1970's: they substituted likely clicks with a copy of the immediately preceding signal, on the assumption that it should be pretty much the same.

Wave Repair does have an equivalent of this very inaccurate approach, via the **Blocks | Copy Preceding Block** option: this simply grabs the same number of samples as are currently selected from the immediately preceding part of the WAV file and copies them over. If you can see that the preceding part of the WAV file looks right, and are careful about defining the selected region, this can work well.

Wave Repair also has a more sophisticated version of this mechanism. It can scan backwards (or forwards) through the WAV file, looking for a block of samples that start and end with sample values that are close to those that start and end the selected region; the hope here is that such a "compatible" block may be suitable to replace a damaged area. Using the **Blocks | Set Compatible Block Params** dialog, you can define the extent of the WAV file to be searched for such a replacement block. It is unlikely that a suitable replacement block will be found more than 1/10th second away from the damaged area; it is also usual that searching backwards is more likely to find a suitable replacement. Repeated uses of **Blocks | Find Compatible Block** will search further afield for the next block that qualifies. The current potential block is displayed in green over the top of the selected region that it might replace; if you want to copy it over, use **Blocks | Copy Compatible Block (Left** or **Right**, accordingly). Note that this mechanism was first introduced in an early version of Wave Repair, and has largely been superceded by the more recently introduced overlay block mechanism, which is described next.

Another way to find and copy over a replacement block is via the **Overlay Block** mechanism. In this scheme, rather than having Wave Repair look for what seems a likely replacement section, the original damaged waveform is displayed as a blue overlay which can be slid around the WAV file, allowing the human operator to use his or her own insight in finding a suitable replacement block. Having determined where the damage is, you're looking for somewhere nearby that has a similar but undamaged section of waveform that will serve as a replacement. The usual sequence of events is as follows:

1. Select the section of waveform that is damaged, plus a little bit either side.
2. Invoke **Blocks | Setup Overlay Block**. The blue overlay appears.
3. Now, using the mouse, grab the blue overlay and drag it backwards or forwards in the WAV file. (If you want to go past the start or end of the currently displayed page, you can use the normal navigation and/or zoom facilities to do so). The aim is to place the overlay on top of an undamaged section of waveform that looks like it might make a good repair for the damaged section.
4. If you need to accurately position the overlay but find this difficult with the mouse, you can nudge it by one sample with the left & right arrow keys while holding down Shift+Ctrl.
5. Once you've got the overlay on top of the desired replacement, invoke **Blocks | Copy Overlay (Left** or **Right**, depending on which channel has the damage). The waveform underneath the overlay will be copied over the place where the overlay was originally set up. After copying, you can listen to the result and undo it in the normal manner if it hasn't made a good repair.

In general, the overlay block mechanism gives the best chance of success, but requires a little more patience to use.

There may be times when a suitable replacement block is found that would work well, except that it is offset vertically. In these cases, you can copy the block, then adjust its vertical position using **Blocks | Nudge (Up** or **Down**, **Left** or **Right**, depending on the channel and required direction). This moves the waveform up or down so that it fits in better with the surrounding context.

Regardless of the method used to copy a block of waveform, after the copy operation the joins at the start and end may need to be tidied up. If the option **Blocks | Smooth Edges After Copying** is on, Wave Repair will automatically add short Bezier interpolations at the two ends of the copied block. In most cases this is appropriate, but if you prefer to tidy up the joins manually, simply switch the option off.

<u>Replacement From Another File</u>

In some cases, it may be that you have a suitable replacement for a damaged section in another file. This may have been prepared with a different WAV editor, or perhaps by a re-recording. It is possible to replace a section of the current WAV file with (part of) another, provided the sampling rates of the two files are the same.

To replace a section of the WAV file, proceed as follows:

1. Position the start of the selected region at the exact sample where the replacement is to commence. You can easily check the exact sample number selected via **Position | Selection Details**.
2. Invoke **Edit | Replace From File** to choose the file containing the replacement samples. (It is possible to choose the same file in order to paste a long section, but if the sections overlap, the results will be wrong. I would recommend that another more sophisticated WAV editor should

be used to do this kind of update).
3. Once the file has been selected, you will be asked to specify the exact sample number and length of the section to be copied. Note that Wave Repair conventionally denotes the first sample number in a WAV file as 1, not 0.

<u>Deletion</u>

A very short glitch may be successfully repaired simply by removing it (ie. deleting the samples). Provided the number of samples deleted is sufficiently small, the timing inaccuracy introduced into the music's tempo will be completely inaudible.

To determine whether a deletion will make a suitable repair, select the region you think could be deleted and invoke **File | Play Context Around Selection**. This will play a little before and after the selected region, but miss out the selection itself. By comparing this to the sound of **File | Play Selection With Context**, you should be able to judge whether deleting the region will work. If it sounds right you can then go ahead and delete the selected region.

To delete the selected region, mark it using **Edit | Mark Selection for Deletion**. At this stage, the samples are not deleted, since to do so would alter the number of samples in the WAV file, which means that the entire file will have to be re-written (a lengthy process). Rather, the marked section is noted for future deletion. You can continue to mark as many other regions to be deleted as you wish, and then finally delete them all in one go using **Edit | Execute Deletions**; this means that only one complete re-write of the WAV file is necessary to effect multiple deletions.

While a section is marked for deletion, it will be omitted during any playback so you can hear the effect of making the deletion before actually carrying it out. If you are unhappy with a deletion mark, it can be removed by selecting a region which completely includes it and invoking **Edit | Clear Deletion Mark(s)**.

<u>Smoothing</u>

Where a lengthy section (by "lengthy", we're talking about a few hundredths of a second) has been damaged by an abrasion, you can try to smooth it out. Select the region to be smoothed, and execute **Edit | Smooth Abrasion**. The general shape of the waveform is retained, but the "spikiness" is reduced. This is somewhat similar to performing low-pass filtering on the selected region (although filtering is not actually the way it is achieved).

Repeated applications of **Edit | Smooth Abrasion** will make the waveform smoother and smoother. Listen at each stage to determine whether any improvement is achieved. If you execute one smoothing too many, it is simple to **Edit | Undo** the last smoothing.

Due to the nature of the smoothing algorithm, a step may appear at the start and end of the smoothed region. If this happens, you can touch it up by interpolating or redrawing the boundary.

<u>Muting</u>

It is very rare that muting (using **Edit | Silence**) makes a satisfactory repair. However, it may be that a particular piece of damage is particularly objectionable and can't be otherwise fixed; in this case, muting the damage may result in a less objectionable sound, even though it will still be audible.

**Real-time Declick Preview**

Wave Repair has a mechanism known as **Declick Preview** that looks for clicks while playing (a section of) the wavefile. The purpose is to allow the user to adjust the click detection parameters while listening to their effects. This makes automatic declicking a much more useable process than by setting parameters, running a declick/remove pass, listening to the results, and then starting over again.

Note: This real-time mechanism does not actually create amendments to the wavefile in the way that the **Declicking | Remove All Clicks** option does (although the algorithms used to detect and repair the clicks is the same); rather it monitors the samples and removes them on-the-fly in the output buffers which are being sent to the soundcard. So, although what you hear is the result of clicks having been removed, no actual updates are made; they can be made later in the normal way using the parameter settings selected during Declick Preview.

To use this function, select the range of samples you wish to declick and run **Declicking | Declick Preview**. The Declick Preview modeless dialog appears and the selected sample range will play over and over again in loop mode. While it is playing, you can adjust the parameters in the dialog box. You can also re-select a different region in the normal manner and, starting with the next pass of the playback loop, the new selected region will be played. Once you are satisfied with the parameter settings, press the **Stop** button to end Declick Preview. Since the last parameter settings you chose will have been retained, you are now in a position to run **Declicking | Find All Clicks** and then **Declicking | Remove All Clicks** to actually perform the declicking should you be happy with the results you heard while in Declick Preview. As a shortcut, the Declick Preview dialog has a button labeled **Remove** which automatically invokes **Find All Clicks** followed by **Remove All Clicks**.

Note that Declick Preview requires quite a lot of CPU power to run successfully. I have verified that a 133MHz Pentium is fast enough in most circumstances and that a 100MHz 486/DX4 is marginal. Note however that detection of "hill" shape clicks requires rather more CPU, and depending on the particular combination of parameters may not work on a P133; it certainly doesn’t work on a 486.

Some Notes on Using Declick Preview

Declick Preview is most useful when you wish to clean up a section of quiet music (eg. the fade out of a song) that has a great number of low level ticks which would be very tedious to find and repair by hand. For removal of odd clicks here and there, manual homing in, scanning and redrawing remains the best option in many cases.

Work on fairly short sections (a few seconds) while you try to find the best settings. This allows you to adjust the settings and hear their effect while you still have a good memory of what the music sounded like on the previous pass, and thus be able to judge if the change is an improvement.

Low level ticks have different characteristics to bigger clicks and pops, and it is futile to try and fix them both in one pass. Try to get rid of the more obvious clicks first, then make another pass to deal with the little ticks. If you try to find settings that do them both together, it is likely that serious distortion will be introduced. (This is because big clicks typically need large click width & relative amplitude settings, while small ticks need small width & amplitude settings: to find both in one pass needs large width and small amplitude settings, resulting in the removal of lots of low level musical peaks).

There are of course a fairly large number of parameters you can fiddle with during Declick Preview:

1. The fundamental goal is to remove clicks with the minimum "collateral damage". Once you've found settings that remove the clicks, you should try to find the largest settings for **relative amplitude** & **relative offset** and the smallest settings for **click width** & **repair width** that still get rid of the clicks.
2. If you hear clicks during Declick Preview, the first adjustment to make is to reduce the **relative amplitude** setting. Keep bringing it down until the clicks start to disappear. If, when they do, they are replaced by a roughness to the sound, try increasing the **click width** setting and bring the **relative amplitude** back up a bit. Reducing **relative offset** might also allow clicks to be removed at a higher **relative amplitude** setting. (In fact, you should keep **relative offset** fairly low at all times; leaving it permanently at zero isn't completely daft). Adjusting the **window size** is rarely useful.
3. In general the most worthwhile click shapes to look for are **spike** and **instant**. Although very big pops probably won't be found except as **hill** shapes, setting that shape will usually find too many musical peaks that aren't clicks or pops. (Basically, it is best to fix big pops manually). The **step** shape sometimes finds the odd discontinuity that is missed by **instant**, but if you do try **step**, don't set the **relative amplitude** too low or else literally thousands of completely innocuous samples will be singled out for elimination. Only try **hill** or **step** if you've exhausted the other adjustments first.
4. If your CPU can handle it, it's usually best to leave **repair with bezier curves** switched on; the cases when a straight line interpolation gives better results are extremely rare.

**Shareware or Freeware?**

The complete Wave Repair package is a special purpose WAV file editor, and is shareware. To use it as such requires that it be registered.

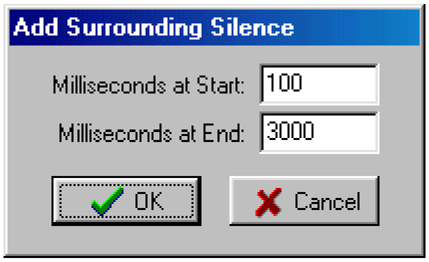
However, there are a few specific features which may be useful to some users who do not require its WAV editing capabilities, and these work properly without registration. When only these facilities are used, Wave Repair is freeware:

Recording and Playback
There are of course a great variety of packages available which can record to hard disk. The main thing which distinguishes Wave Repair is the quality of its record level metering.

Creating cue sheets in preparation for writing a CDR with packages such as CDRWin
CDRWin is a well-known CD writing package which can place track and index markers onto a CD created from a small number of large WAV files, but in order to do so it requires a cue sheet to define the positions of the tracks and indexes. Some other packages may also be able to use CDRWin-style cue sheets. Wave Repair can create such cue sheets.

Splitting Tracks in preparation for writing a CDR with packages which do not use cue sheets
Many other CD writing packages (such as Roxio's Easy CD Creator) require that each track be in a separate WAV file. Wave Repair can split large WAV files into separate files for this purpose.

## Add Surrounding Silence Dialog




When the **Edit | Add Surrounding Silence** option is invoked, this dialog appears so that you can specify the amount of silence to be added at the start and end of the selected region. The initial values are those given as defaults in the Options dialog.

**Playback**

All playback from Wave Repair concerns itself with the currently selected region. There are three basic playback options:

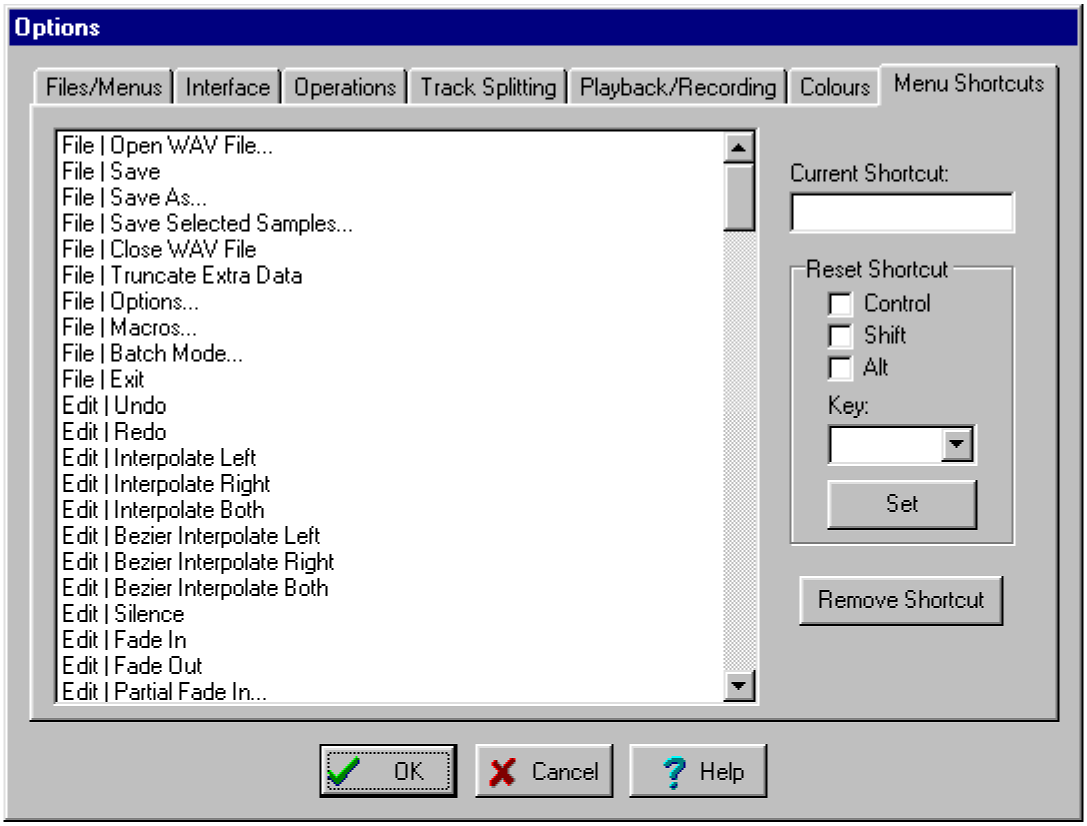
1. Play the selected region. This allows you to listen precisely to a part of the WAV file.
2. Play the selected region plus a little extra before and after. The main purpose of this is to allow you to listen to a very small selected region within its immediate context. The amount of time added at the start and end is controlled via the Options dialog.
3. Play the context around the selected region, but omit the region itself. The purpose of this is to allow you to listen to what the effect would be of removing the selected region entirely before actually doing so.

Note the following special behaviours during playback:

1. If a part of the region to be played has been marked for deletion, it is omitted from the playback.
2. The Space Bar can be pressed to place markers at interesting places which you can return to later. As the space bar is pressed, a green vertical line is placed at the point of interest. This would typically be used to mark audible damage so that it can be easily located later.
3. The Escape key can be used to pause and resume playback.
4. All options (other than **Stop Playback** and **Pause/Resume Playback**) are disabled; this is deliberate in order to prevent their accidental execution.

The other form of playback is when a preview option is in effect (ie. Declick Preview, Amplify/Compress Preview, Equalise Preview, or Filter Preview). This repeatedly plays the selected region while applying the appropriate parameters for the preview in progress. This allows you to adjust the parameters while you listen to their effect.

**Options Dialog: Menu Shortcuts Tab**




On the left is a list of the available menu items. **Current Shortcut** shows the currently assigned shortcut for the selected menu item in the list.

If you wish to change a shortcut, select the desired key from the drop-down list, check the **Control** and/or **Shift** boxes as required, and click the **Set** button. You should then see the new shortcut appear in the **Current Shortcut** box. If you wish to delete a shortcut (so that the menu item no longer has a shortcut), click the **Remove Shortcut** button.

Wave Repair checks whether you place the same shortcut on more than one menu item. If you do, then it warns you and gives you the option of backing out the update, but you are able to keep clashing shortcuts if you wish (since this may be the most convenient way to rearrange many shortcuts). If you do get into the position where your menu shortcuts have become completely messed up, you can return to the default shortcuts by following these instructions.

**Creating CDRWin-style Cue Sheets**

To create a cue sheet, special markers known as **cue points** must be placed at the appropriate positions in the WAV file. There are three types of cue point you can place, but before these are described it is necessary to explain how track and index marks work on an audio CD.

Each track begins with a track mark, which can either be index 0 or index 1. Consider the behaviour of the display on an audio CD player as it plays a CD. As the player moves from one track to the next, there are two types of behaviour:

1. The time counter continues to count up after the previous track ends, and as the next track begins the track number goes up and the timer begins again from 0:00. This behaviour happens when the track begins with index 1.

2. After the previous track ends, the track number goes up, but the time display begins counting backwards. When it reaches zero, the next track begins and the time starts counting up again from 0:00. This behaviour happens when the track begins with index 0: as index 0 is reached, the track number is incremented and the display starts counting down; when the countdown reaches zero index 1 is reached and the display starts counting up again.

On some CDs, while playing a track, some CD players show index marks. The track number remains the same, and the time display continues to count as normal, but a secondary number (the current index within the track) increments. Many modern CD players do not support access to index marks, and few CDs manufactured these days have them. They were originally intended as ways to locate specific sections within a musical piece. For example, a four movement symphony might have four tracks to mark the movements, and within each movement the sections (such as main theme, development, recapitulation) might be marked with indexes. Although they have largely fallen into disuse, you might wish to place them onto your CDRs, and so Wave Repair supports them.

Three types of cue point can now be described:

**Track, index point 1**
This marks the beginning of a new track which starts at index 1. If you do not require a countdown before the track, place one of these markers.

**Track, index point 0**
If you require a countdown before a track, place one of these markers at the position where the countdown is to begin.

**Index**
If you have placed a **Track, index point 0** type cue point, then you must also place a separate Index cue point at the position where the countdown is to end and the track itself starts. You can also place extra **Index** cue points within a track if you wish to add indexes.

To add a cue point, click the left mouse button at the place where the track or index point is required, and then execute Cue Points | Add Cue Point. Note that you should not attempt to place a cue point at the start of the WAV file to specify the first track; Wave Repair assumes that track

1, index 1 begins at the start of the WAV file.

There are some subtle issues concerning the exact positions where cue points are best placed. If you wish to consider these, they are discussed here.

Once all the required cue points have been placed, a CDRWin-style cue sheet can be created by executing **Cue Points | Write Cue Sheet**. If you need to include several WAV files on the CDR, the next WAV file can be loaded into Wave Repair, the cue points for that file placed, and then added to the same cue sheet by choosing the **Append** option. You can write a cue sheet even if there are no cue points set (in this case a single track mark at the start of the WAV file is written to the cue sheet). This allows a single cue sheet to be built for multiple WAV files which contain one track per file by appending to the same cue sheet for every file.

For further details see help about the Cue Points Menu.

**Exactly Where Should the Cue Points be Placed?**

This section is intended to help you decide precisely where cue points should be placed.

Firstly, note that you can zoom in as tightly as you wish in order to place the cue points as accurately as you like. When you do this, you may notice that a cue point is placed at a slightly different position to where you clicked the left mouse button. This is because track and index marks on CDs must be placed on block boundaries; each block on a CD is 1/75th second (588 samples). Wave Repair ensures that the cue points are placed at block boundaries.

When you play a commercial CD, you will no doubt have noticed that the track number tends to increment at exactly the point where the music for the track begins. This would imply that you should place the cue points to start tracks right where the track starts. When a CD is played through normally, this would be perfectly adequate, but the possibility that you may choose to use a CD player's track search buttons invokes two subtle issues:

1. When a CD player moves to a new track it mutes its output while it finds the track. Some CD players have a slight delay in un-muting their output once the track is found. This means that if the cue point is placed hard up against the start of the track, there is a danger that the initial transient of the track will be lost while the CD player un-mutes. As a general rule it is advisable to place the cue point about 200 milliseconds before the start of the track to avoid this possibility.

2. Also, when the CD player un-mutes to start playing the new track, if the samples at the cue point are not close to zero then there is the danger that a "click" will be heard as the new track begins. For this reason it is advisable to place the cue point at a place where the sound is very close to silence.

So, in summary, the cue points at the start of tracks should be placed at a position approximately 200 milliseconds before the start of the track, and at a point where the samples are close to zero.

**Other Effects Menu**

Normalise (With Undo)
Normalises the selected region so that it peaks at a given maximum volume. Also includes an option to remove any DC offset that may be present. When invoked, the Normalise dialog appears so that you can select the level and any other options you require. This particular menu item performs the normalisation as an update stored in memory or a temporary file so that its effect can be removed with the Undo option. The decision whether to use memory of a temporary file depends on the **Where Edits are Stored** setting in the Options dialog. Note that if you select the **Always in Virtual Memory** setting, and normalise a very long section, then this would consume a vast amount of memory. If your computer has the resources to do this, then it will work but may run extremely slowly.

Normalise (Direct Overwrite)
Normalises the selected region so that it peaks at a given maximum volume, optionally removing a DC offset as well. This menu item performs the normalisation by directly overwriting the WAV file on disk. This has the advantage that it runs faster than Normalise (With Undo), but the disadvantage is that it cannot be backed out with the Undo option. You should be confident that you wish to normalise before using this menu item.

Amplify/Compress (With Undo)
Amplifies the overall volume and/or compresses the dynamic range of the selected region using the parameters set up by the most recent execution of Amplify/Compress Preview. This menu item performs the amplification and compression as an update stored in memory or temporary file so that its effect can be backed out with the Undo option. As with Normalise (With Undo), this can consume a vast amount of memory and run extremely slowly.

Amplify/Compress (Direct Overwrite)
Amplifies the overall volume and/or compresses the dynamic range of the selected region using the parameters set up by the most recent execution of Amplify/Compress Preview. This menu item performs the compression by directly overwriting the WAV file on disk. It runs faster than Amplify/Compress (With Undo), but it cannot be backed out with the Undo option. Be sure you are happy with the parameters set up in Amplify/Compress Preview before using this menu item.

Amplify/Compress Preview
Repeatedly plays the selected region through the soundcard whilst applying volume amplification and/or dynamic range compression. The Amplify/Compress Preview dialog allows you to adjust the parameters while you listen to their effect. When you stop the playback the parameter settings are retained for use in a subsequent Amplify/Compress (With Undo) or Amplify/Compress (Direct Overwrite).

Equalise (With Undo)
Equalises the selected region using the parameters set up by the most recent execution of Equalise Preview. This menu item performs the equalisation as an update stored in memory so that its effect can be backed out with the Undo option. As with Normalise (With Undo), this can consume a vast amount of memory and run extremely slowly. Equalisation may be run as a one

or two pass operation. If run in one pass, there is a danger that clipping distortion could be introduced, especially if excessive bass boost has been applied. By using two passes, no clipping is possible (the overall amplitude is reduced accordingly to compensate), but it does take twice as long to run. The choice between one and two passes is set in the Options dialog.

Equalise (Direct Overwrite)
Equalises the selected region using the parameters set up by the most recent execution of Equalise Preview. This menu item performs the equalisation by directly overwriting the WAV file on disk. It runs faster than Equalise (With Undo), but it cannot be backed out with the Undo option. Be sure you are happy with the parameters set up in Equalise Preview before using this menu item.

Equalise Preview
Repeatedly plays the selected region via the selected soundcard whilst applying equalisation. The Equalise Preview dialog allows you to adjust the equalisation parameters while you listen to their effect. When you stop the playback the parameter settings are retained for use in a subsequent Equalise (With Undo) or Equalise (Direct Overwrite).

Filter (With Undo)
Filters the selected region using the parameters set up by the most recent execution of Filter Preview. This menu item performs the filtering as an update stored in memory so that its effect can be backed out with the Undo option. As with Normalise (With Undo), this can consume a vast amount of memory and run extremely slowly.

Filter (Direct Overwrite)
Filters the selected region using the parameters set up by the most recent execution of Filter Preview. This menu item performs the filtering by directly overwriting the WAV file on disk. It runs faster than Filter (With Undo), but it cannot be backed out with the Undo option. Be sure you are happy with the parameters set up in Filter Preview before using this menu item.

Filter Preview
Repeatedly plays the selected region via the selected soundcard whilst applying filters The Filter Preview dialog allows you to adjust the filters while you listen to their effect. When you stop the playback the parameter settings are retained for use in a subsequent Filter (With Undo) or Filter (Direct Overwrite).

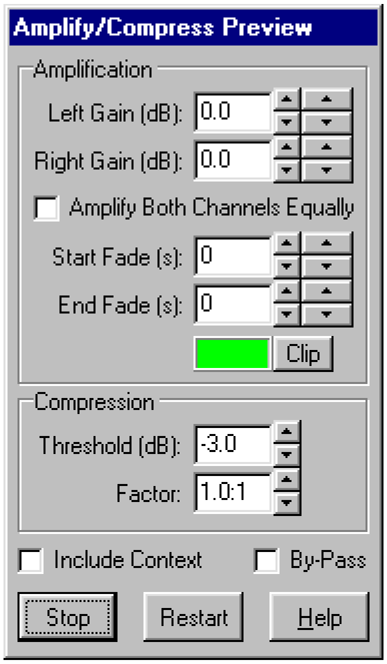
Channel Mix (With Undo)
Mixes the channels in the selected region using the parameters set up by the most recent execution of Channel Mix Preview. This menu item performs the mixing as an update stored in memory so that its effect can be backed out with the Undo option. As with Normalise (With Undo), this can consume a vast amount of memory and run extremely slowly.

Channel Mix (Direct Overwrite)
Mixes the channels in the selected region using the parameters set up by the most recent execution of Channel Mix Preview. This menu item performs the mixing by directly overwriting the WAV file on disk. It runs faster than Channel Mix (With Undo), but it cannot be backed out

with the Undo option. Be sure you are happy with the parameters set up in Channel Mix Preview before using this menu item.

Channel Mix Preview
Repeatedly plays the selected region via the selected soundcard whilst mixing the channels The Channel Mix Preview dialog allows you to adjust the mixing settings while you listen to their effect. When you stop the playback the parameter settings are retained for use in a subsequent Channel Mix (With Undo) or Channel Mix (Direct Overwrite).

**Amplify/Compress Preview Dialog**




This is a modeless dialog which appears while **Other Effects | Amplify/Compress Preview** is in operation. It allows the amplification and compression parameters to be adjusted while the selected samples are playing. When Amplify/Compress Preview ends, the values which have been set here are retained for use when amplification and compression is actually applied.

Amplification is completely straightforward, and allows the volume of the waveform to be adjusted up or down by a maximum of 50dB either way. **Gain** defines the amount of amplification to be applied; a gain of 0.0dB means no amplification is applied. The two channels can be adjusted independently of one another; alternatively if **Amplify Both Channels Equally** is on, then adjusting either channel will make the same adjustment to the other. Note that an excessive increase in volume can cause the waveform to be clipped, in which case the green "LED" turns red; pressing the **Clip** button resets the "LED" to green. **Start Fade** and **End Fade** specify the time in seconds during which the amplification effect is "faded in" at the start and "faded out" at the end. This allows you to bring the amplification in and out gradually rather than suddenly.

Compression works by squashing the dynamic range of the loudest parts of the music, which allows the entire waveform to be amplified so as to increase the perceived volume. **Threshold** defines the point where the squashing will begin. For example, if it is set to -9.0dB, then only those parts of the music which are louder than 9dB below full volume are squashed. Adjusting the threshold below -9dB is rarely sensible, although Wave Repair will allow it to be taken as low as -12dB. **Factor** is the amount of squash that will be applied; a factor of 1.0 means no compression is applied. Using a factor above about 3:1 is rather extreme, and is rarely useful. The lower the threshold and/or the larger the factor, the more extreme the compression. The particular compression mechanism used by Wave Repair maintains the peak volume, by ensuring that the amount of amplification after squashing exactly matches the amount of squashing applied. This also ensures that it is impossible to introduce clipping through the use of compression alone. Note that extreme amounts of compression, while not causing any clipping, can still distort the waveform and cause unpleasant harshness to the sound.

If you wish to adjust the peak volume without compressing dynamic range, set the compression factor to 1.0:1 and adjust only the amplification. Conversely, to compress dynamic range without changing the peak volume, set the amplification gain to 0.0dB and adjust only the compression parameters.

If **Include Context** is checked, an additional amount before and after the selected region is included in the playback, but is not included in the amplification and compression. This allows you to listen to how the transition points will sound.

If **By-Pass** is checked, the amplification and compression is switched off. This makes it easy to do a before-and-after comparison of the current settings.

The **Stop** button ends Amplify/Compress Preview, which can also be ended via the **File | Kill (Stop) Output** menu option or by pressing the **Stop Playback** button on the toolbar.

The **Restart** button interrupts the current pass through, and begins playback again at the start.

**Hum & Rumble Removal**

If you find that there is some hum which gets recorded along with the music, you should try to eliminate it at the source. Hum is generally caused by a difference between the ground (earth) potentials of the stereo system and computer. Arranging a common ground for these two devices can often help to eliminate hum.

If all attempts to eliminate hum fail, you might find Wave Repair's filtering feature useful, using its notch filters. Hum is usually present at 60Hz and 120Hz (where the mains frequency is 60Hz, eg. in the USA), or 50Hz and 100Hz (where the mains frequency is 50Hz, eg. in the UK). Aim to use the steepest possible notch filters that successfully remove the hum. The steeper the filter, the less impact it will have on the bass content of the music.

Another low frequency problem you may encounter is turntable rumble. This is a general "grumbling" noise that poorer quality turntables may generate, usually due to less than perfect main bearings. A suitable High Pass filter might be able to reduce the effects of rumble without affecting the music too badly. Note that the preset Rumble filter uses a frequency (35Hz) which may well be inappropriate for some turntables: you should experiment with the frequency in order to find the one which matches the rumble characteristics of your own turntable. As with the hum filters, it is best to use as steep a filter as possible to avoid affecting the music higher up in frequency.

**Normalisation & Compression**

These two processes are used to adjust the final volume of the music.

Normalisation is a fairly simple procedure, whereby the WAV file is analysed, and then an amplification factor is applied so that the loudest peak(s) are set to a defined maximum. Often you will choose that maximum to be the largest possible (ie. peaking at 100%, or 0dB), but the maximum level can be reduced slightly if you have a special reason for doing so. If you have managed to record the WAV file so that the peak record level was close to 0dB, then normalisation will have little effect; it is more relevant if the WAV file was recorded at a low level.

It may not always be appropriate to normalise a recording. For example, if the recording is of a quiet instrument, such as flute, then the listener would not expect it to be particularly loud on playback.

There are some disadvantages to normalisation. Firstly, any noise (such as hiss or vinyl surface noise) is amplified along with the music. Secondly, there is a very minor reduction in resolution due to rounding errors in the arithmetic, although when dealing with recordings of analogue sources this effect is likely to be unnoticeable.

Normalisation is a straightforward process: select the region you wish to normalise (usually an entire track or WAV file), and invoke **Other Effects | Normalise** (refer to the description of the Other Effects Menu for an explanation of the differences between the two types of Normalise menu options).

An option during normalisation is to remove any DC offset. A DC offset is where the waveform is not centred exactly around zero, and is caused by some minor inaccuracy in the analogue circuitry of the playback system or soundcard during recording. Most analogue circuitry has some degree of DC offset, but unless it is severe it can usually be ignored.

Compression reduces the dynamic range of the louder parts of the music so that the overall volume can be increased. The more compressed music becomes, the higher its perceived volume (even though its peak volume is no higher than before it was compressed). Compressed music tends to sound more "punchy", so it is particularly suitable for rock and pop music. Ironically, many people perceive compressed music to have greater dynamics, when in fact the opposite is the case. If you wish to make a faithful restoration of vinyl LPs, then my advice is not to apply any compression, although of course the choice is entirely yours. You might, for example, wish to make your CDR transfers of LPs sound as loud as commercial CD releases, which in the rock & pop field tend to be very highly compressed. The main situation where compression is useful is in balancing the perceived volume of a variety of tracks from different sources which you are putting together into a compilation. This can apply to digital as well as analogue sources.

Intuitively, you might expect to apply compression first and then normalisation, but in fact it is easier to do it the other way round. Once the WAV file is normalised, compression can be applied in one step without the need to worry about whether further normalisation is necessary. This is because Wave Repair's compression operates in a manner which maintains the peak volume regardless of the amount of compression. Were you to apply compression first, you would not know how much more volume would be added by a subsequent normalisation, thus complicating the whole process. Compression should be used with discretion; it is a rather clumsy tool which can make music sound initially impressive while actually robbing it of the genuine dynamics it may have originally possessed.

Excessive compression will certainly make hiss and vinyl surface noise much more noticeable. After compression you may find that tiny clicks and ticks which were previously unimportant or even inaudible become quite noticeable, so a further declick pass may be necessary. As with normalisation, rounding errors in the arithmetic will cause a very slight reduction in resolution, but this will be swamped by the much grosser effects of the compression itself.

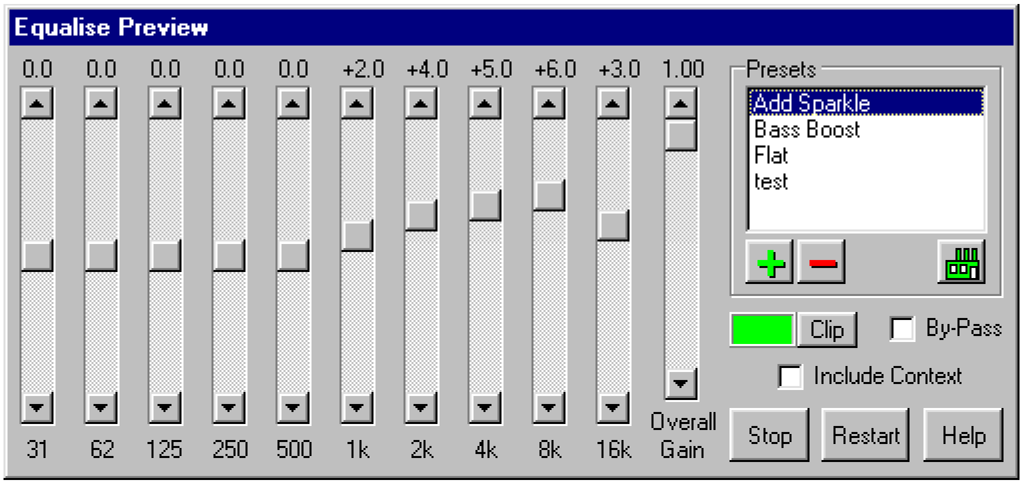
Before committing any compression to your WAV files, you should experiment with **Other Effects | Amplify/Compress Preview** to establish what settings are appropriate. Once you have found the required settings, invoke **Other Effects | Amplify/Compress** to actually perform the compression. Again, see the Other Effects Menu description for an explanation of the two Compress options.

Choosing the right degree of compression can only be done with experience; there is no rule you can follow which will work well at all times, since the effects of any particular compression settings depend greatly of the characteristics of the music to start with. However, if you are trying to balance the perceived volume of a compilation of tracks, the following procedure may help:

1. Normalise all the tracks independently of one another.
2. Choose the loudest track as a reference; this track will remain uncompressed.
3. For every other track, use Amplify/Compress Preview to discover the settings which match its perceived volume to the reference track chosen in step 2, and then compress the track with those settings.

The easiest way to compare the reference track with the one you're experimenting with is to have the tracks in separate WAV files, and to have two copies of Wave Repair running (one with the reference track loaded; the other with the track to be compressed) so that you can easily switch back and forth to listen to their relative volumes.

## Equalise Preview Dialog




This is a modeless dialog which appears while **Other Effects | Equalise Preview** is in operation. It allows the equalisation parameters to be adjusted while the selected samples are playing. When Equalise Preview ends, the values which have been set here are retained for use when equalisation is actually applied.

On the left are ten sliders which operate like a typical graphic equaliser found on some home stereo systems. Below each slider is the center frequency (in Hz) of the boost/cut band, and above each slider is the current boost/cut being applied (in dB) to that band.

Next along is a slider labeled **Overall Gain**, which adjusts the amplification which is applied to the equalised result. When applying equalisation to a selection which is already fairly loud, it is quite possible that the tonal adjustments will cause some samples to be clipped, in which case the green "LED" turns red; pressing the **Clip** button resets the "LED" to green. To control this, the overall gain can be adjusted during the preview. Then, if a single pass equalisation is subsequently invoked, this overall gain will be applied. You should bear in mind that adjusting the gain so there is no clipping in the selection previewed is no guarantee that the same equalisation settings applied to a different selection will not cause clipping. The way to make sure that equalisation cannot introduce clipping is to use the two-pass method; in this case the overall gain is ignored, since the operation guarantees to avoid clipping in any case.

Within the **Presets** area, a list of named settings are shown which can be selected. New presets can be added by clicking the button with the green plus symbol. An existing preset can be removed by clicking the button with the red minus symbol. The button with the picture of a factory reinstalls the default presets that are shipped with Wave Repair.

If **Include Context** is checked, an additional amount before and after the selected region is included in the playback, but is not included in the equalisation. This allows you to listen to how the transition points will sound.

If **By-Pass** is checked, the equalisation is switched off. This makes it easy to do a simple before-and-after comparison of the current settings.

The **Stop** button ends Equalise Preview, which can also be ended via the **File | Kill (Stop) Output** menu option or by pressing the **Stop Playback** button on the toolbar.

The **Restart** button interrupts the current pass through, and begins playback again at the start.

**Equalisation**

This process adjusts the tonal balance of the music. Wave Repair provides a mechanism that operates very much like the graphic equalisers that are found in some home stereo systems, with ten frequency bands offering up to 12dB of boost or cut. The procedure to follow is to use Equalise Preview to listen as you adjust the equalisation settings, and once you are happy with them apply the equalisation over the desired region by invoking **Other Effects | Equalise (With Undo)** or **Other Effects | Equalise (Direct Overwrite)**.

When restoring vinyl LPs the most likely requirement for equalisation will be to add a bit of extra "sparkle" at the top end. There are two common reasons why vinyl LPs can sound a little dull:

1. A small amount of wear is inevitable every time a record is played. This wear tends to remove high frequencies more than others.
2. Some records (especially cheap reissues) are made using "high mileage" stampers that are frankly beginning to wear out; again, it is the high frequencies that wear out first.

It is usually best to perform equalisation before declicking. The reasoning behind this is that equalisation may change how audible some clicks are; therefore if you equalise after declicking, it may be necessary to do another declick pass afterwards.

**Dropouts and/or Stuttering During Recording and/or Playback**

There are a host of possible reasons for anomalies during recording and playback. Most of them are system configuration issues, and they are discussed here in the order that you should investigate them, not necessarily because the earlier ones are more likely to be the cause, but because they are easier to check and/or correct. (It is also worth noting that similar configuration issues could be the cause of buffer underruns during CDR writing). Remember that after making any Windows system configuration changes it may be necessary to reboot the PC before they take effect.

Before considering the various possibilities, a note about playback of edits in Wave Repair. If you find that playback "stutters" only when playing back a section which includes edits that have been made and not yet saved, this is a good indication that your hard disk is not quite fast enough to be used in the default manner that Wave Repair works. Ensuring that your hard disk is running in DMA mode (if it supports it) will help. If you still have problems, then you should instruct Wave Repair to store edits differently, telling it to put them in virtual memory instead of in a temporary file. Use the Options Dialog to set this.

OK, now on to a discussion of the various system configuration issues which may cause dropouts:

Compressed Hard Disks

There is nothing that can be said about this except: don't try to use compressed hard disk partitions for digital audio.

Rogue Video Card Behaviour

Some video cards, especially those which perform Windows acceleration, obtain a small performance boost by not bothering to check if the bus is free before attempting to use it. The result of this is that, if the bus isn't free, then the video card tries to use it and blocks the bus, resulting in audio samples getting lost.

A temporary solution might result from turning down the level of graphics acceleration (in Win95/98, via **My Computer | Properties | Performance | Graphics**). If you experience dropouts during recording, you could try switching off the record level meters and/or the elapsed time counter, which will reduce the amount of video activity.

The proper solution is to contact the video card manufacturer and find out whether there is a way to configure its driver so that it always checks that the bus is free before using it.

FindFast and Other Background Tasks

Programs lurking in the background which might spring into life at inappropriate moments are to be avoided. Things such as screen savers, email servers and the like should be switched off. One particular program of this type is **FindFast**; if you have installed a recent version of Microsoft

Office it is probably on your system. The problem with FindFast is that you never really know it's there unless you take the trouble to check what tasks are running. If you discover that it is running, kill it, and also remove it from the Programs Menu Startup group (where it is likely to have placed itself).

<u>File System Read-Ahead</u>

By default the Windows file system reads ahead, on the assumption that the running program is soon likely to want the data that follows. This is all very well for a typical PC performing a general mixture of "normal" computer activity. It is, however, the last thing you need while you're recording audio. Try reducing the amount of file system read-ahead (via **My Computer | Properties | Performance | File System**).

<u>WaveRepair Buffering</u>

By default Wave Repair uses quadruple buffering for recording and playback, each buffer being 1/10th second long. This gives a total buffering of 0.4 seconds. This should be enough on most systems, but you may find that increasing the buffering can eliminate dropouts. Another popular WAV editor, CoolEdit, has a much more generous default of 8 buffers each 1/2 second long (ie. a total buffer space of 4 seconds). You can, if you feel the need, set Wave Repair up (via **File | Options | Playback/Recording | Configure Soundcard**) to use this much buffering. It costs nothing but a little memory, and most modern systems have plenty to spare.

<u>System vcache Settings</u>

The Windows virtual cache is quite often the culprit when occasional (rather than constant) dropouts are experienced. That said, it is probably better to eliminate other possibilities first, since to adjust the vcache requires that you edit the SYSTEM.INI file rather than just adjust some settings.

If the vcache is allowed to change in size, it'll be just your luck that it does so in the middle of an important recording, with a consequent loss of samples. The trick is to set the vcache up with the same settings for its minimum and maximum size, thereby stopping Windows ever resizing it. To do this, you need to edit the file **SYSTEM.INI** in the Windows system directory (usually **C:\WINDOWS**). Open the file with any text editor (eg. Notepad) and find the section headed **[vcache]**. Add two lines beneath, so that the section looks like this:

```
[vcache]
MinFileCache=4096
MaxFileCache=4096
```

If there is no [vcache] section, add one at the bottom of the file. If the existing [vcache] section already has settings for MinFileCache and MaxFileCache, adjust them accordingly. The actual value to use (4096 in the example above) is the size in kilobytes of the cache. 4096 (ie. 4MB) is appropriate for a PC with about 32MB of main RAM. Generally about 1/8th of the main RAM is a suitable amount to use, but if you have more than 64MB of main RAM, it probably isn't worth setting your vcache above 8192.

IRQ Conflicts

If your soundcard attempts to use the same IRQ as another device, anything can happen. If all your devices are plug-and-play, it is easy to check that there are no conflicts by going through each device under **My Computer | Properties | Device Manager**. If you have any non-plug-and-play cards, they must be set up manually to avoid conflicts. Moreover, you must also instruct the BIOS to prevent allowing the IRQs used by those non-plug-and-play cards to be assigned to plug-and-play cards. To do this, it is necessary to enter the BIOS setup during PC bootup, find the place where IRQ assignments are made, and set those IRQs being used by non-plug-and-play cards to the appropriate value (often called something like "ISA", or "N/A", or "Legacy").

Soundcard Drivers

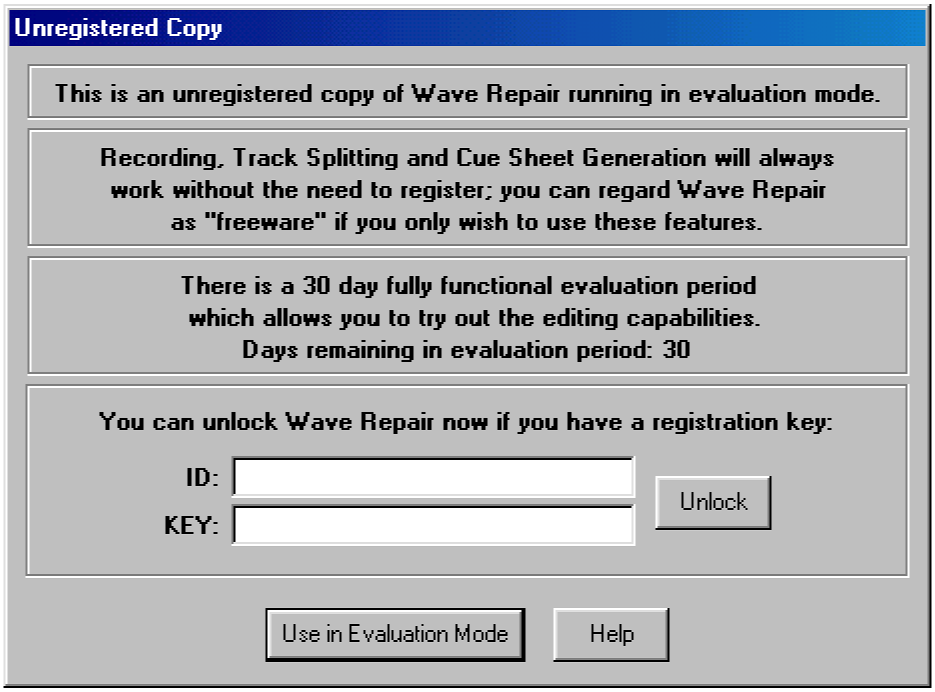
Although unlikely, it is possible that the drivers for your soundcard have some kind of incompatibility with the timing of your main system board. Modern PCs are less likely to have this problem; it was more an issue with old ISA bus 386 and 486 machines. Nevertheless, you should make sure you have the latest drivers for your soundcard.

Disk Fragmentation

In these days of Ultra-ATA and SCSI hard disks, fragmentation is rarely an issue. Modern hard disks are so fast that the data rates required by simple stereo audio is no problem unless there is an extreme degree of fragmentation. However, occasionally defragmenting your hard drive can't hurt, so it's worth doing it now and again.

## Unlocking

When you start up an unregistered copy of Wave Repair, the following screen will appear:




(If this dialog does not appear, Wave Repair is already unlocked).

If you wish to use Wave Repair in "freeware mode" (using only the recording, track splitting and cue sheet generation features), then you can ignore this screen. Simply press the **Use in Evaluation Mode** button to proceed. Use of the "freeware mode" means you will have to put up with this screen appearing every time you start up Wave Repair.

If you have registered, you will have been sent an unlock key by email. To install your unlock key, enter the ID and KEY which were sent to you into the appropriate boxes and press the **Unlock** button.

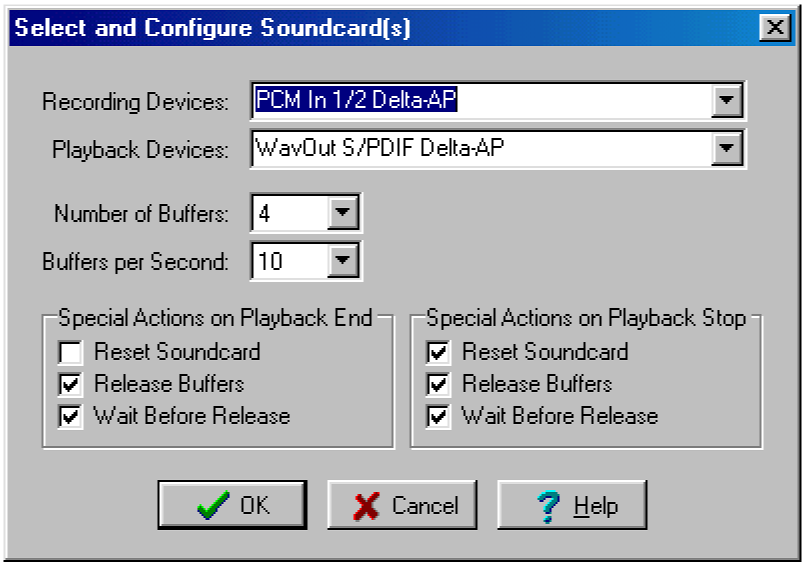
<u>Note that the ID is case sensitive and never contains any spaces, so be sure to enter it using upper and lower case letters exactly as supplied</u>.

A "Successfully Registered" message should appear. If it does not, and you are sure you have entered both items correctly, please email me to report this. (Since it is possible that my email address may need to change, please check on the Wave Repair web page at **www.waverepair.com** for the current email address).

If you press the **Use in Evaluation Mode** button, Wave Repair will be run in a special mode. During the initial 30 day evaluation period, the program is fully functional. After this period has expired, it will not be possible to save edits back to disk.

**Configuring the Soundcard(s) to be Used**

When the **Configure Soundcard** button is pressed on the Playback/Recording tab of the Options dialog, the following sub-dialog appears:




The **Recording Devices** and **Playback Devices** drop-down lists allow you to pick the desired soundcard and recording & playback lines to use. Clearly, this option is only of interest to users whose computers have more than one soundcard, or where the installed soundcard provides more than one line for playing/recording WAV files. There is no compulsion to use the same soundcard for both playback and record. The installed cards with the necessary capabilities (ie. only those capable of 16bit stereo 44.1/48kHz operation) are listed in the drop down boxes.

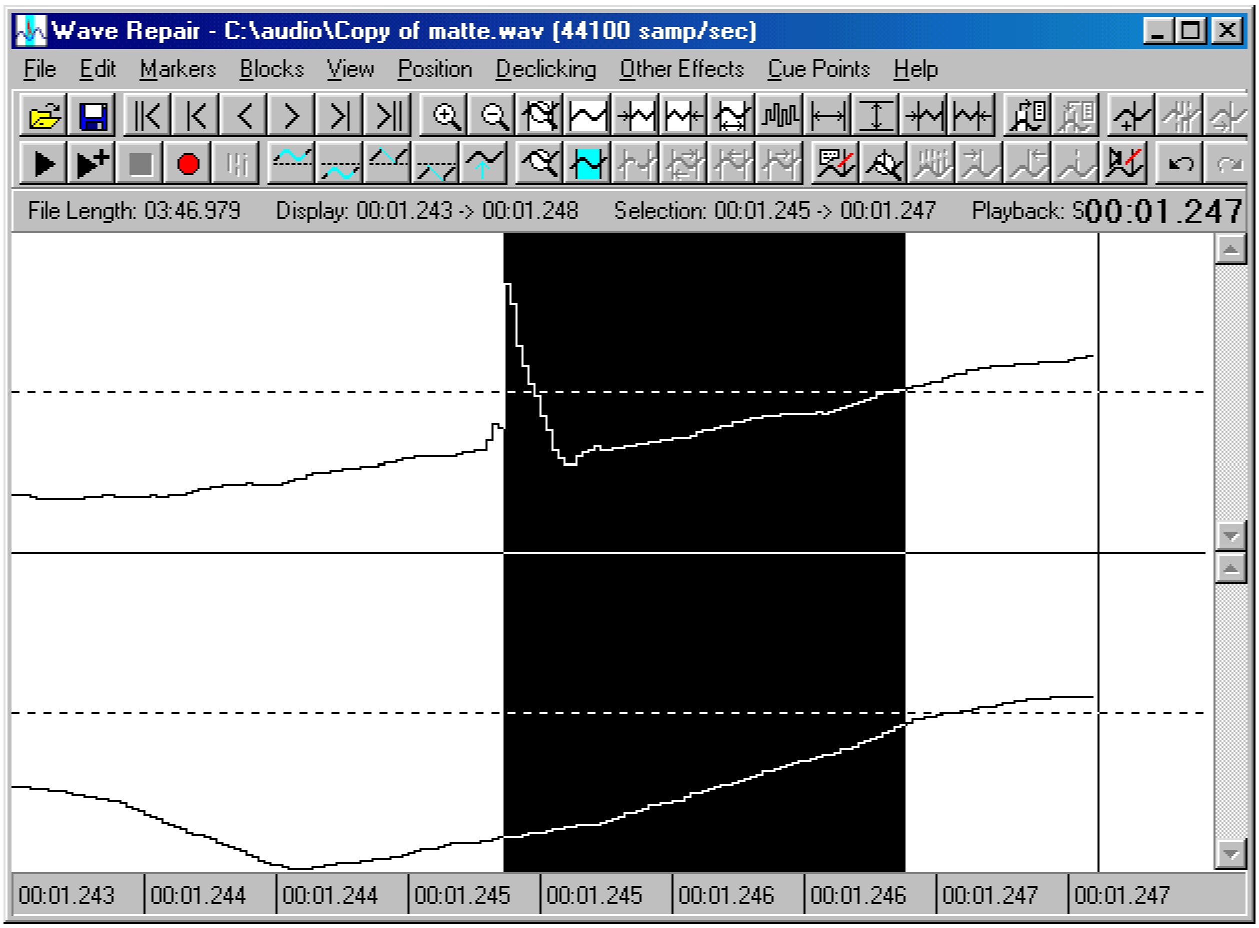
The **Number of Buffers** and **Buffers per Second** options allow you to alter the size and number of buffers used during playback and recording. **Number of Buffers** specifies how many buffers are used in a cyclic queue. Using only 2 buffers is unlikely to work unless you have a very fast computer; using more than 8 buffers is not likely to be necessary. **Buffers per Second** determines the size of each buffer; the larger this number the smaller the buffers. For example, if you select 2 buffers per second, then each buffer will be half a second long. Larger buffers use more memory, although on modern PCs this will not be an issue (even if you select 16 buffers at 1 buffer per second, less than 3MB of total buffer space is used). The default values are suitable for most systems; only experiment if you experience dropouts or "stuttering" during recording or playback. Increasing the number of buffers and/or decreasing the buffers per second may help to eliminate dropouts. If you are having problems with dropouts, also consider these other issues.

The two groups of checkboxes marked **Special Actions** allow you to set special behaviour to be executed when playback finishes. These options are necessary because the Windows drivers for some soundcards behave in uncommon ways. The default settings are suitable for most soundcards; you should only change these settings if you experience error messages, crashes or hang-ups when playback ends. It is not possible to suggest appropriate settings for particular soundcards; all you can do is experiment to see if you can find settings which work for your soundcard. Note that the **Wait Before Release** option is ignored if **Release Buffers** is not checked. **Special Actions on Playback End** controls what happens when playback is allowed to end normally; **Special Actions on Playback Stop** controls what happens when playback is terminated by executing **File | Kill (Stop) Output** or pressing the **Stop** button

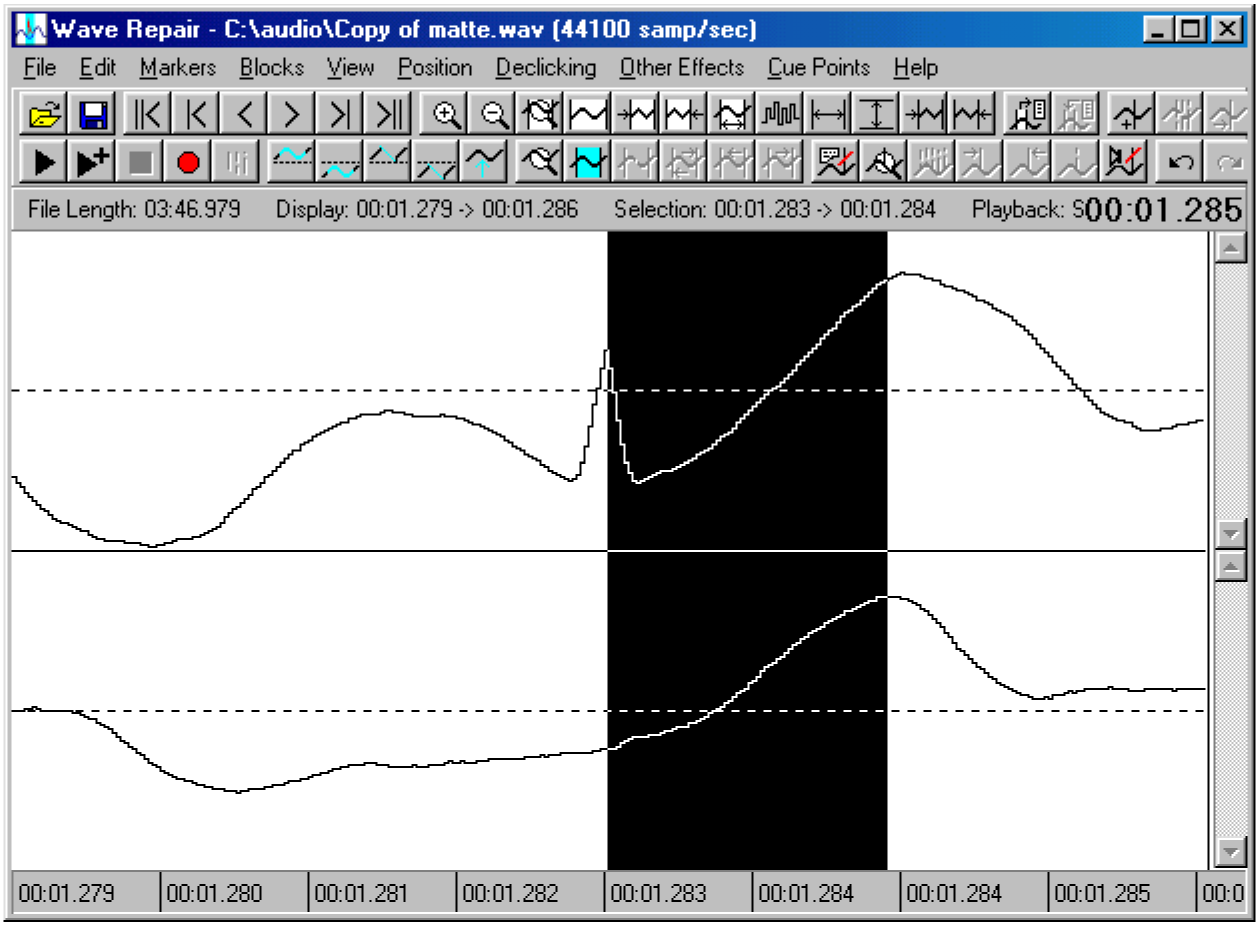
**Analyse Click**

This facility is used to automatically set up declicking parameters based on an analysis of a specified portion of the waveform.

Before invoking it, you should manually find a click whose characteristics you wish to analyse. For a discussion of how to find the click, see Locating Damage. It is important that you choose a click on its own, where there are no other clicks within 4000 samples (about 1/10th second) on either side. Once a suitable click is identified, zoom in on it so that you can see the individual samples clearly. Now, set the start of the selected region to be right at the most significant sample of the click. For example, on a click which shoots straight up (or down), set the start of selection to the sample where the sudden change occurs:




On the other hand, where a click is more symmetrical, place the start of selection in the middle of the click:




It doesn't matter where the end of selection is placed; this is ignored during the analysis. Once the click has been located and the start of selection placed accordingly, invoke **Declicking | Analyse Click (Left)** (if the click is on the left channel) or **Declicking | Analyse Click (Right)** (when the click is on the right channel). The waveform immediately surrounding the start of selection is then scanned, looking for the least aggressive click detection parameters which detect the specified click, but which do not detect any other clicks either side of it for 4000 samples in each direction. (This is why it is important to choose an isolated click).

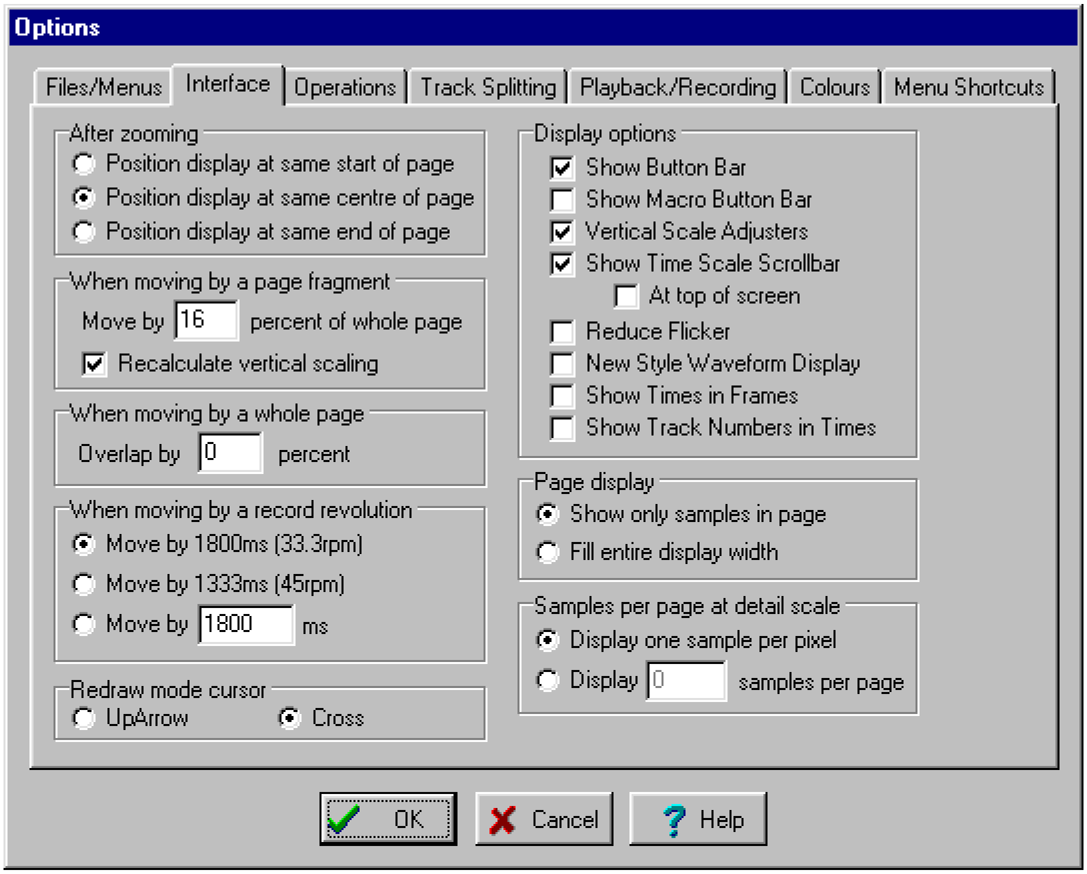
Note that the analysis runs in two distinct phases. First, it looks for **Instant Rise**, **Spike**, and **Step** click shapes; these are both more common and quicker to analyse. Only if no suitable parameters are found for one of these click shapes does it then consider looking for **Hill** and **Blip** click shapes. In this case, the user is asked whether they wish to continue analysing for **Hill** and **Blip** click shapes. Since these click shapes are rarely useful and take very much longer to analyse, you may often decide not to continue.

The click detection parameters are left set to the values found by the analysis. Note that it is not possible to guarantee that a suitable set of parameters will be found in every case; some types of click may not be able to be detected by Wave Repair's detection algorithm.

Once a set of parameters is found, you can redefine the selected region in order to search for clicks using those parameters. If you wish to save the parameters for later use, they can be stored as a preset in the normal manner via the Click Detection/Repair Parameters dialog.

Please note that only the detection parameters are set up by this facility. The click repair width is not set, and you should set this manually afterwards. A good way to discover a suitable setting for the click repair width is to invoke Declick Preview on the region surrounding the analysed click and adjust only the repair width parameter as you listen. Choose the smallest repair width which removes the click without audible artifacts.

**Options Dialog: Interface Tab**




**After zooming** determines what part of the WAV file is shown after the display is zoomed in or out. When the **same start of page** option is chosen, the page shown will start at the same place as the page that was shown prior to the zoom. When the **same centre of page** option is chosen, the centre of the page shown will be the same place as the centre of the page that was previously shown. When the **same end of page** option is chosen, the page shown will end at the same place as the page that was shown prior to the zoom.

**When moving by a page fragment** determines by what proportion of an entire page the display moves when the **Next Fragment** or **Previous Fragment** navigation options are used. This is expressed as a percentage of the whole page. In the example above, 16% represents about one sixth of the page; in other words **Next Fragment** would need to be invoked about six times to move on to the next page. When moving by a fragment of a page, it might be appropriate that the vertical scaling not be recalculated (in order to ease the task of visually correlating the new display with the previous); in this case you can switch off the **Recalculate vertical scaling** option. (Note that this is only of relevance when the **Maximise Amplitude** option is switched on).

**When moving by a whole page** determines what percentage of the previously displayed page will be included in the next page. For example, a non-zero overlap would be appropriate if you wish to visually correlate the last part of one page with the start of the next.

**When moving by a record revolution** allows you to specify a time (in milliseconds) by which the display will be changed. The idea here is that a scratch across a record is likely to result in multiple areas of damage separated by the time it takes the record to rotate. The time specified here is used by the **Next Revolution** and **Previous Revolution** options. (78 rpm is not included as a pre-determined time for space reasons, but should you wish to set the time for 78 rpm, it is 769 ms).

**Redraw mode cursor** determines the type of mouse pointer used when in redraw waveform mode.

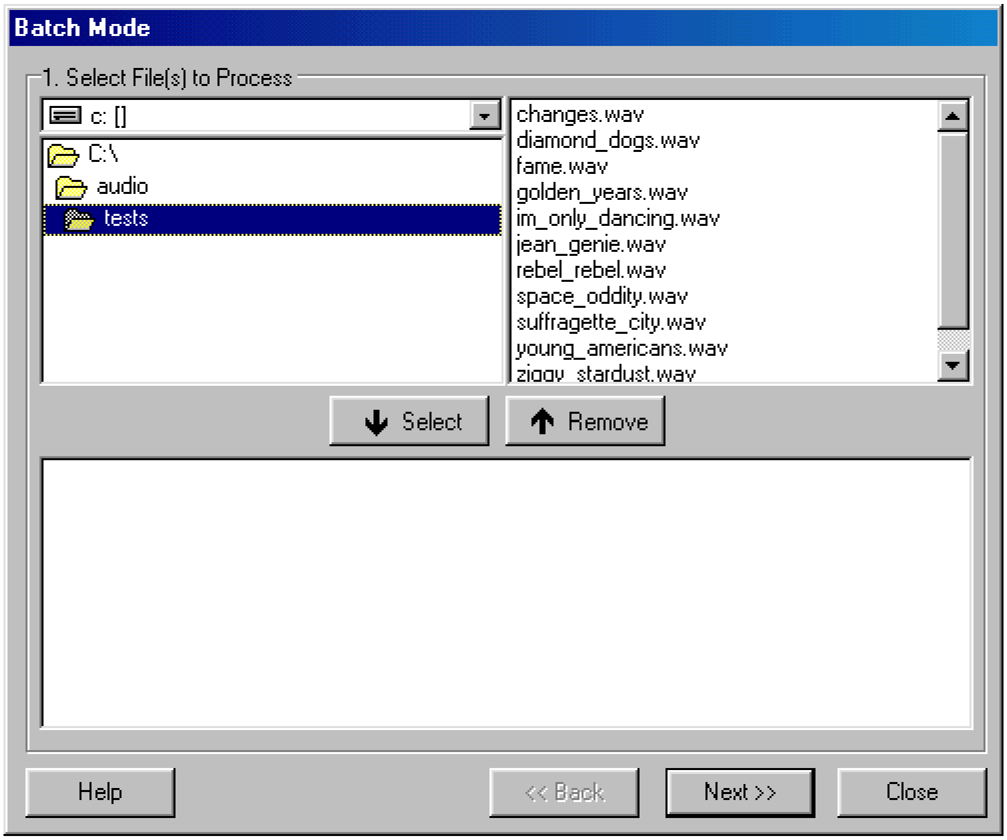
**Display options** controls a number of miscellaneous features. **Show Button Bar** determines whether the toolbar of iconic buttons is shown across the top of the display. **Show Macro Button Bar** determines whether an additional toolbar, which provides buttons to invoke all 24 available macros, is shown. **Vertical Scale Adjusters** are a pair of scroll bars down the right hand side of the display that allow you to manually alter the vertical scaling and offset of the displayed waveform; these scroll bars only operate when **Maximise Amplitude** is switched off. **Time Scale Scrollbar** is a scroll bar along the bottom of the display (or along the top of the display if **At top of screen** is on) which can be used to navigate through the WAV file using the mouse. **Reduce Flicker** controls a display option that causes less flicker when the display is repainted. (I have noticed that this option can cause a problem for some video cards, so it is switched off by default). If **New Style Waveform Display** is selected, the displayed waveform will have a more "solid" appearance at low levels of zoom. If **Show Times in Frames** is selected, all times will be given in the form **min:sec.frameno** (where there are 75 frames per second) rather than **min:sec.millisecs**. If **Show Track Numbers in Times** is selected, all times will be given in the form **trackno+min:sec.millisecs**.

**Page Display** selects how the available pixels across the main window are used. In order that waveform redrawing can be very accurate at any degree of zoom, it is necessary to use an exact whole number of samples per pixel (or pixels per sample at high zoom levels). If **Show only samples in page** is selected, only that part of the main window that is needed to display the current page is used, which can leave some blank space at the right hand edge. If **Fill entire display width** is selected, then extra samples at the right hand end of the page will be displayed to fill the main window.

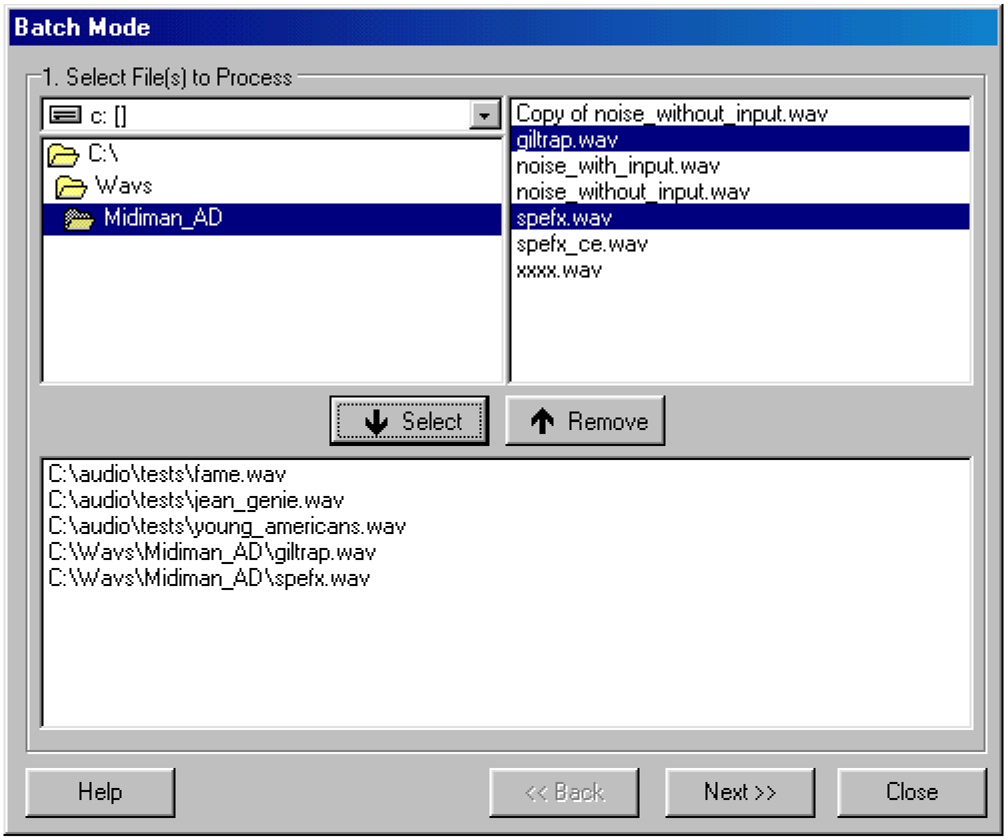
**Samples per page at detail scale** defines what zoom factor will be used when **Standard Detail Scale** is used. Setting this to **one sample per pixel** is appropriate in most cases, as this gives a degree of zoom that makes most types of damage fairly easy to spot.

**Batch Mode**

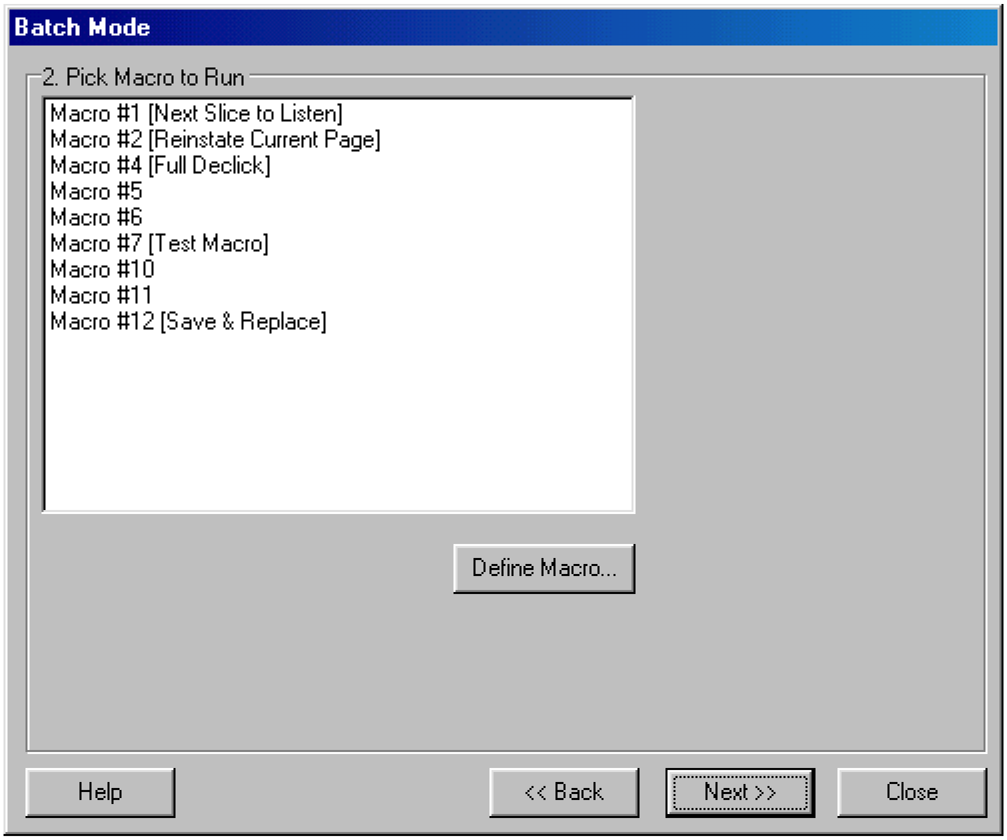
Wave Repair provides a simple batch mode facility to run a given sequence of operations over many files. Some programs describe this as a "scripting" facility. To make use of batch operation, invoke the **File | Batch Mode** option, which brings up this dialog:




This is the first stage in a "wizard" style interface, in which you select the file(s) you wish to process. To select the required files, use the two panels at the top left to navigate around your hard disk(s). Whenever a directory is selected, the WAV files it contains are listed in the panel on the top right. You can select the files you require in that panel, and press the **Select** button to add them to the list of selected files in the panel at the bottom. Multiple files can be selected at once through the normal Windows use of the Control and Shift keys. After having selected file(s) from one directory, you can navigate to another in order to add more files, eg:



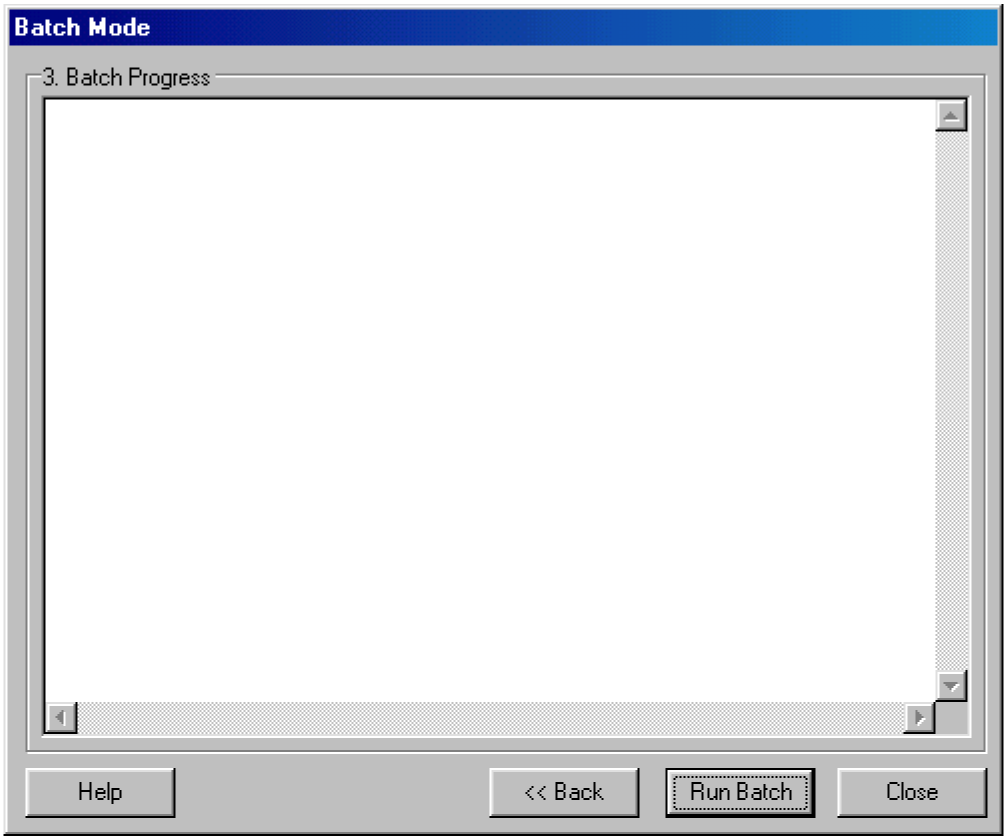

Once all the required files have been selected, pressing the **Next** button moves on to the next stage; picking a macro to be run against all the files:



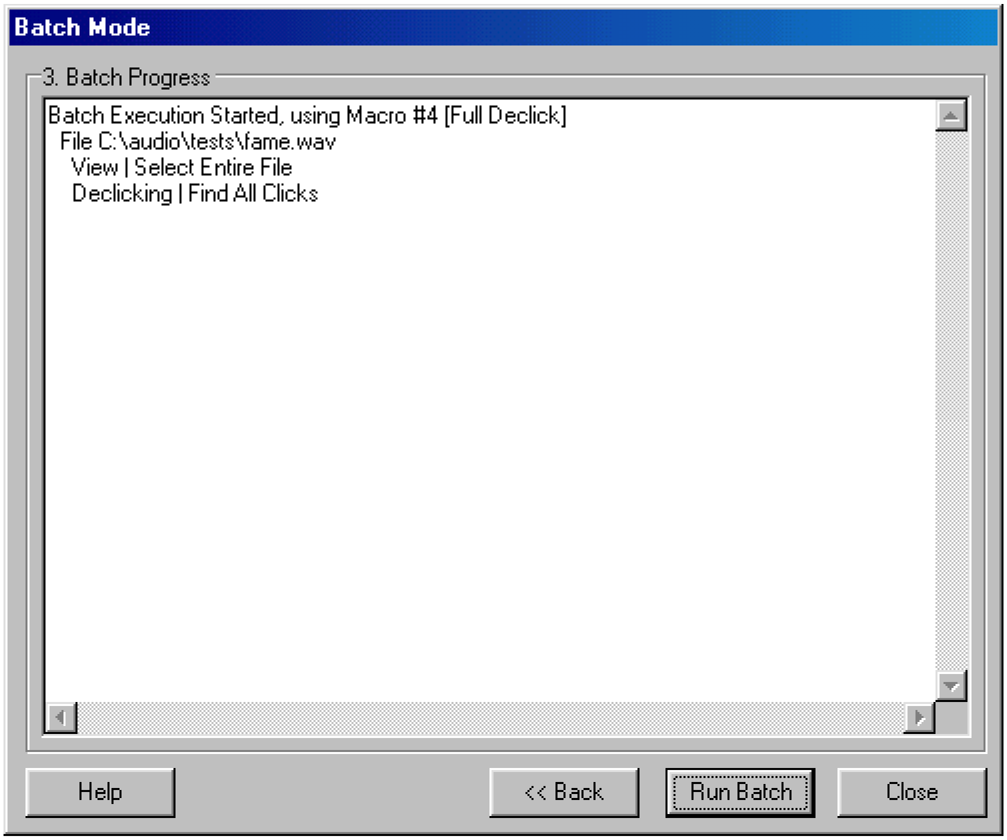

All currently defined macros are listed. Where a macro has been given a name, then that is included in the listing. The **Define Macro** button invokes the Macro Setup dialog as a convenience in case you wish to define a new macro for this batch operation. Pick the macro you wish to run over each file, then press the **Next** button to proceed to the final stage of the wizard.

Before describing the final stage of the wizard, a word about the seemingly restrictive nature of what can be run is in order. At first sight, it would seem that being restricted to just a single macro call is very limiting. However, if you think about it, you can set up a macro to do just about anything you want. If you need to perform an especially complex series of operations, just set up a macro accordingly to be used in batch mode. Remember that the last operation in a macro to be used in batch mode will probably need to be **File | Save** in order to save whatever updates have been made. An exception to this would be where the macro invokes only operations that do direct overwrites (for example you might set up a macro to run **Other Effects | Normalise (Direct Overwrite)** and then set up a batch to normalise a group of WAV files).

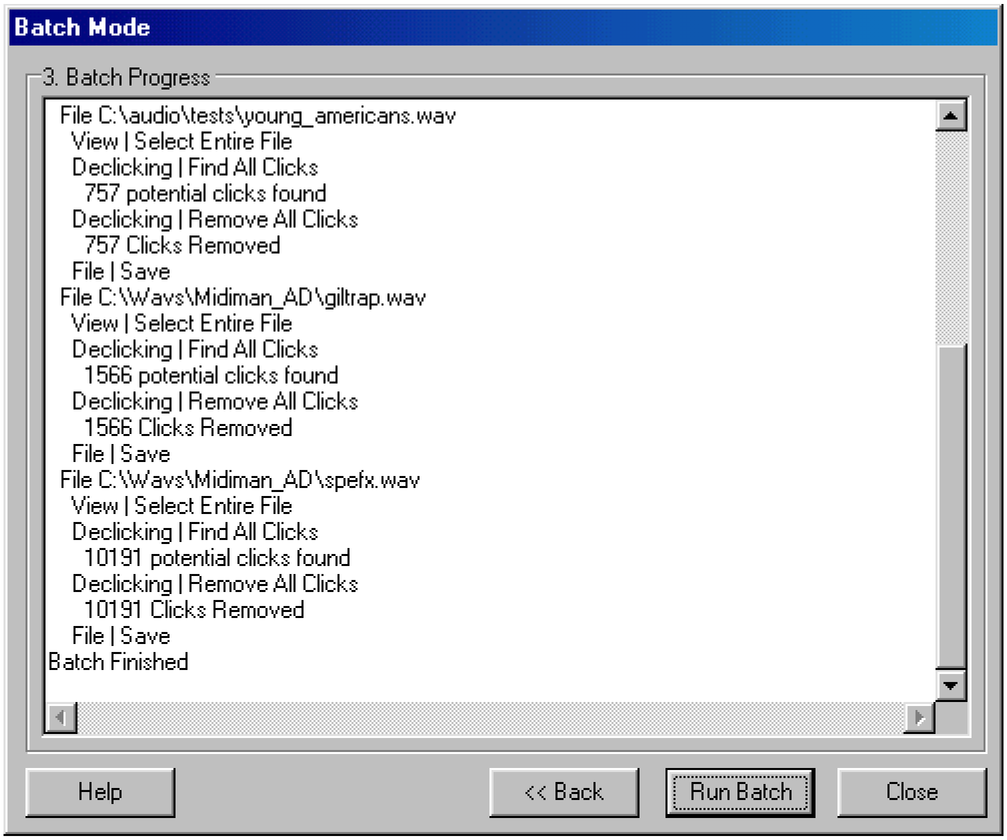
Back to the wizard. After pressing the **Next** button in stage 2, the final screen appears:




This screen is waiting to run the batch. Press the **Run Batch** button to start it. As the batch runs, a progress commentary is displayed in the panel:

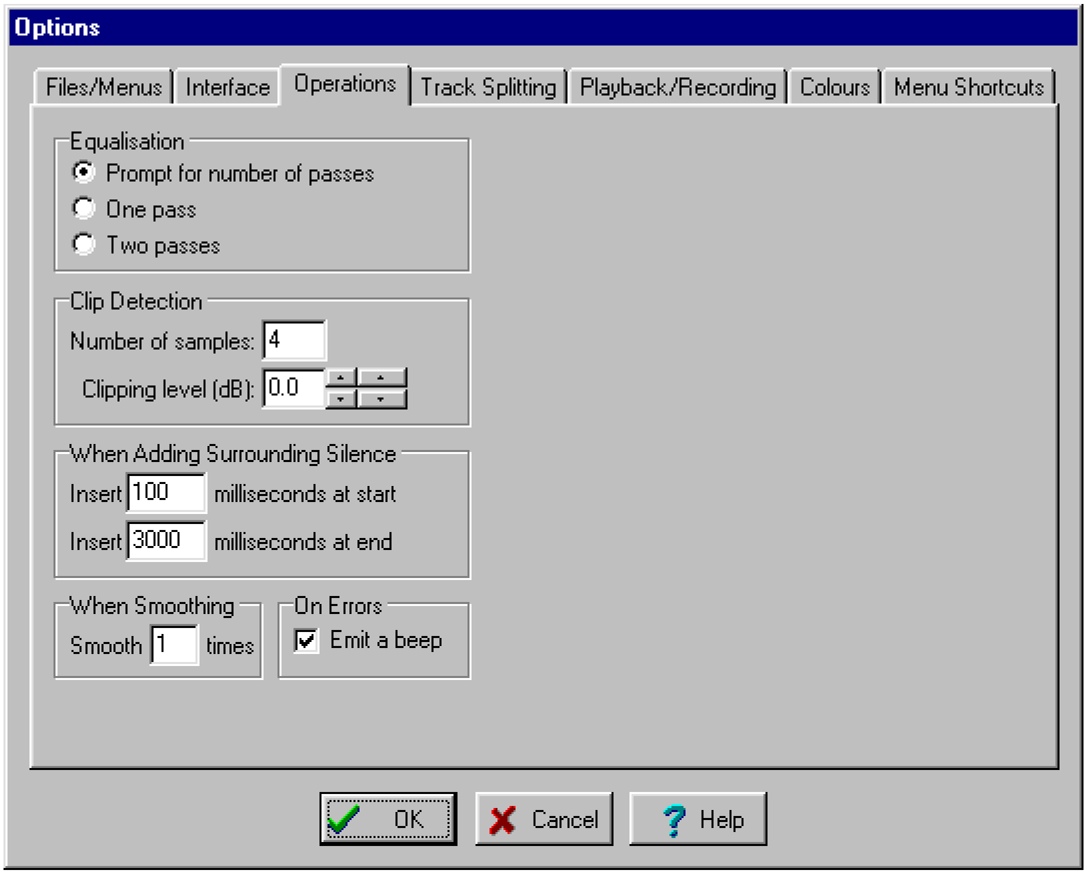



If the processing will take a long time, you can just walk away and leave it running; Wave Repair ensures that during a batch run there no dialogs or prompts that require user input. A message in the commentary indicates when the batch is finished:




At this stage, you would normally dismiss the wizard with the **Close** button. However, it is perfectly reasonable to use the **Back** button to select a different macro and/or a different set of WAV files to process, and then to come back to this final screen to run another batch.

**Options Dialog: Operations Tab**




**Equalisation** determines if any equalisation operations will be done in one pass (which takes less time but involves the danger of introducing clipping, especially if bass boost is applied), two passes (which takes longer but ensures no clipping can be introduced), or whether the user will be asked to choose the number of passes.

**Clip detection** is used by the clipping detection facilities (ie. during recording and when scanning for clipping).

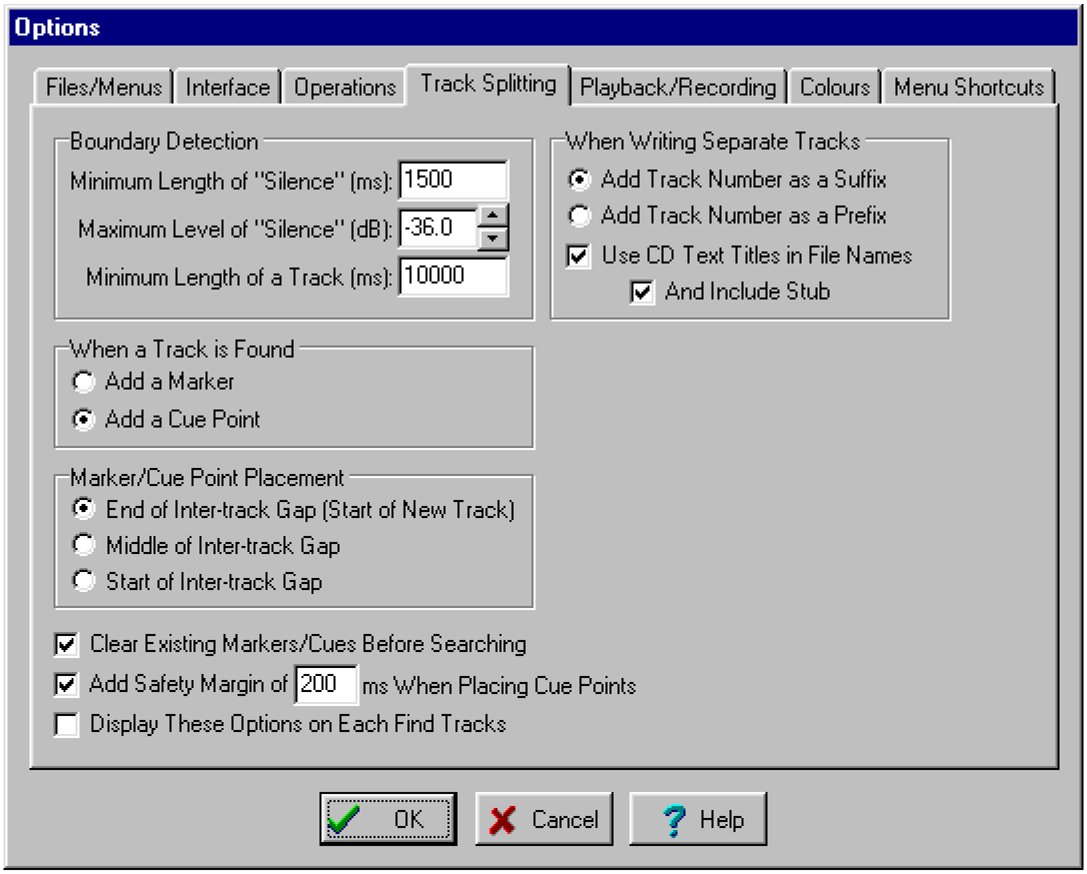
**Clipping level** is the peak level (in dB below full scale) at which clipping is deemed to start. In most cases **Clipping level** should be set to 0.0 (ie. clipping starts at full scale). Setting a lower level is appropriate if your soundcard has an analogue saturation problem such that it cannot deliver full scale samples. **Number of samples** is the number of consecutive samples at or above the **Clipping level** which must occur in order for clipping to be considered to have happened. Since the odd single sample at clipping level is unlikely to be genuine clipping, it is advisable to set this to a value a little greater than 1. If your soundcard can deliver full scale samples, then a value for **Number of samples** of 4 or 5 is appropriate; if you set **Clipping level** a little lower, then a larger value for **Number of samples** (eg. 10) may be appropriate.

**When adding surrounding silence** defines the length of silence that is inserted before and after the selected region by the **Add Surrounding Silence** option.

**When Smoothing** defines the number times the basic smoothing operation is applied in one go. The default of 1 is a fairly subtle smoothing; the larger the number, the more drastic the effect.

**Emit a beep** determines whether a Windows alert is output when an error occurs. Switching this off will prevent the loud "clang" that is emitted though the soundcard. Note, however, that Wave Repair cannot prevent the alert which Windows outputs whenever a warning is given from a standard dialog (eg. a File Save dialog). The way to prevent all possible loud "clangs" is to switch off the Windows system sounds, in which case the Windows alert will be a quiet beep through the PC speaker.

**Options Dialog: Track Splitting Tab**




This set of options controls how the automatic track splitting facility (invoked via the **Find Tracks** option) works.

**Boundary Detection** gives the characteristics that should be considered to indicate the start of a new track. A new track is considered to start if there is a quiet section that lasts for at least **Minimum Length of "Silence"** milliseconds and does not exceed **Minimum Level of "Silence"** during that period. Of course, no one setting can be right in all cases; there will always be times when a new track is missed, or a quiet section of music is mistaken as a new track. In general, a time period of about two seconds is an appropriate value for **Minimum Length of "Silence"**. If the WAV file being scanned was recorded from an LP, then a suitable value for **Minimum Level of "Silence"** would probably be somewhere between -25 and -45 dB, depending on the degree of surface noise present. For recordings from digital sources such as CDs, a lower level would be appropriate (eg. -50dB). **Minimum Length of a Track** defines the shortest period that will be regarded as a track. The purpose of this is to avoid the situation where many very short tracks are found within the gap between tracks due to the surface noise level fluctuating around the defined "silence" level.

**When a Track is Found** specifies whether a simple marker or a cue point should be added. Either way, you can review the positions where the track boundaries have been placed and change them if necessary.

**Marker/Cue Point Placement** determines where within the detected inter-track gap the marker or cue point is placed. Note that nearly all commercial CDs place their track start points right up where the music of the new track begins, so as a general rule the **End of Inter-track Gap** choice makes the most sense.

If **Clear existing markers before searching** is on, then all markers or cue points within the selected region will be deleted before **Find Tracks** runs. The removal of previous markers may help to reduce any confusion between them and any that **Find Tracks** adds.

If **Add Safety Margin When Placing Cue Points** is on, then whenever the start of a track is about to be marked with a cue point, a safety margin of the given number of milliseconds will be incorporated. This applies also if markers are being changed to cue points with the **Convert Markers to Cue Points** option.

If **Display These Options on Each Find Tracks** is on, then every time **Cue Points | Find Tracks** is invoked, these options will be shown so they can be changed if necessary. The reason for this is that it is quite often necessary to experiment with the track splitting settings in order to find the right values, and in this case it is much easier to have the settings presented directly rather than having to separately invoke **File | Options** every time you wish to try new settings.

**When Writing Separate Tracks** controls the filenames generated when the tracks are written to individual WAV files. These options can be useful when preparing playlists or MP3 files. The track number is added either at the end of the filename (when **Add Track Number as Suffix** is selected) or the start of the filename (when **Add Track Number as Prefix** is selected). If **Use CD Text Titles in File Names** is selected, then the name of the output file will be the title from the CD Text information that has been set up (if any; otherwise the name reverts to what it would have been had this option not been selected). If **And Include Stub** is selected, then the name of the output file will incorporate the stub name.

For example, if the stub name chosen is **fred**, then the output file names will be **fred01**, **fred02**, etc if **Add Track Number as Suffix** is selected, otherwise the output file names will be **01fred**, **02fred**, etc. However, if **Use CD Text Titles in File Names** is also selected, then if a track has its CD Text title set, that name will be used instead. For example, if the second track has a CD Text title of **longsong**, then the output file name will be **longsong** instead of **fred02**, or if **And Include Stub** is selected, then the output file name will be **fred_longsong**.

**Saving Recordings to Hard Disk**

A question that often gets asked is, "once I've recorded something using Wave Repair, how do I save it to hard disk, because the **Save** option isn't enabled?". The answer to this is, "you don't need to; the recording is already on hard disk in the file whose name you gave before you started the recording".

The reason for this confusion is probably because most other audio editors that can record do so by recording to some temporary space in an internal format. It's only when you save this that the file on hard disk is created where you specify. Wave Repair doesn't work that way: it writes the recorded samples directly to your chosen file as it records. When the recording finishes, all it has to do is add the necessary header at the front of the file.

**Track Splitting Confusion**

In order to split tracks, it is necessary to place special marks called Cue Points. These cue points define the boundaries between tracks. A more detailed discussion of placing cue points can be found here.

There is also sometimes some confusion concerning whether to place Markers or Cue Points in order to split tracks. The simple answer is that you must use cue points, not markers. See here for a more detailed discussion of the difference between the two.

The final area of confusion surrounding track splitting is that, once cue points have been placed, some users expect to perform the actual track splitting using the Save option, and find to their surprise that it is not enabled. The reason for this expectation is probably because another well-known track splitting program (CD Wave) behaves in this way. Wave Repair doesn't: in order to perform the actual track splitting, it is necessary to invoke **Cue Points | Split Tracks**.

You may notice that this option is enabled even if you haven't placed any cue points. This doesn't mean that it will automatically split tracks. Rather, it is enabled to allow you to pad a file out to an exact number of CD blocks should you wish to do so. If you do want Wave Repair to try and split the tracks automatically, the option **Cue Points | Find Tracks** will search the file and place markers or cue points (depending on options set in the Options dialog) at the places it thinks are likely to be track boundaries.

## Markers and Cue Points

Some users have become confused over the distinction between markers and cue points.

**Markers** (shown as vertical green lines) are nothing more than interesting locations within the WAV file that you might like to revisit. The green line is a visual reminder of where those interesting locations are. You can mark a location for whatever reason you like: hold down the Control key and left-click the mouse at the position where you want the marker. In addition, during playback or recording, you can hit the space bar to place a marker at the current playback/record position.

Some of Wave Repair's functions also place markers. When it detects possible clipping during recording or amplification (or if you ask it to scan a file for clipping), then it places markers at the clipping points. If you ask it to find possible track boundaries, then it places markers to show where it thinks tracks start (unless you have asked it to place cue points instead). When Wave Repair places markers, you are then free to use or ignore them in the same way as you would if you had placed them yourself.

**Cue Points** (shown as vertical lines with a track or index label at the top) are concerned with the interpretation of a WAV file as CD tracks and/or indexes. They are used only when preparing to write a CD, either for the creation of a cue sheet file, or for track splitting. Moreover, if you wish to create a cue sheet or split tracks, you must use cue points; markers do not figure in these operations. Note that whereas markers can be placed absolutely anywhere in a WAV file, cue points can only ever be at CD block boundaries (each CD block is 588 samples long). This is why when you place a cue point at a specific position, it may actually get created slightly later in the file, so that it is on a block boundary.

### The Save Option

The confusion surrounding the **Save** option (on the **File** menu) concerns the fact that it does not always become enabled after some process has been performed that will require something to be written back to the hard disk in order for it not to be lost. Some users have asked whether this is because somehow Wave Repair isn't properly unlocked, but this is not the reason.

Wave Repair has three distinct kinds of operations that require saving:

1. Changes to the actual audio samples in the WAV file, where the number and position of the samples is not changed. Examples of this type of edit are interpolations, fades, mutes, block copies, declicking, etc. For edits of this type, the new sample values are stored in memory, ready to be written back out to the WAV file on hard disk. Since the number and position of the samples has not changed, these edits can be saved very quickly, by simply overwriting just those samples which are affected. It is edits of this type which cause the **Save** option to become enabled. (These edits are also the ones which the **Undo** and **Redo** options operate on).

2 Changes to the actual number of samples in the WAV file. The operations which would result in this are deleting a block or inserting some silence. When the number of samples is changed in this manner, it is necessary to rewrite the entire WAV file. This is a lengthy operation which we would wish to avoid doing repeatedly. Therefore, in order to delete some samples the user first marks the region to be deleted (with **Edit | Mark Selection for Deletion**). Similarly, to insert some extra silence the user places "silence insertion marks" (with **Edit | Add Silence Insertion Mark** or **Edit | Add Surrounding Silence**). Many such deletions and insertions can be specified, and then all can be acted upon in one overall rewrite of the WAV file. One might argue that it would make sense that any outstanding deletions and insertions should be performed whenever the **Save** option is invoked, but many users have a habit of making a few edits and once they are happy with them invoking **Save,** then carrying on with further editing. It would be extremely irritating if they were to forget they had marked something and have to wait while the entire file is rewritten. It is for this reason that the **Save** option does not perform deletions and insertions. So, after marking selections for deletion and/or adding silence insertion marks, the **Save** option is not enabled. To perform the deletions and insertions, it is necessary to invoke **Edit | Execute Deletions/Insertions**.

3 Changes to items other than the actual audio samples. Examples of this are the placement of markers and cue points, the detection of clicks, etc. All these operations concern the creation of extra descriptive data about the audio samples rather than edits to the samples themselves. It would be possible to store this extra information at the end of the WAV file using whatever private format a program chooses. However, a deliberate decision was taken that Wave Repair would not do this, since the WAV files that it writes will almost certainly need to be read by other programs, and it is impossible to predict how other programs might react to such proprietary information. Therefore, these extra types of data are stored in other files, usually (but not necessarily) in the same folder and with the same name stub as the WAV file. To store these files requires the execution of the appropriate options (such as **Markers | Save Markers in File**, **Declicking | Save Clicks in File**, and **Cue Points | Write Cue Sheet**). So,

the **Save** option has no relevance to data of this type.

**Command Line Parameters**

Wave Repair would usually be run without any parameters, in which case it will start up with an initially empty main window. However, it is possible to supply a parameter, which should be the name of a file. Wave Repair's behaviour depends on the type of that file, which it deduces from the file's suffix:

If the file has a suffix of **.WAV**, then Wave Repair will assume it is a WAV file, and attempt to load it into its main window. Thus, when the main window appears, the file given as a parameter will already be shown.

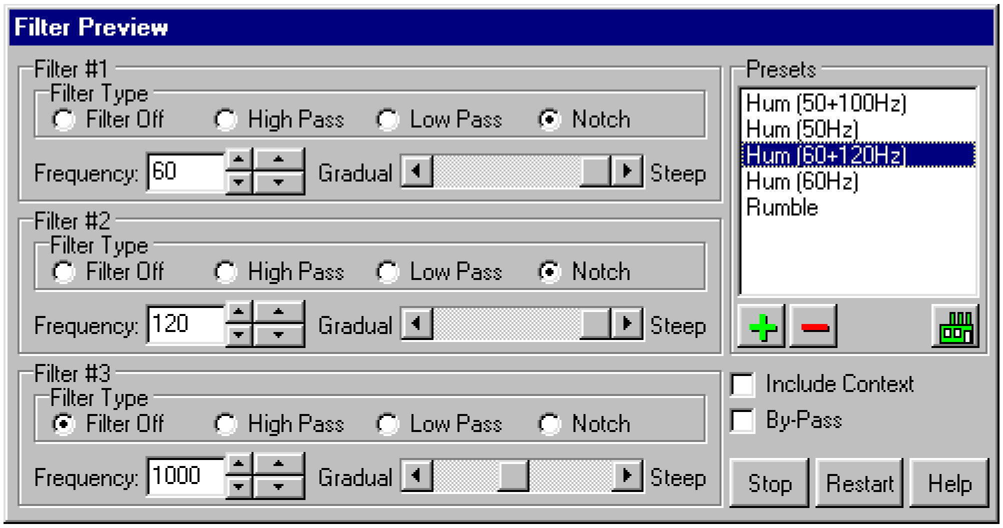
If the file has a suffix of **.MKR**, Wave Repair will assume it is a set of markers that have been stored via **Markers | Save Markers in File**. It will construct the name of the corresponding WAV file by replacing the **.MKR** suffix with **.WAV** and will attempt to load that WAV file. It will then load the set of markers in the MKR file.

If the file has a suffix of **.CLK**, Wave Repair will assume it is a set of clicks that have been stored via **Declicking | Save Clicks in File**. It will construct the name of the corresponding WAV file by replacing the **.CLK** suffix with **.WAV** and will attempt to load that WAV file. It will then load the set of clicks in the CLK file.

If the file has a suffix of **.CUE**, Wave Repair will assume it is a cue sheet that have been stored via **Cue Points | Write Cue Sheet**. It will construct the name of the corresponding WAV file by replacing the **.CUE** suffix with **.WAV** and will attempt to load that WAV file. It will then load the cue points in the CUE file.

It is envisaged that you might associate the suffixes **.MKR** and **.CLK** with Wave Repair so that files of this type can be double-clicked and cause Wave Repair to be started. The same could also be done with the suffixes **.WAV** and **.CUE**, although it is acknowledged that there are other programs which you might prefer to leave associated with those suffixes (eg. **.WAV** might be associated with another audio processing program, and **.CUE** might be associated with a CD writing program which can process cue sheets).

## Filter Preview




This is a modeless dialog which appears while **Other Effects | Filter Preview** is in operation. It allows the filter parameters to be adjusted while the selected samples are playing. When Filter Preview ends, the values which have been set here are retained for use when filtering is actually applied.

Up to three filters can be applied simultaneously, and the characteristics of each are given in the three panels marked **Filter #1**, **Filter #2** and **Filter #3**. Each filter may be either a **High Pass** (removes low frequencies), **Low Pass** (removes high frequencies), or **Notch** (removes a specific narrow band of frequencies). If **Filter Off** is selected, the corresponding filter is not used.

The **Frequency** of each filter denotes the center frequency at which it operates, and the slider marked **Gradual/Steep** determines how steeply the filter is applied. For High and Low Pass filters, a steep filter has less effect on the frequencies above and below the center frequency respectively, and removes the filtered frequencies much more completely. For Notch filters, a steep filter removes a narrower band of frequencies than a gradual one.

Within the **Presets** area, a list of named settings are shown which can be selected. New presets can be added by clicking the button with the green plus symbol. An existing preset can be removed by clicking the button with the red minus symbol. The button with the picture of a factory reinstalls the default presets that are shipped with Wave Repair. The factory presets provided are primarily intended to assist in reducing the effects of hum and/or turntable rumble which you may be unable to eliminate prior to recording.

If **Include Context** is checked, an additional amount before and after the selected region is included in the playback, but is not included in the filtering. This allows you to listen to how the transition points will sound.

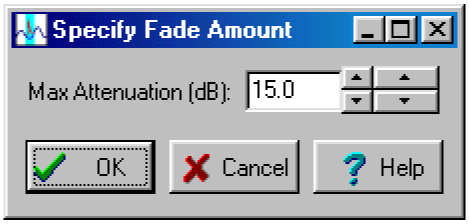
If **By-Pass** is checked, the filtering is switched off. This makes it easy to do a simple before-and-after comparison of the current settings.

The **Stop** button ends Filter Preview, which can also be ended via the **File | Kill (Stop) Output** menu option or by pressing the **Stop Playback** button on the toolbar.

The **Restart** button interrupts the current pass through, and begins playback again at the start.

**Partial Fade**

This dialog appears when **Partial Fade In** or **Partial Fade Out** is invoked:

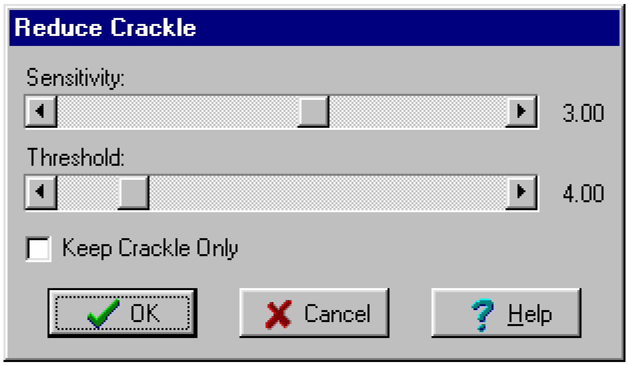



**Max Attenuation** defines the level of attenuation that will be applied at the extreme end of the fade. When fading in, the start of the selected region will be attenuated by this amount, and the level will gradually be increased until at the end of the selected region, no attenuation is applied. When fading out, the start of the selected region will have no attenuation, and then the level will be gradually decreased until the end of the selection, where the specified attenuation will be applied.

**Reduce Crackle**

Before a Decrackle operation is possible, a "noise fingerprint" must first be taken from a section containing only noise and light crackle. Bear in mind also that decrackling does not remove larger clicks and ticks; it is targeted at constant background small ticks.

This dialog appears when the **Noise Reduction | Decrackle** option is invoked:




The amount of decrackling which will be performed is determined by the two adjustments **Sensitivity** and **Threshold**. Decrackling works by first isolating the background noise (which is controlled by sensitivity: the higher this is set, the more background noise is isolated), then removing the crackle from that background noise (which is controlled by threshold: the lower this is set, the smaller a piece of crackle needs to be to be removed).

So in summary, increasing the sensitivity and/or decreasing the threshold will increase the amount of decrackling, but at the danger of possibly degrading the musical signal. Conversely, decreasing the sensitivity and/or increasing the threshold will be less likely to damage the musical signal but will remove less crackle. The aim is to find the smallest sensitivity and the largest threshold which adequately reduce the crackle while not damaging the music. The precise manner in which these two parameters interact depends on the nature of the record's background noise and cannot be predicted with any degree of confidence. The default settings of 3 for sensitivity and 4 for threshold have been chosen as a reasonable compromise, but it is advisable to experiment for each individual record to be decrackled.

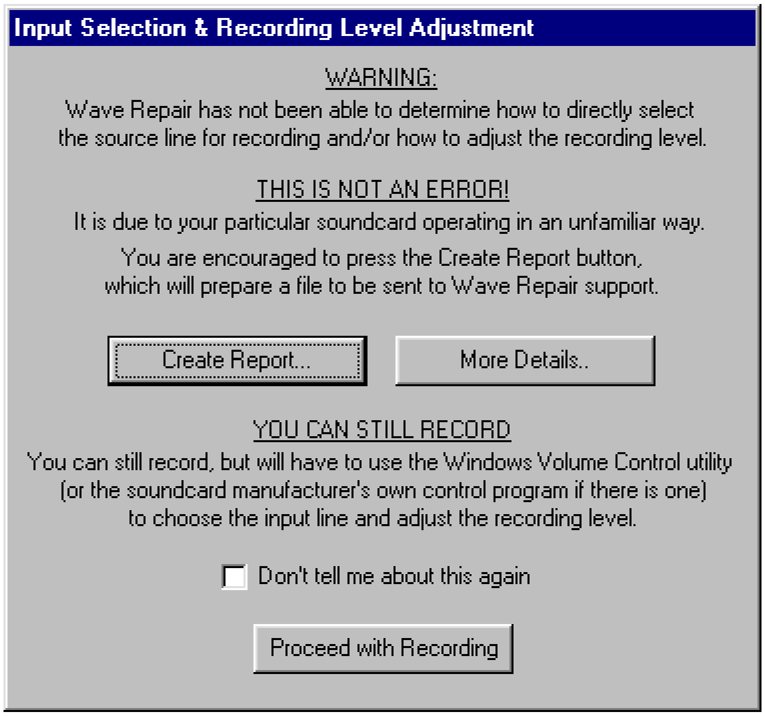
Note that it is inevitable that some records will have a combination of crackle and music which means that you cannot remove all the crackle without damaging the music unacceptably. Reed instruments (such as oboe, saxophone), brass (eg. trumpet) and "metallic synthesiser sounds" are most at risk of distortion if an excessive amount of decrackling is applied.

Unfortunately, due to the complex nature of decrackling, there is no realtime preview available. For this reason it is best to experiment on a few short sections first in order to find the appropriate reduction factor setting.

The **Keep Crackle Only** option allows you to keep only the crackle that would normally be removed. This can be used to get a feeling for how much crackle would be removed, and to check that little or none of the wanted signal is being discarded along with the crackle.

**Selecting the Soundcard Input and Setting the Recording Level**

If you receive this warning message:




Please send a trace file to Wave Repair support. To do this, press the **Create Report** button. This will write a small text file called **soundcard_trace.txt** into the same directory that contains the Wave Repair executable (WAVREP32.EXE). Please email this generated text file to me; I will then study the way that your soundcard behaves with a view to adding support for it in the next release. Wave Repair itself does not attempt to email the file; you must do this manually. Since it is possible that I may need to change my email address, please check on the Wave Repair web page at **www.waverepair.com** for the current email address.

Note that the appearance of this warning message does NOT prevent you actually recording; it's just that you will have to use other facilities (eg. the soundcard's own control program, or the Windows Volume Control utility) to select the input and adjust the record level.

Background Discussion

The standard Windows Volume Control utility can be rather confusing, and many people have found it difficult to use in order to select the required soundcard input for recording. For this reason, Wave Repair attempts to provide facilities to select the input and set the recording level directly from the Recording dialog. Unfortunately, soundcards vary in the way that these facilities are controlled. Wave Repair supports some common methods used by many soundcards, but it cannot know in advance how to do this for every possible soundcard. Therefore, in some cases it fails to set up the input selector and level controls and the above warning message is displayed. The times when this warning message will appear fall into two categories:

1. Some soundcards intentionally do not support the standard Windows Mixer Interface. Many professional and semi-professional soundcards fall into this category. They tend to be unlike mainstream soundcards, and conforming to the interface would not be appropriate. In this case, there is no way that Wave Repair can present the desired controls and it will be necessary to use the soundcard's own mixer utility. It is advisable in this case to select the **No Record Input Line/Level Control** option on the Options dialog, to tell Wave Repair not to even bother trying; this will then eliminate the warning message which would otherwise be given. You can easily tell if your soundcard is of this type by attempting to run the standard Windows Volume Control utility: if it does not run, then your soundcard does not support the interface.

2. Other soundcards do support the Windows Mixer Interface, but in a manner that is different to the common ways that Wave Repair knows about. Soundcards of this type can in principle be supported by Wave Repair, but their individual characteristics will have to be added in a future release. In this case, you are encouraged to generate the trace file mentioned above.

**Decrackling**

Some records have a constant background of tiny ticks that sound a little like bacon frying, or rice crispies in milk. This type of background noise is commonly referred to as "crackle". If there is crackle on the record being processed, it should be reduced before any other restoration steps (with the exception of hum and/or rumble removal, which should be done first).

You may have read elsewhere that medium and large clicks should be removed before attempting decrackling, as the clicks can confuse the decrackling algorithms. With many other decrackling tools this well be true, but Wave Repair's decrackling mechanism is fairly immune to the presence of larger clicks. While it is certainly true that removing clicks before decrackling cannot hurt, and in rare cases could reduce the possible artifacts of decrackling, there is on balance a good reason to perform decrackling before declicking. This is that declicking is a fairly intensive manual process, and it is impossible to know whether particular small clicks might be removed by the decrackling. If you decide to declick before decrackling, then you only have to declick again afterwards to check for small clicks that the decrackling missed. It is much less effort to do the decrackling first and then make just one declick pass over the file.

Wave Repair's decrackling mechanism replies on having a representative sample of what the background noise is like. We refer to this sample as the "noise fingerprint". Just as individual people have their own unique fingerprints, every record has its own unique noise fingerprint. Therefore, before a decrackling operation can be performed, you must first select a region which contains only background noise (along with the crackle). It should not contain any musical signal or large clicks. The run-in groove, or the gap between two tracks, is usually a good place to find such a region. (If you choose the run-in groove, be careful not to include the sound of the stylus first entering the groove in the region).

Once a region of background noise and crackle has been selected, invoke **Noise Reduction | Get Fingerprint**. This will analyse the selected noise and save its characteristics for subsequent decrackling operations.

To actually perform the decrackling, select the region to be decrackled, then invoke **Noise Reduction | Decrackle**. The Reduce Crackle dialog will appear which allows you to select the amount of decrackling required.

Due to the complex nature of decrackling, there is no realtime preview available, so it is best to experiment on some short sections first in order to find a suitable amount of decrackling before processing a lengthy section. Note that after experimenting on a short section, you should undo the operation before applying the final decrackle. After decrackling is complete, you should review the results to determine whether they are satisfactory before saving.

Many records will only require decrackling to be applied to quiet sections (louder music tends to mask any crackle), and by limiting the decrackling operations to just the regions where it is audible will save time.

**Noise Reduction Menu**

Get Fingerprint
This option analyses the selected region and records its characteristics as a "noise fingerprint". This fingerprint is required for subsequent noise reduction and/or decrackling operations. The selected region thus analysed should ideally contain only background noise and light crackle. It should not contain any musical signal or large clicks.

Decrackle (With Undo)
This reduces the amount of low-level "crackling" (the constant background of tiny ticks) found on vinyl records. It requires a fingerprint to have been taken first. This particular menu item performs the decrackling as an update stored in memory or a temporary file so that its effect can be removed with the Undo option. The decision whether to use memory or a temporary file depends on the **Where Edits are Stored** setting in the Options dialog. Note that if you select the **Always in Virtual Memory** setting, and decrackle a very long section, then this would consume a vast amount of memory. If your computer has the resources to do this, then it will work but may run extremely slowly.

Decrackle (Direct Overwrite)
Performs the same operation as Decrackle (With Undo), but stores the results by directly overwriting the WAV file on disk. This has the advantage that it runs faster than Decrackle (With Undo), but the disadvantage is that it cannot be backed out with the Undo option. You should be confident that the chosen decrackling settings are acceptable before using this menu item. You are strongly advised to do some experiments on short sections using Decrackle (With Undo) to establish the correct settings before using this option.

Reduce Noise (With Undo)
This subtracts the characteristic noise that was taken as a fingerprint from the selected region. Overly ambitious amounts of noise reduction will result in unacceptable degradation of the music signal, so it should be used in moderation. This menu item performs the noise reduction as an update stored in memory or a temporary file and its effects can be removed with the Undo option.

Reduce Noise (Direct Overwrite)
Performs the same operation as Reduce Noise (With Undo), but directly overwrites the WAV file on hard disk. This runs faster than Reduce Noise (With Undo), but the results cannot be backed out with Undo. You should be confident that the noise reduction will produce acceptable results before using this menu item.

Save Fingerprint in File
Stores the current noise fingerprint in a file.

Load Fingerprint from File
Reloads a noise fingerprint that was previously saved.

**Support: Contacting the Author**

If you need to contact the author for any reason (whether it be for support or advice), please do so via email.

Since it is possible that the author's email address might change in the future, please check on the Wave Repair web page at **www.waverepair.com** for the current email address.

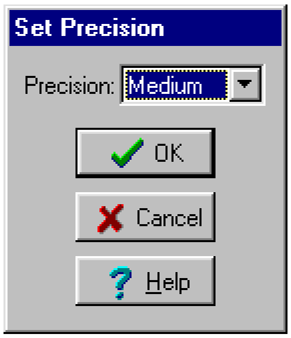
**Slow Operations When There Are Unsaved Edits**

Because many of Wave Repair's editing features involve changes to very small numbers of audio samples relative to the total size of the WAV file, the way it stores edits is optimised for these short edits. As a result, lengthy edits which involve changes to all of the audio samples in a selected region (such as amplification, equalisation, decrackling, etc) are not stored particularly efficiently. As a general rule one long edit of this type can be performed without any trouble, but once the edit has been made, subsequent lengthy edits will proceed very slowly unless you first save the results of the previous edit. For example, suppose you wish to equalise and then normalise: the initial equalisation will proceed normally, but if it is left unsaved the subsequent normalisation will run very slowly. In order to achieve good performance on the normalisation, it is necessary to save the results of the equalisation first. (Of course, you should review the results of the equalisation before saving it).

(Note that this does not apply to operations carried out using the "Direct Overwrite" option).

**Set Precision Dialog**

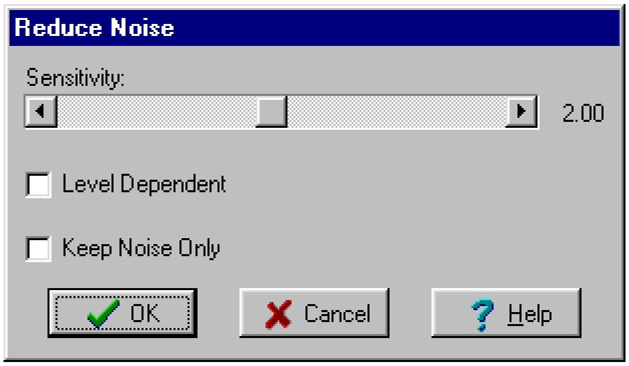
When taking a noise fingerprint for use in decrackling or broadband noise reduction, this dialog appears:




This selects the accuracy with which the fingerprint (and subsequent decrackling or noise reduction) is performed. The higher the selected precision, the more accurate the results will be, but the processing takes longer. As a general rule, a precision of **Medium** is sufficient for decrackling, since the accuracy is not especially critical when decrackling. For broadband noise reduction, a precision of **High** or **Very High** is generally advisable (although lower precisions can be experimented with if speed of processing is desired).

**Reduce Noise Dialog**

This dialog appears when **Noise Reduction | Reduce Noise** is invoked:




The amount of noise reduction that will be performed is determined by the **Sensitivity** setting. Increasing the sensitivity will increase the amount of noise reduction applied, but at the danger of degrading the music signal.

Unfortunately, due to the complex nature of broadband noise reduction, there is no realtime preview available. For this reason it is best to experiment on a few short sections first in order to find the appropriate setting.

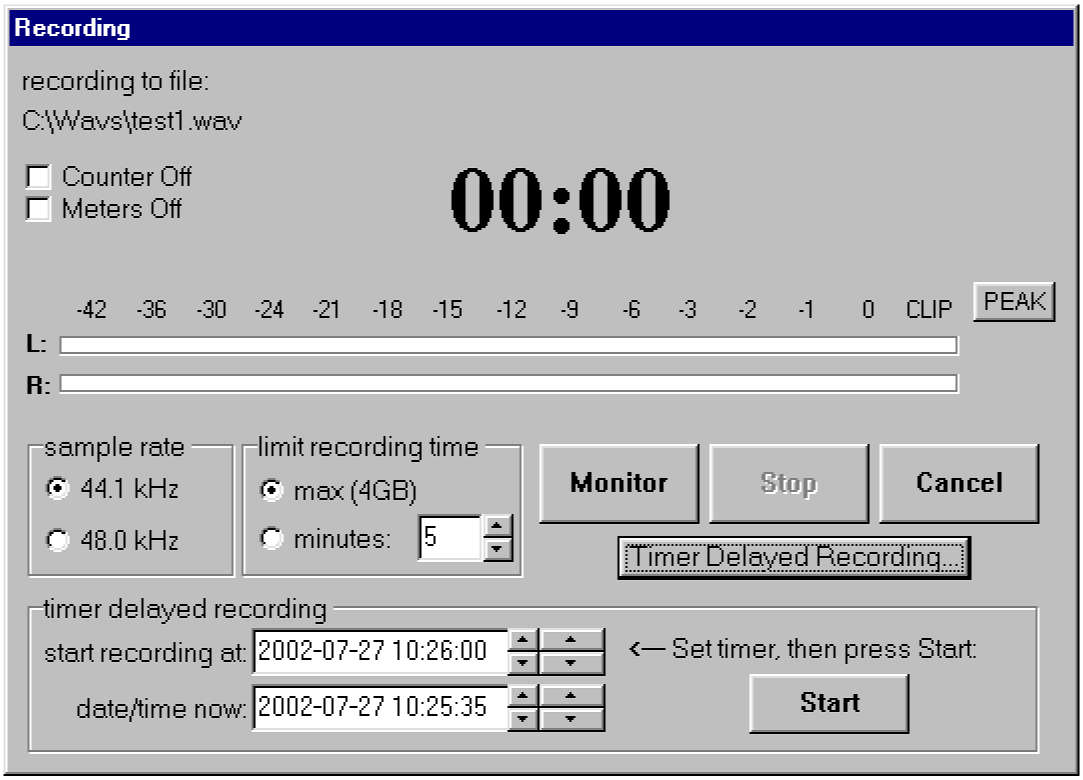
If the **Level Dependent** option is switched on, the amount of noise reduction applied is inversely dependent on the level of the music. When the music is loud, less noise reduction is applied; when the music is quiet, more noise reduction is applied. The rationale behind this is that loud music tends to mask the background noise, so it is possible to get away with less noise reduction, and hence minimise the danger of audible artifacts.

One situation in which the **Level Dependent** option is very useful is where you wish to gradually introduce noise reduction during the fade-out of a song. At the start of the fade-out, when the music is not particularly quiet, you need to apply a small amount of noise reduction (otherwise there will be an audible change in the noise characteristic at the start of the noise-reduced section). Meanwhile, towards the end of the fade-out, much more noise reduction is required so as to eliminate the (now prominent) background noise. One approach would be to select short sections and apply noise reduction with varying **Sensitivity** settings (low at the start of the fade-out, high at the end). This of course is a perfectly acceptable method, but it is rather tedious. By using the **Level Dependent** option, the amount of noise reduction will automatically be gradually increased as the level of the music decreases during the fade-out. (This applies to fade-ins at the start of songs as well, of course).

The **Keep Noise Only** option allows you to keep only the noise that would normally be removed. This can be used to get a feeling for how much noise would be removed, and to check that little or none of the wanted signal is being discarded along with the noise, although you should note that it is inevitable that some recognisable portion of the music will be heard in the kept noise. The only reliable way to judge if the music is unacceptably degraded is to listen to the results of a noise reduction with the **Keep Noise Only** option switched off.

**Timer Delayed Recording**

When the Recording dialog appears, if you press the **Timer Delayed Recording** button, an additional section of the screen appears:

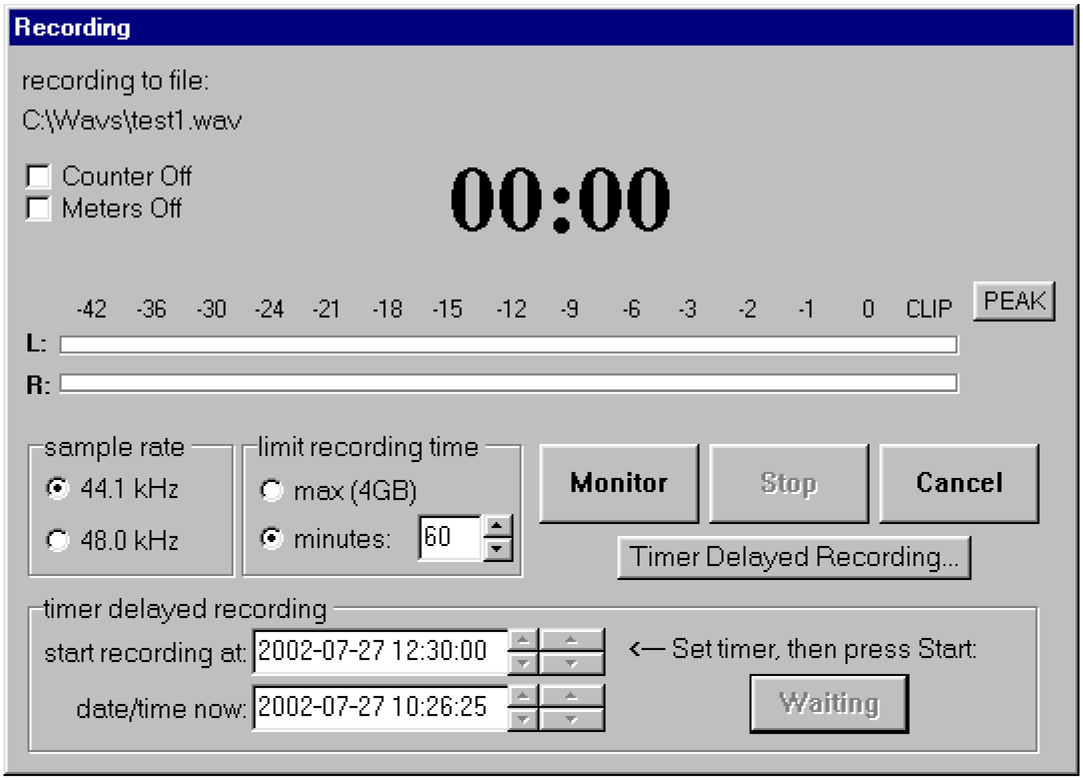



The **start recording at** box is where you specify when the recording should begin. The small up & down buttons adjust the start time by minutes; the large buttons adjust it by hours.

The **date/time now** box displays the current date and time. This begins by showing the date and time according to the PC's clock. However, since PC clocks can sometimes be inaccurate, you can adjust this using the up & down buttons. It is important to bear in mind that adjusting the date/time now does <u>not</u> change the PC's clock. It simply tells Wave Repair to pretend that the time is as selected. The reason for this facility is as follows. Suppose you wish to record a radio program that will start at 9pm, and suppose also that your PC's clock happens to be 13 minutes fast. It is much easier to "normalise" the **date/time now** to the actual current time and set **start recording at** to 21:00 than it is to work out that **start recording at** should be set to 20:47.

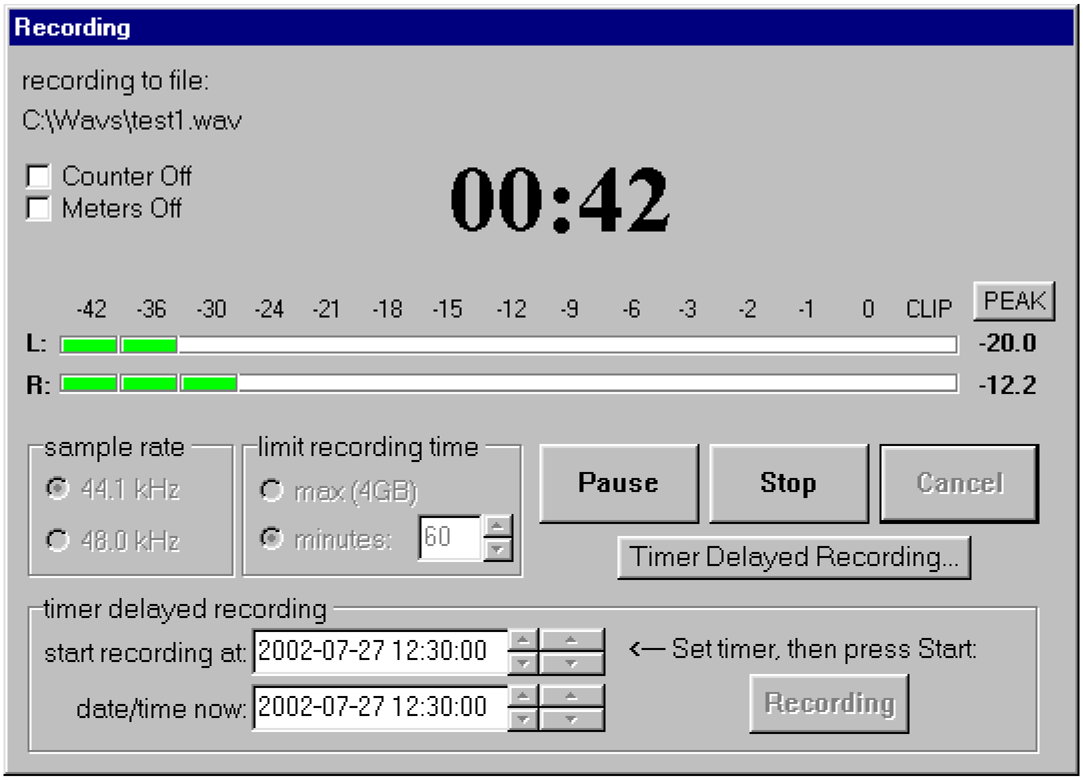
To specify when recording should finish, simply set the **limit recording time** option accordingly.

Once the start time and recording time have been set, press the **Start** button. Wave Repair will now wait for the specified start time to arrive:




At any time while the screen is waiting to begin recording, you can manually override it by pressing the **Monitor**, **Record** or **Cancel** buttons if you wish.

Once the designated start time arrives, recording automatically begins:




Note that the **date/time now** counter stops incrementing once recording has started. This is done deliberately to reduce the (already extremely small) chance that the resources consumed by the ticking clock might cause dropouts during recording.

Once recording has begun, the behaviour of the recording facility reverts to its standard mode: you can use the **Pause** and **Stop** buttons manually if you wish.

**Play/Record Menu**

Output (Play) Selection
Plays the selected samples via the selected soundcard (which must support 44.1/48kHz 16-bit stereo).

Output (Play) Selection - Left Only
Plays only the left channel of the selected samples.

Output (Play) Selection - Right Only
Plays only the right channel of the selected samples.

Play Selection with Context
Plays the selected samples together with the surrounding context. The context is a user-definable time period preceding and following the selected samples that will also be played.

Play Selection with Context - Left Only
Plays the left channel of the selected samples with the surrounding context.

Play Selection with Context - Right Only
Plays the right channel of the selected samples with the surrounding context.

Play Context Around Selection
Plays the samples before and after the selected samples (as in Play Selection With Context), but skips the selected samples themselves. This can sometimes be useful in determining whether a particular short sequence of samples includes an audible piece of damage that is difficult to identify visually.

Play Context Around Selection - Left Only
Plays the left channel samples before and after the selected samples.

Play Context Around Selection - Right Only
Plays the right channel samples before and after the selected samples.

Kill (Stop) Output
Stops playback of the selected samples.

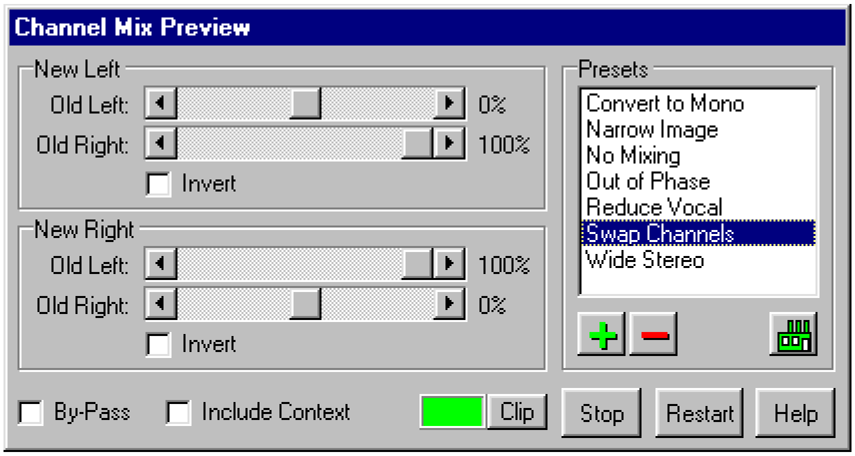
Play Left Channel
Toggles whether the left channel will be included during playback. It can be useful to switch off one of the channels if you wish to concentrate only on the other channel.

Play Right Channel
Toggles whether the right channel will be included during playback.

Input (Record) New WAV File
Records a new WAV file via the selected soundcard (which must support 44.1/48kHz 16-bit

stereo). See Recording dialog.

**Channel Mix Preview**




This is a modeless dialog which appears while **Other Effects | Channel Mix Preview** is in operation. It allows the mixing settings to be adjusted while the selected samples are playing. When Channel Mix Preview ends, the values which have been set here are retained for use when mixing is actually applied.

**New Left** controls the proportion of the existing left and right channels that will be mixed together to produce the resulting left channel. If the **Invert** option is checked, then the resulting new left channel with have its phase inverted (ie. the waveform will be "turned upside down").

**New Right** controls the proportion of the existing left and right channels that will be mixed together to produce the resulting right channel. If the **Invert** option is checked, then the resulting new right channel with have its phase inverted.

Within the **Presets** area, a list of named settings are shown which can be selected. New presets can be added by clicking the button with the green plus symbol. An existing preset can be removed by clicking the button with the red minus symbol. The button with the picture of a factory reinstalls the default presets that are shipped with Wave Repair.

If **Include Context** is checked, an additional amount before and after the selected region is included in the playback, but is not included in the channel mixing. This allows you to listen to how the transition points will sound.

If **By-Pass** is checked, the mixing is switched off. This makes it easy to do a simple before-and-after comparison of the current settings.

The **Stop** button ends Channel Mix Preview, which can also be ended via the **File | Kill (Stop) Output** menu option or by pressing the **Stop Playback** button on the toolbar.

The **Restart** button interrupts the current pass through, and begins playback again at the start.

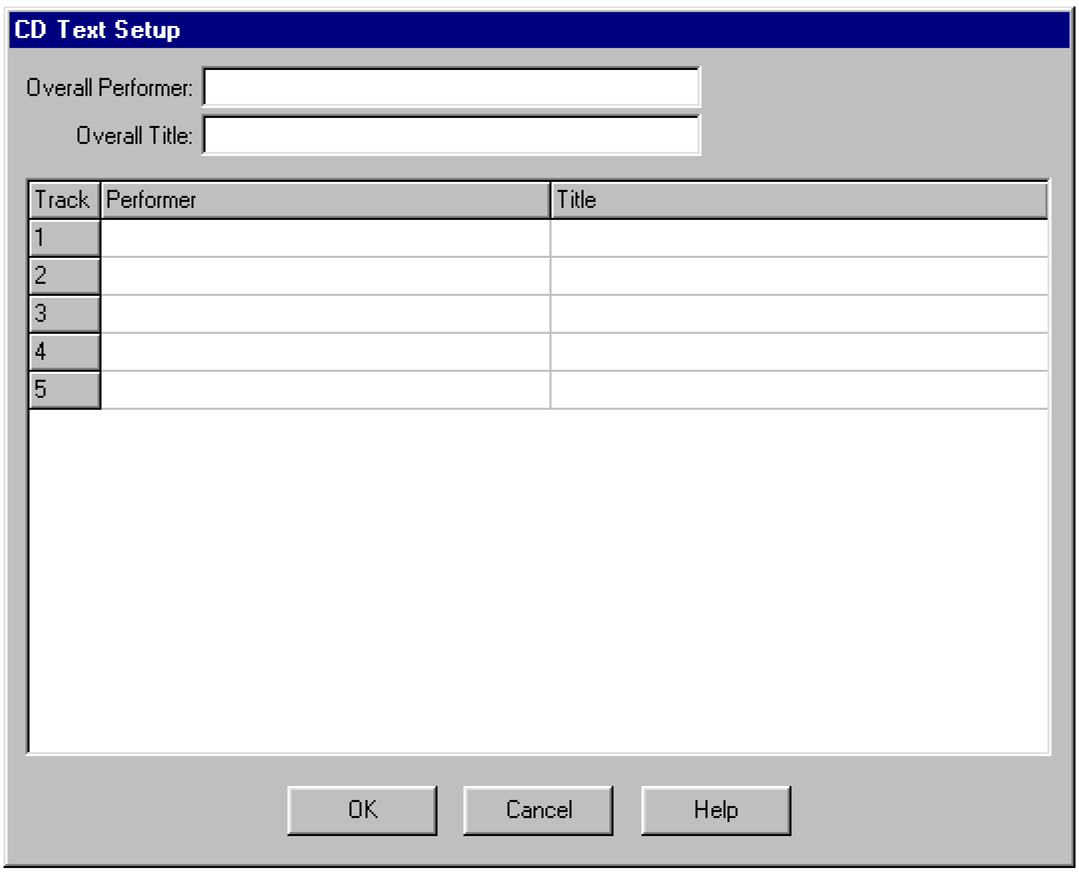
## CD Text Setup Dialog

Some CD writers are capable of adding CD Text information while they write an audio CD. Wave Repair can assist in this, but only if you write the CD using a cue sheet. (If your CD writing software is of the type which does not use cue sheets, any CD Text information you require will have to be set up within the CD writing package itself).

Please note that CD Text information in a cue sheet should simply be ignored by CD writers which do not support CD Text, although it is possible that the presence of CD Text data in a cue sheet could result in an error message from some writers which do not support CD Text.

Although the CD Text standard includes a number of options, only two are supported by Wave Repair: **Performer** and **Title**. These two options are supported on both an overall and per-track basis.

When **Cue Points | Setup CD Text Data** is invoked, a dialog similar to this appears:

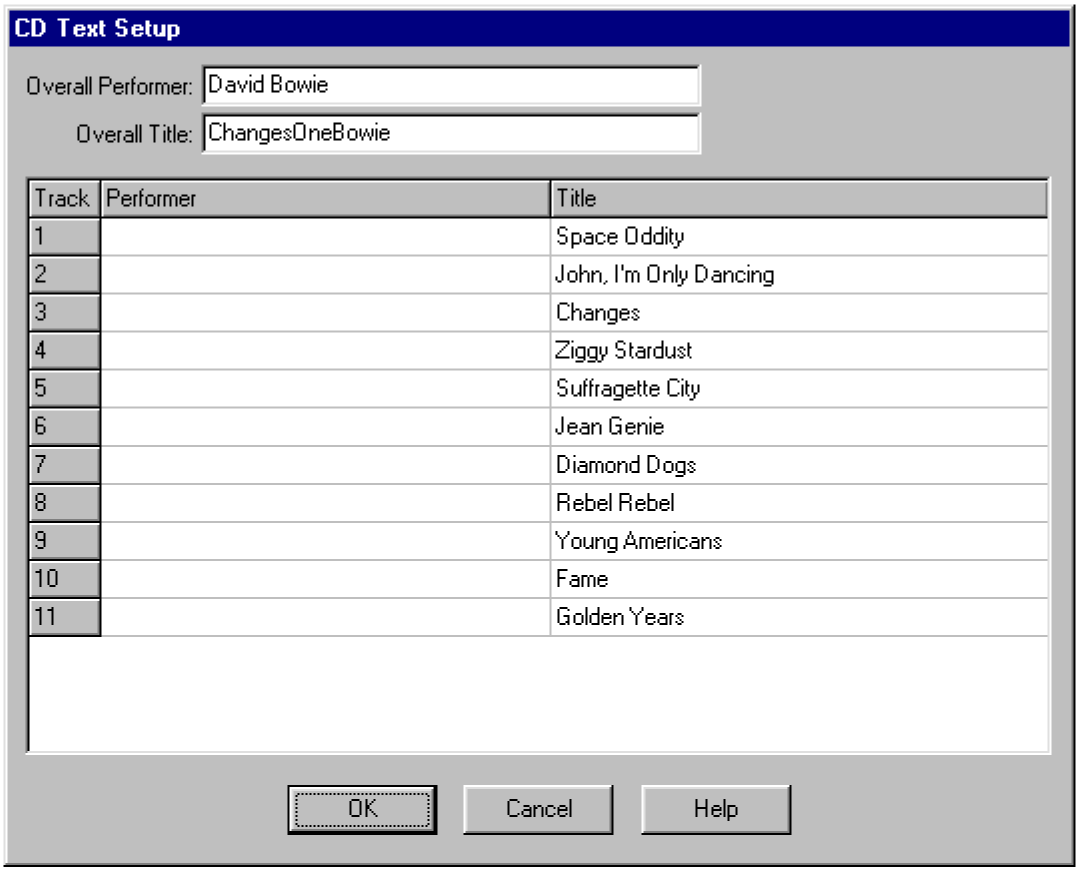



The number of entries in the grid of individual tracks depends on the number of cue points which have been added. In this example, four cue points (representing tracks 2 to 5) have been added (track 1 is assumed to start at the beginning of the file).

**Overall Performer** and **Overall Title** allow you to set up the performer and title of the entire work. For each track in the grid underneath, you may set up the performer and title if required.

Typically, if you are dealing with a single LP record, you would set the overall performer and title, and just the titles of the individual tracks. You would normally only set the performer of the individual tracks if this is a compilation of various artists.

For example, suppose you are restoring the album "ChangesOneBowie", you might set up the CD Text data thus:

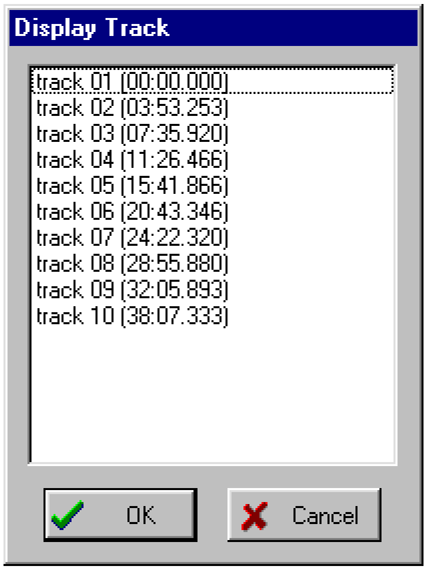



The data set up in this way is retained by Wave Repair, and if you subsequently write a cue sheet (using **Cue Points | Write Cue Sheet**), then this CD Text data is added at the appropriate places.

Please note that if, when you write the cue sheet data, you append to an existing cue sheet, the overall performer and title are ignored: the existing overall performer and title (if any) in the existing cue sheet are retained.

**Display Track Dialog**

This dialog appears when **Cue Points | Display Track** is invoked:




This dialog shows the tracks currently defined by the cue points that exist. Note that Track 1 is always assumed to exist, and extends from the start of the file to the first track cue point (or the end of the file if no track cue points exist). The desired track is selected, and the display is changed so as to display the entire track.

**PayPal**

I have a PayPal account, so you can send the registration fee that way if you wish. The amount is 30 US dollars, and it should be sent to **clive@delback.co.uk**. Be sure to specify that it is for Wave Repair Registration, and to include your email address in the accompanying note so that I know where to send the unlock key. When PayPal notify me of the transaction, I will send your unlock key by email.

**Cash Payment**

If you are prepared to risk it, you can send cash by postal mail. I can accept either 30 US Dollars, 30 Euros, or 20 British Pounds. I am sorry for the rather crude conversion between currencies, but it is impractical to continually review exchange rates. I will send your unlock key as soon as the money arrives. The address to send payment is:

32 Foxleys
Watford
WD19 5DE
UK

Be sure to include your email address in your letter, so that I know where to send your unlock key. And PLEASE make sure the email address is legible: there have been a few occasions where I've had to make a guess as to what the email address is.

**Personal Cheque Payment**

If you have an account with a UK bank or building society, I can accept personal cheques. The amount in this case should be 20 Pounds. Please make cheques payable to Clive Backham. I will send your unlock key as soon as the cheque arrives. (I do not wait for the cheque to clear - if you're prepared to trust me, the least I can do is trust you!). The address to send the cheque to is:

32 Foxleys
Watford
WD19 5DE
UK

Be sure to include your email address in your letter, so that I know where to send your unlock key. And PLEASE make sure the email address is legible: there have been a few occasions where I've had to make a guess as to what the email address is.

I'm sorry, but I cannot accept checks from non-UK banks, as the fee charged by my bank to cash them is far too high to make it worthwhile.

**Electronic Bank Transfer Payment**

You can wire the money directly to my UK bank. Your local bank should be able to perform this transfer for you, but you will probably have to pay them a fee for sending the money. I also pay a fee to my UK bank to receive the transfer, but I absorb this from the registration fee. The amount to be transferred is 20 British Pounds (your bank may refer to this currency as "UK sterling"). The money should be sent to:

UK bank name: Lloyds Bank
bank number: 30-92-86
account number: 01667974
account name: C A Backham
bank address:

Lloyds Bank
Polegate Branch
41 High Street
Polegate
East Sussex
BN26 5AB
United Kingdom

Please email me to let me know that you have sent the funds. I will then confirm the transfer with my bank and email your unlock key to you. Since it is possible that my email address may need to change, please check on the Wave Repair web page at **www.waverepair.com** for the current email address.