

ClickRepair 3.9 & ClickRepairRT 1.4

www.kagi.com/clickrepair

(for Java-enabled platforms)

User's Manual

Brian Davies

clickrepair@kagi.com

Preamble – Please Read.

I began to write this software in 2004. I had started to transfer my LP collection to CD, and had everything I needed. Or so I thought, until I decided that it might be nice to remove the clicks and cracks from my recordings, without doing too much damage to the music. My search for suitable software, if I wanted to do a half-decent job at an affordable price, was frustrating. So began this project.

If you find ClickRepair useful, please help by posting information to news and discussion groups, and ratings feedback on the site where you found it. This way the circle of users will increase, and I shall feel encouraged to devote the time and effort required to maintain and improve the software.

ClickRepair and ClickRepairRT are written in Java, a system which allows the same application to run on different platforms (Mac OS X, Windows, Linux). The Windows installer is produced using “Inno Setup 5” by Jordan Russell. The manual is produced using Apple’s “Pages” application, typeset in Garamond, and exported as PDF.

I wish to thank those users who have contributed to the design of the Interface. I owe a particular debt to Michel Poulain and Willy Kunz, who spent many hours working on the French and German translations, and also provided much feedback.

Table of Contents

1	Introduction	5
	<i>Basics</i>	5
	<i>Important Warnings</i>	5
	<i>Organisation of Manual</i>	6
	<i>Recommendations</i>	6
	<i>Further Information</i>	6
2	Installation and Registration	7
	<i>Apple Mac OS X</i>	7
	<i>Microsoft Windows</i>	7
	<i>Registration</i>	8
	<i>Access to Registration Information</i>	8
	<i>Important Notice</i>	9
3	Quick Start	10
	<i>Initial Interface</i>	10
	<i>Basic Controls</i>	10
	<i>Choosing Files</i>	11
	<i>Progress in Automatic Mode</i>	12
4	Producing and Organising Files	13
	<i>File Requirements</i>	13
	<i>Recommended Order of Processing</i>	13
	<i>Notes about File Structure</i>	14
	<i>File Lengths</i>	15
	<i>Sample Sizes</i>	16
	<i>Sample Rates</i>	17
	<i>Reading from CDs</i>	18
5	User Interface Basics	19
	<i>Visual Interface</i>	19
	<i>The Sound Bar</i>	19
	<i>The Display Window</i>	20
	<i>The Scale Controls</i>	20
	<i>Menus</i>	21

	<i>Keyboard Shortcuts</i>	22
	<i>Making a Repair</i>	22
	<i>Audio Monitoring</i>	23
	<i>Sound Latency</i>	23
	<i>The Progress Indicator</i>	24
6	Detection and Repair Settings	25
	<i>The DeClick Control</i>	26
	<i>The DeCrackle Control</i>	26
	<i>The Automatic Control</i>	27
	<i>Method – Simple versus Wavelet</i>	27
	<i>Pitch Protection</i>	29
	<i>Possible conflict between DeCrackle and Pitch Protection</i>	30
	<i>Saving and Recalling Settings</i>	30
	<i>Deciding on Appropriate Settings</i>	31
7	Audio Monitoring and the Sound Buffer	32
	<i>Java Sound Output settings</i>	32
	<i>The Sound Bar</i>	32
	<i>Listening to the Sound Buffer</i>	33
	<i>Undoing Repairs in the Sound Buffer</i>	34
	<i>The Last Buffer</i>	34
8	Using ClickRepairRT	35
	<i>Audio Input and Output</i>	35
	<i>Using a Sound Input Jack</i>	35
	<i>Limitations and Differences</i>	36
9	Processing Mono Records	37
	<i>File Formats</i>	37
	<i>The Stereo>Mono Option</i>	38
	<i>Viewing Merge Details</i>	39
	<i>Shellac Records</i>	41
10	More Advanced Methods	43
	<i>Pops and Bumps</i>	43
	<i>Wavelet ..x2 .. x3</i>	43

	<i>Flattening of Manual Repairs</i>	44
	<i>A Genuine Pop</i>	46
	<i>Time-reversed Processing</i>	46
11	Manually Adjusting a Repair	48
	<i>Using the Mouse</i>	48
	<i>Using the Arrow Keys</i>	48
12	Batch Processing	50
	<i>Organisation</i>	50
	<i>Assembling a Batch</i>	51
	<i>Processing</i>	52
13	The Preferences Dialog	53
	<i>Preferences</i>	53
	<i>Language</i>	53
	<i>Sound Buffer Length</i>	53
	<i>Log Files</i>	54
	<i>Files and Processing</i>	54
	<i>Breakpoints and Markers</i>	54
	<i>Resetting Defaults</i>	55
14	Other Features	56
	<i>Behaviour of Buttons</i>	56
	<i>Abort Button</i>	56
	<i>File Dialogs</i>	56
	<i>Keyboard Operation</i>	57
	<i>Mouse Actions</i>	57

1 Introduction

Basics

ClickRepair is an application for cleaning clicks and crackle from audio files produced by digitizing vinyl (LP) and shellac (78) records. This kind of intermittent noise is typically the result of damage due to scratches, dust specks, etc.

It is important to understand the following points:

- It is impossible to distinguish music from damage, with absolute certainty.
- Repairing damage without a high degree of automation is very time consuming.
- Filtering the file with a “noise filter” degrades the musical content whilst merely reducing intermittent noise such as clicks and crackle.

ClickRepair is not a noise filter; it searches for, and repairs, localised damage. For records in reasonable condition, this means that 99.5% or more of the audio is unchanged. If the source material is an LP record in good condition, captured on good equipment, it is quite possible that the 99.5% sounds at least as good as, or even better than, a typical CD. The underlying philosophy of ClickRepair is to preserve that 99.5% and use it to find and replace the damage. It is indeed surprising how annoying and disturbing a sprinkling of clicks and dust specks can be to the listening experience.

ClickRepair works with mono or stereo sound files up to 32 bits at 192 kHz. This includes standard CD and DVD audio formats. It is designed to run automatically, although there are mechanisms for manual intervention. It is also possible to stop at Markers previously inserted by a suitable sound editor.¹

ClickRepairRT is a small application, bundled with ClickRepair, which is a simplified “real-time” version of ClickRepair. It accepts input from the sound-input stream provided to the Java run-time by the computer operating system, applies the declack algorithm of ClickRepair, and sends the output to the computer sound-output stream via the Java run-time. It is designed for situations where a dedicated computer is available for insertion into the play-back stream.

Important Warnings

While many users report excellent results using automatic processing with appropriate settings, please note:

- Automatic processing may not work well with some types of music.

¹ Regrettably, the “standard” method of inserting markers into the audio files themselves, thus making them interoperable between different applications, has been largely ignored by the developers of most Audio Editors, except for a couple that are popular on Mac OS X.

- If you find automatic settings which work well, you should save them with names that suggest the music type for which they were satisfactory.
- Clicks may be detected and removed in manual mode, permitting ClickRepair to be used with difficult music types. Inevitably, this will be tedious.
- Do not delete the original file until you are happy with the quality of the new file.
- The Quick Start chapter is not a substitute for reading the manual – after you have made some initial experiments, please take the time to read it.
- Before purchasing a license key, be sure to appraise the usefulness of ClickRepair for your particular needs. That is the purpose of the trial period!

Organisation of Manual

This manual is organised into the following chapters:

- Installation and Registration (Chapter 2).
- Quick Start (Chapter 3).
- Producing and Organising Files (Chapter 4).
- User Interface (Chapter 5).
- Detection and Repair Settings (Chapter 6).
- Audio Monitoring and the Sound Buffer (Chapter 7).
- Using ClickRepairRT (Chapter 8).
- Processing Mono Records (Chapter 9).
- More Advanced Methods (Chapter 10).
- Manually Adjusting a Repair (Chapter 11).
- Batch Processing (Chapter 12).
- The Preferences Dialog (Chapter 13).
- Other Features (Chapter 14).

Recommendations

Once you have downloaded and installed the software, I recommend that you start the program, and use chapter 3 (Quick Start) to process one or two files in automatic mode, with the sound turned on. This will serve:

- As an initial introduction to the application.
- To ensure that you are properly set up, in relation to producing the required files, using whatever software you choose for that purpose.

Subsequent exploration of the features of ClickRepair, and the corresponding chapters of this manual, may then be made as necessary.

Further Information

A detailed history of all versions of ClickRepair is on the web-site, www.kagi.com/clickrepair.

2 Installation and Registration

Apple Mac OS X

Users should download the file ClickRepair.dmg. Upon double-clicking the file, it will mount as a volume. The contents should be dragged to the desired location on your hard disk – most likely the applications folder.

ClickRepair and ClickRepairRT for Mac OS X require a Java run-time, either Apple’s legacy Java 1.6 run time, or a more up-to-date Java run-time from Oracle. Mac OS X does not have Java pre-installed – you will be prompted to install it the first time you attempt to run a Java application. Moreover, the current version Mac OS X 10.11 (El Capitan) is the last for which Apple will continue this provision, and it will be necessary to use the Java “jar” file with Oracle’s Java run-time. No functionality will be lost; moreover some bugs in Apple’s Java 1.6 will be circumvented.²

There is more information in the READ_ME file in the download.

Microsoft Windows

Microsoft discontinued Java deployment for Windows some time ago. ClickRepair users should download and run the installer CRInstall.exe, which does not come with a Java run-time. You will be prompted to install one, if necessary. My applications have been tested with various Java run-time versions on Windows XP through to Windows 10.

Shortcuts to the Application and to the Manual are installed in the Windows Start>Programs menu at Start>Programs>ClickRepair.

Alternative Deployment (All Platforms)

Linux, Mac OS X and Windows users can all run the Java “jar” files directly. To do this, download and unpack the Linux version (a zip file), which will deliver two jar files and this manual (pdf). These jar files are identical to the jar files included in the Mac or Windows versions, but they lack the normal packaging of applications specific to those platforms. Double-clicking ClickRepair.jar or ClickRepairRT.jar should launch the applications using the built-in Java launcher, provided the system is equipped with a modern Java run-time. Linux users may need to invoke them from the command line, or they may respond to double-clicking. This method of deployment can be useful, on Mac or Windows systems, if the launch mechanisms that are part of the standard downloads fail to work for some reason. There is no package or installer provided for Linux users.

² In particular, the real-time version will have a much better range of input/output audio options using third-party Java run-times, and sound monitoring of audio above 48kHz will be more accurate. Actual processing of files is not affected, however.

Registration

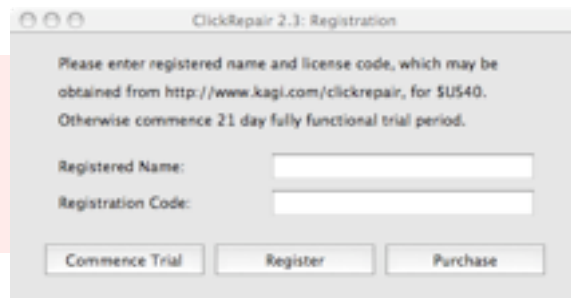
Each time you start ClickRepair, a registration dialog will appear. Click “Commence trial” or “Continue trial” in order to evaluate the program. There is a limit of 21 days trial use. Evaluation mode imposes no limitations on the features of ClickRepair. However, you will be reminded of the number of days remaining each time you start the program.

ClickRepairRT also has a 21 day trial period, but no registration mechanism. It will be registered automatically once ClickRepair is registered; a single license suffices for the pair of applications.

License costs depend on the method of purchase. Click the “Purchase” button, or follow the link from the home page: www.kagi.com/clickrepair/license (under the “Software” tab), for further details.

Some users, for example residents of many European countries, may be charged applicable tax (VAT, TVA, etc.) at the relevant rate for their state/country of residence.

The program is licensed to an individual user. If you give it to a friend (for evaluation – please do!) they should buy their own license key to continue using it.

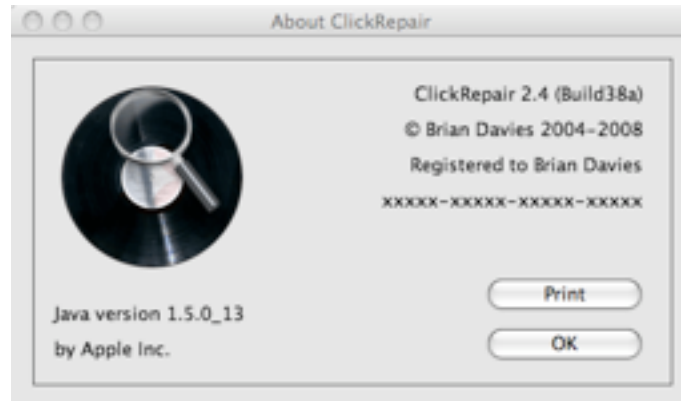


When purchasing the key, you will be asked for your name – minimum six characters. This information is encoded into the license key, and you will need it again, in precisely the same form, to register the program. (Note that it is not case sensitive – John Brown, JOHN BROWN, and john brown are equivalent.) The information should be emailed automatically by Kagi, to the email address you gave, once the transaction is approved.

Access to Registration Information

Immediately after registering ClickRepair, the “About” box will appear, as shown below. Notice that there is print button – print the information immediately and keep it in a safe place for the inevitable and unexpected day when it is required. The information is always available by choosing “About ClickRepair” in the menus, and can always be printed from there, but only so long as the computer is functioning!

On the first couple of launches after registering, the About Box can only be dismissed either by using the “Print” button or by waiting 10 seconds, as a positive encouragement to do this.



Users may also access the information via the “Registration” item on the “Help” Menu. The resulting dialog is shown below. The example is for a French user, whose name had an accent that was mangled by the Kagi website (Sébastien was SA(c)bastien). Although the user name may not be changed in a way which disagrees with the license code, accented characters are recognized and used by ClickRepair when making this check. Note in the example below that the user has entered the correct characters in the ClickRepair registration despite the incorrect Kagi website translation. Also, upper- and lower-case letters may be interchanged.



The information in this dialog may be saved into a text document using cut and paste, as an alternative to using the “About” box to print it.

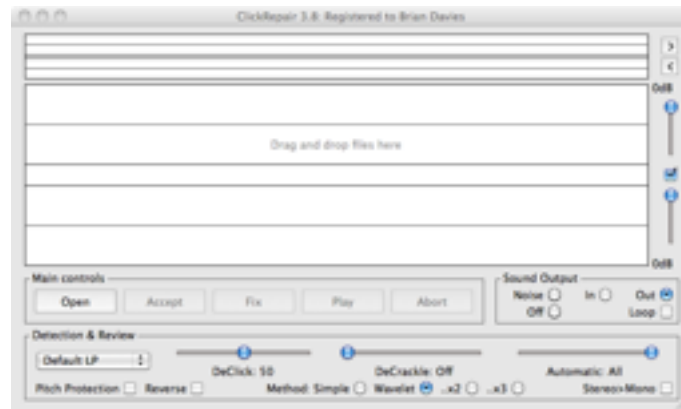
Important Notice

I do not have a license retrieval system. It is the responsibility of the user to print, and keep, a copy of the registration information.

3 Quick Start

Initial Interface

On starting ClickRepair, the default window has the appearance shown below. There are controls that affect the way that damage is detected, reviewed, and repaired, and windows for reviewing and adjusting repairs. When the program is started for the first time the defaults are set for automatic processing of vinyl (LP) stereo records in reasonable condition (Default LP), as here. Initially, the display window is blank, except for a message that it can be used to drag and drop a file for processing



Basic Controls

ClickRepair does not rely on menus for the most important operations, just buttons, check-boxes, and sliders. The more important functions have keyboard shortcuts, which match the menu shortcut keys. Shortcut keys may be displayed in a separate window, accessible from the “Help” menu. For example, the first step in using the application requires a file to be chosen for processing. The alternatives are:

- Drag and drop drop one or more files.
- Select “Open” from the File menu.
- Use the menu shortcut key + “O” to make this menu selection.
- Click the “Open” button using the mouse.
- Press the “Return” key – see Chapter 5 for details of how to find these shortcuts.³

Drag and drop is new since ClickRepair 3.6. If only one file is dropped, it becomes the input file, for immediate processing and the output file – a new file with a different name – is automatically created. The convention for naming this file, and the choice of folder where it goes, is determined on the basis of the preference settings. A dialog will appear showing the full path name of the out-

³ Unlike menus, this last action is not coupled with using a menu shortcut key. A complete list of keyboard shortcuts is given in a later chapter; menu shortcuts may be read off directly from the menus.

put file. If more than one file is dropped, the “Batch Setup” dialog is opened and the files are added using the normal rules for batch processing. See Chapter 12 for details of how this works.

If you want complete control over the naming and destination of the output file, do not use drag and drop – use the Open command. After a dialog allowing the input file to be selected, there will be a second dialog allowing the output file to be named and placed.

Setting of the sliders and check-boxes may be done individually, or collectively by using the “Settings” menu or the drop-down list to access the pre-defined and custom settings. This is all described in detail in later chapters.

Choosing Files

Before you can go on, it is necessary to decide how to organise input and output files. ClickRepair operates by opening the (input) file to be processed, as read only, together with a second (output) file to which the repaired audio is written. The files must be uncompressed, mono or stereo, in 16, 24, or 32-bit “AIFF” or “WAVE” format. The output file will be in the same format as the input file. More details of file formats are given in the next chapter.

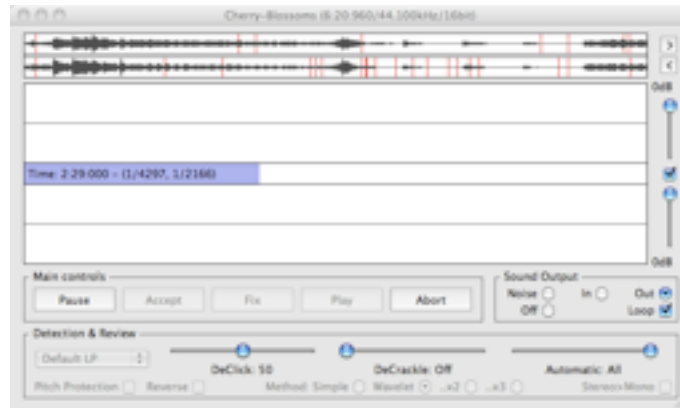
To choose the files using standard file dialogs, click the “Open” button, and choose the input file. Once this is done, a second dialog will appear for naming the output file.

The two files may not be the same file. This means that either they must reside in different folders (directories), or they must have different names.

ClickRepair has two modes, controlled by the “Append -cr” check-box in the Preferences. If this option is selected (the default), the save dialog will assume that you want to use this when naming the output file in relation to the input file, i.e., “beatles.aiff” becomes “beatles-cr.aiff”. If the option is de-selected, the save dialog will suggest that the output file have the same name as the input file, and ClickRepair will refuse to save it in the same folder (with the suggested name).

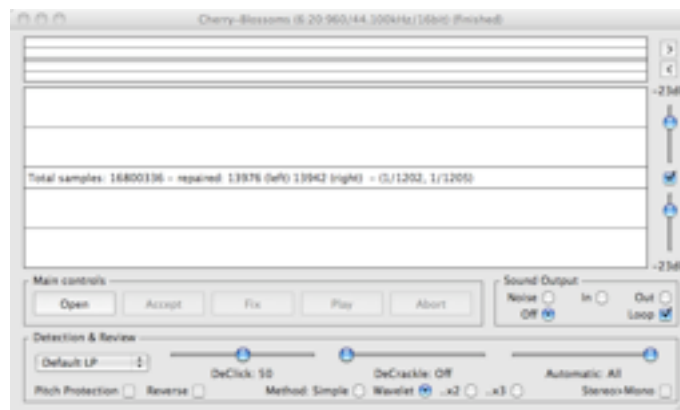
Each time you choose an input file, the dialog will go back to the folder previously selected (and, in most cases, to the previously selected file in that folder). Each time you choose a location for an output file, it will go back to the previously chosen folder either for input or output, depending again on a choice which may be set in the Preferences. Of course, you are free to make changes in the dialogs themselves.

After the files are open, the window title will change to indicate the file name, total length, the sampling rate, and the bit size. The “Open” button will change to a “Start” button. Click “Start” to begin.

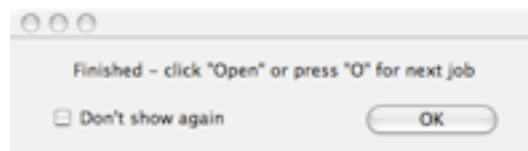


Progress in Automatic Mode

With the default settings, processing will continue unabated. A short segment of the file (524,288 samples – about 12 seconds) is resident in computer memory (RAM) at any given time; as new sound samples are read in, repaired samples are written out to make room for them. In this manual, this data cache is called the “sound buffer”. You can observe the state of this buffer in the upper display; there is also a progress indicator bar in the lower display. For more information on the sound buffer and its use, consult Chapter 7.



At the end, but only after the repaired file is completely written out, a dialog box will appear and statistics will be displayed. The program is then ready to process the next file. Notice that, in the example shown here, less than 1 in 1000 of the samples have been changed. (Information displayed in the progress bar – see Chapter 5 for further information.)



As with many of the dialogs, there is a “Don’t show again” check-box. I recommend that you do not turn off these dialogs until you are proficient in use of the program.

4 Producing and Organising Files

File Requirements

Before you can repair audio files, they must first be captured. This is done by playing the record into a suitable sound editing program via a sound card or other analog to digital converter, to capture the data as a digital stream. There are a number of useful resources for learning about this, for example www.delback.co.uk/lp-cdr.htm and www.a-reny.com/iexplorer/restoration.html. Both give thorough coverage of the process, from connecting the sound equipment to the computer and capturing the recording to preparing and burning the CD. The files must be uncompressed mono or stereo, in either “AIFF” or “WAVE” format. Acceptable sample sizes are 16, 24, or 32-bit integer, (aiff, wav type 1) and normalized 32-bit float (aifc type fl32, wav type 3). Be aware, however, that internal processing is 25 bit integer in all cases. The output file will automatically be in the same format as the input file.

The signal from the turntable equipment itself is totally unsuitable for conversion to a normal digital CD. This is because records are produced with a heavy emphasis on the high frequencies, and de-emphasis on the low frequencies, and must be played back through a suitable equalizer. This equalizer de-emphasizes the high frequencies and re-emphasizes the low frequencies. This is the reason why a good quality vinyl LP has relatively little “hiss”; equally, it is the reason why “hum” and “rumble” can be difficult problems to deal with.

For shellac (78) records, I recommend some careful research; one place to start is Roger Wilmut’s page www.rfwilmut.clara.net/

The most likely choice is to connect the turntable equipment to the phono input of your music system, and capture the output from auxiliary out sockets into the computer. In addition, there are other solutions for sound input, such as purchasing a dedicated preamplifier, or a USB sound card system.

ClickRepair is not an application for capturing and editing sound files. There are many possibilities for that, depending on the platform, and your budget.

If you already have some files which have been captured and burned to CD, see below.

Recommended Order of Processing

After capturing the sound to file, you may want to perform some or all of the following:

- Remove subsonic and low-frequency noise.⁴
- Remove the clicks and crackle.
- Reduce continuous noise, such as hiss.

⁴ DeNoiseLF, part of the DeNoise package on my website, is designed for this purpose.

- Separate the tracks of the record into distinct sound files, so they can be navigated on the CD.
- Fade the beginning and end of each track before burning the CD.

All of the processes, except for click and crackle removal, will require other software. Hiss and rumble reduction may be carried out using the “DeNoise” and “DeNoiseLF” applications, which may be found on the ClickRepair website; general editing functions, such as audio capture, separating and fading tracks, etc., will require an audio editing program.

Some users have questioned the order recommended here, given that there are other commercially available packages which recommend general noise reduction as the first step. The reasons for removing the clicks and crackle first, when using ClickRepair, are simple:

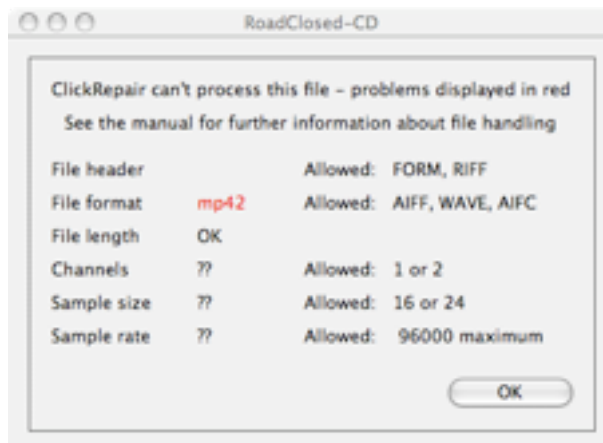
- If left in, large clicks can significantly upset any continuous noise reduction process.
- Clicks and crackle are partially removed by filtering, leaving them audible, but harder to find.
- ClickRepair is highly non-linear, whereas most digital audio processing is linear.

The most obvious way in which ClickRepair is non-linear is that it will process the same section of the file more than once if clicks are found, but it might only process other sections once. But ClickRepair is non-linear in a more fundamental way, which is impossible to describe in a few lines, or without the use of advanced mathematics. For readers with any knowledge of statistics, it relates to the fact that clicks do not fit the pattern of a “normal” statistical distribution – in fact they are a total misfit.

One other word of warning. Before subjecting files to subsequent noise reduction, listen to historical recordings that have been professionally restored. You will find that some of the hiss and other continuous noise is still there! This is because such noise cannot be completely removed without inflicting even greater damage in terms of loss of quality and presence of the original signal. So – tread carefully!

Notes about File Structure

There are many types of sound files. They are generally organised in “chunks”, with small “header chunks” indicating the actual structure and type. If you attempt to open a file which ClickRepair is not designed to accept, a dialog will appear, giving relevant, if somewhat technical, information. An example of such a dialog is shown here. The most likely reasons for rejection of a file are:



- The file uses something other than 16, 24 or 32 bit (integer) sample type, or 32 bit normalized float. Sample types not supported include 8 bit integer, and float formats other than 32 bit normalized.
- The file uses some form of compression. Note that WAVE files can use various compression formats, whereas AIFF files are uncompressed. Other file types (MP3, AAC, ...) are also unacceptable.

Uncompressed sound files are huge – about 10MB for each minute of CD quality data. It follows that most, but not all, of the file is the sound data itself. This is the only part of the file to which ClickRepair will make any changes. In particular:

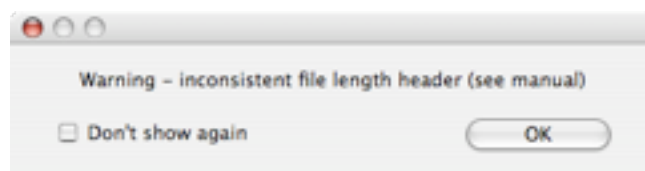
- All chunks which occur before the sound data are copied to the new file verbatim.
- All data which occurs after the sound data is copied verbatim, without any format checking.
- File name extensions are copied from the input file, so as not to change file associations.
- On Mac OS X, the file type and creator, which is used by the Finder in determining file associations, are also copied.

File Lengths

The maximum file length acceptable to ClickRepair depends on the file format and is determined by the standard formats for WAV and AIFF files, together with what may be indexed using 32-bit integers. For WAV files the maximum length is about 2GB (actually 2,147,483,647 bytes), for AIFF files the limit is 2,147,483,647 audio samples. For 24-bit stereo, this amounts to about 12GB. Attempts to open longer files will result in an error dialog. For stereo files in 32-bit/96kHz resolution, 2GB is a little over 46 minutes. At a resolution of 24/96 it is about 62 minutes; at a resolution of 16/44.1kHz it is over 3 hours.

Some Windows Audio Editors can open longer WAV files, using proprietary file formats and/or meta-data files. ClickRepair is not able to handle these special file formats.

ClickRepair has one other dialog related to file-length issues, shown here. It is a warning that the file length given in the header disagrees with the actual (physical) file length on disk. This alert is provided for information only – in most cases it will not affect the use of the file, either by ClickRepair, or by other audio editing programs.



In more detail:

- Provided that the file is at least as long as the sum of its required chunks, this error will not affect processing, and there will be no problem in proceeding.
- If the file is incomplete, this warning will be followed by the larger error alert above, with the number of missing bytes stated. Processing will not be possible in this case.
- ClickRepair does not change these (or any other) headers, even if they are inconsistent.

Sample Sizes

The sound data used for standard CDs is 16-bit, meaning that each sample is a whole number (integer) in the range -32768 to +32767. For 24-bit data, the range is -8388608 to +8388607. It is important, when processing CD sound data, to adjust the levels so that the loudest passages do not exceed the maximum level – saturation renders the audio virtually unusable. In ClickRepair, audio level is measured relative to this critical saturation level, which is labeled 0dB. A signal which leaves one data bit of headroom has a level of -6dB, a signal leaving two bits of headroom has a level of -12dB, etc.

Audio levels are not measured in numbers and fractions, rather in decibels (dB). This is a scale for measuring multiplicative increase and decrease in an additive way – a logarithmic scale. An audio level of -6dB corresponds to one-half the saturation level, -12dB corresponds to one-quarter, -18dB to one-eighth, etc.

Note that a signal whose amplitude is reduced by one-half has its power reduced by one-quarter – power is proportional to the square of the amplitude. This is why half power is -3dB while half amplitude is -6dB. In the ClickRepair display, we are viewing the amplitude.

In considering what 16-bit audio quality means, the following comments may help. In a 16-bit number, the 16th bit is used to indicate whether the data is positive or negative. The lowest bit contains as much quantization error as signal, since 100.49 might be recorded as 100, 100.51 as 101, magnifying the difference. Therefore the maximum signal-to-noise ratio attainable on a standard CD is about 87 dB (14 1/2 bits), quite adequate for capturing a record, provided that the transfer is not made at too

low a level. But, if the transfer is made at a level which never peaks (for example) above -12dB, 2 bits of the available digital data is unused, and 12dB of the available signal to noise capacity is lost.

Internal processing uses 25-bit integer representation,⁵ regardless of whether the data files are 16, 24, or 32-bit, integer or floating. Capturing and processing in 24-bit format give greater flexibility for editing the audio, even if the data is finally truncated to 16-bit for burning CDs; indeed this is the format widely used by professions for audio editing and production (usually at 96 kHz as well). Note, moreover, that the resolution and dynamic range of audio captured as 16-bit data is not improved by subsequent conversion to 24 or 32-bit formats – for this it is necessary to perform the original analogue to digital conversion using hardware which provides 24-bit.

Sample Rates

As mentioned previously, ClickRepair is designed to work with audio files at sample rates of 44.1 kHz (CD) and 48 kHz (DAT). Simply put, click removal involves modeling the musical signal as accurately as possible, and one critical factor in the process is the relation of the sample rate to the actual time scales. For full-frequency range material, working at a lower sample rate than 44.1 kHz is obviously counter-productive since the CD rate is an absolute bare minimum for high quality sound reproduction, and resampling is not a recommended extra procedure. (Some comments on special techniques for Shellac records may be found at the end of Chapter 9.)

Given 88.2 kHz or 96 kHz data, the actual processing is performed on two parallel data streams, each at half the sample rate, using wavelets to perform exact reconstruction re-sampling. This means that the output file will exactly match the input file, sample-by-sample, except for those samples which are repaired. The technique is in no way equivalent to using normal audio software to down-sample the original 96 kHz file to 48 kHz. After that process, up-sampling to 96 kHz cannot restore what was lost in the down-sampling.

Wavelets are a method of data analysis which combines frequency and time information in a sophisticated way. This differs from Fourier analysis, which treats signals in blocks, transforming each block of samples, which are organised sequentially in time, in to a block of the same length which is organised sequentially by frequency content. In the form used by ClickRepair, a wavelet transform takes an incoming data stream and separates it into two streams, containing the high- and low-frequency components, with the extra property that each stream is sampled at half of the original rate. The magic of the transforms employed here is that when the two streams are subsequently merged, the reconstruction is exact. This is just as well, since the high- and low-frequency separation does not have a steep cut-off. Wavelets are used in other components of ClickRepair, as described in later chapters.

The maximum sample rate is 192 kHz, which triggers an extra layer of wavelet decomposition and reconstitution. In this case no repairs are made to the high-frequency stream since it contains no audio information useable for this purpose. Nevertheless, the original data is preserved exactly.

⁵ Effectively 24-bit audio – the 25th bit is used to achieve perfect wavelet reconstruction.

Reading from CDs

Some users may have previously captured records and burnt them to CD, and now wish to de-click them. Obviously it would be convenient to process such files directly from the CD, without first extracting the tracks.

The sound data on a standard CD is 16 bit stereo, sampled at the rate of 44.1 kHz. The physical format of the data is such that it has to be “ripped” to recover a valid computer sound file. On Mac OS X, each track on a CD is presented, by the Finder, as a file of type “AIFC”, whose encoding is of type “sowt”. This is a standard file type, and since the operating system is doing the “ripping”, the data can be read directly by ClickRepair. Of course, data cannot be written back to the CD, but since ClickRepair will not edit the original file, it is possible to process directly from it.

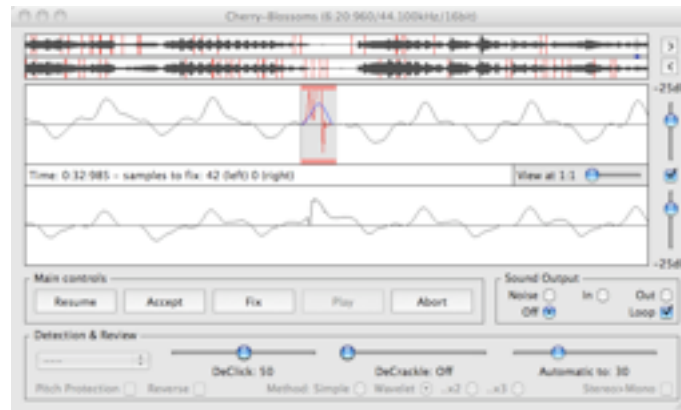
It has been explained already that ClickRepair makes no changes to any part of an audio file, except for the sound data itself. Therefore, files processed directly from CD will be saved with the same headers and file extension as were passed in by operating system. You will be able to open these new files with any program which is able to deal with the original file directly.

If you want to test this, drag one of the tracks from the CD, to the desktop, using the Finder. This will copy it. Now attempt to open it with your Audio Editor or other application. If it works for the original, it should work for the file produced by ClickRepair.

5 User Interface Basics

Visual Interface

Once you have tried the program, I recommend making some experiments, setting the “Automatic” slider to “Off”, or at least to a small number. This will cause the program to stop, before making any repair whose length is greater than the chosen maximum, enabling manual intervention.



In the figure here, the program stopped because 42 consecutive samples in the left channel are proposed for repair, more than the “Automatic” setting of 30. The features of the user interface may now be seen. Notice that:

- The “Start” button reads “Resume”, because processing has paused.
- There are two views displayed, the top one long and narrow. Each shows the left and right channel, with the left channel over the right, which is standard for Audio Editors.
- There are buttons for other actions, particularly “Accept” and “Fix”.
- A horizontal scale slider is now visible – this is linked with the “View” menu.

The Sound Bar

The upper (smaller) view displays the contents of the sound buffer – by default about 12 seconds in length. It shows where repairs have been made, as vertical red lines, and may be used to navigate around, and make further changes to, samples still in the sound buffer. There is also a narrow central strip, in which a small blue indicator shows what segment of the sound buffer is being displayed in the lower view. The two buttons to the right of the sound bar scroll the samples displayed in the lower view, and this is also indicated by movement of the indicator.

I shall refer to this upper view as the “sound bar”, since it has some similarity with the horizontal scroll bar of a word or image processing program. The indicator shows how the samples displayed in the lower view relate to the totality of samples resident in the sound buffer. Moreover, clicking in the sound bar, or in the adjacent right and left shift buttons, causes the lower display to scroll.

The Display Window

The lower view is used for examining, making, and editing repairs. I call it the “display window”, although it does more than display sound data (and technically, it is not a window). It is linked with three scaling controls (two vertical sliders and one horizontal slider).

In the simplest case, the samples to be repaired will be found automatically, and a repair constructed automatically, as in the figure. The highlighted samples have been marked for repair – the detected bad samples in red, the proposed repair in blue. Because the channels are processed independently in stereo mode, the selection is different between left (top) and right (bottom).

The central strip of the display window shows the time of the first bad sample, and the extent of the proposed repairs. When the program is running automatically, this central strip serves as a progress indicator and displays repair statistics of the sound buffer as processing proceeds. Finally, when processing is complete, it displays useful statistics, already seen.

If the program stops at a marker, or is stopped using the “Pause” button, the central strip displays this fact, together with the time, and the name of the marker, if appropriate.

The Scale Controls

To the right of the display window, there are two slider controls, with a check-box in between. The sliders control the vertical scaling employed in the display window. Normally, this is chosen automatically, and is determined by peak levels, and indicated by the state of the check-box. When the check-box is selected, vertical scaling is automatic. Moving the vertical sliders automatically deselects the check-box. Automatic scaling is restored once you resume processing, or you can reselect the check-box. To magnify the display, move the slider down, which reduces the display peak level. This might seem counter-intuitive, but is a consequence of showing the peak level.

ClickRepair scales the display window automatically, usually the most convenient scheme. To reveal as much detail as possible, damage is not included in setting the level. Peak levels are displayed in decibels (dB), just above and below the sliders. The sliders also move down as the peak level falls. Information about sample sizes and peak levels was given in the “Sample Sizes” section in Chapter 4. Generally, there is no need to worry about these controls. Small clicks in a quiet passage might be quite disturbing, yet the same clicks in a loud passage might not be noticed. This is intuitive when the display is scaled automatically, and the click might even extend off the display (in red).

The horizontal scale used in the display window is controlled by the “View” menu, and also by a small horizontal slider, visible in the central strip of the display window whenever processing is paused. You can use either for changing the scale, but only when processing is paused.

The effect of increasing the ratio is that more samples are displayed across the display window, at the expense of having a less detailed view. The scales progress in powers of 2, starting with the

most detailed “View at 1:1”, then “View at 2:1”, up to “View at 256:1”. The viewing scale also determines how many sound samples are displayed per pixel across the display window.

As the scale increases, the fraction of the sound buffer which is displayed increases. The indicator of the sound bar changes to show this, becoming progressively wider.

For manual inspection and adjustment of repairs, it is advisable to use the most detailed scale – 1:1.

Menus

On all platforms, there are six menus, with menu items as given here:

- File – Batch Setup, Breakpoints, File Dialogs, Open, Start, Pause, Abort.
- Edit – Undo, Fix, Accept.
- Settings – Default LP, Default Brass, Default Percussion, Default Mono, Default 78, Import, Export, [saved settings, if any], Custom.
- Sound – Play/Stop, Off (faster), Removed Noise, In (Original), Out (Repaired), Loop, Mixer Settings.
- View – from 1:1 to 265:1.
- Help – Registration, Website, Keyboard Shortcuts.

On Mac OS X, the Application menu has the items:

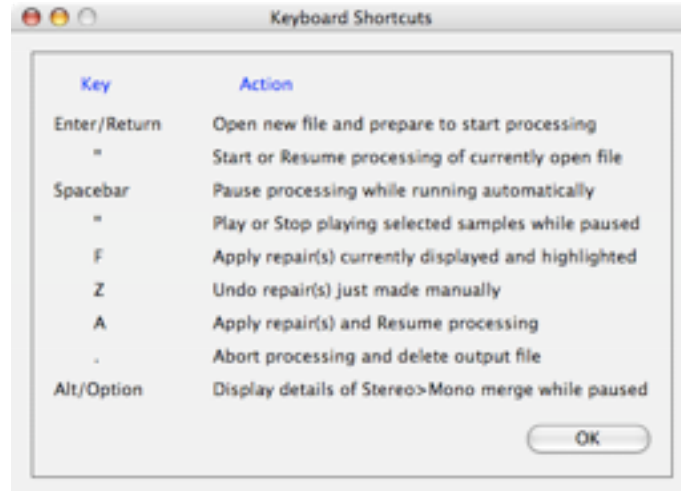
- Quit ClickRepair.
- About ClickRepair.
- Preferences.

On other platforms, these three appear as extra items on the File and Help menus.

Even if you are using the in-window controls (buttons, etc.) and/or keyboard shortcuts most of the time, the menus serve to provide extra information where there is insufficient room for this on screen. For example, the Edit>Accept item changes to Edit>Accept (Fix & Resume) when there is a repair waiting to be made. Similarly, the Sound output menu items contain more explanation of the four choices which are possible.

Keyboard Shortcuts

A complete list of keyboard shortcuts (not to be confused with the usual menu-shortcut keys associated with the command/control key) may be viewed by selecting this item on the “Help” menu.

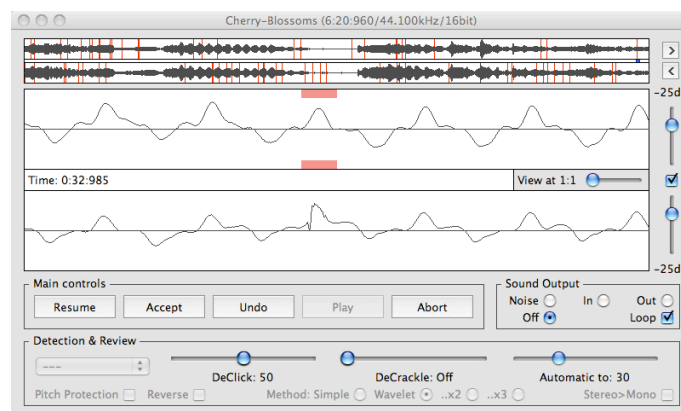


The resulting window is shown here. It is not a modal dialog and may be left open during normal operations, for ready reference, although it can be rather obtrusive.

Making a Repair

Once a set of samples has been identified for repair, the surrounding samples are analyzed for content, and a repair is constructed.

In the example shown at the start of this chapter, it is evident that the damage should be repaired; this is done by clicking the “Fix” button. On doing this, the display changes its appearance somewhat (see next figure); only the repaired samples are displayed, now in black, but there are light red rectangles above and below, marking the fact that repairs were made. In addition, the button itself changes to “Undo”.



The repair may be undone immediately by clicking the “Undo” button, after which the same button (which reverts to “Fix”) may be used to reapply it immediately. This is the simplest mode for making and/or undoing repairs.

After making a repair, click “Resume” to continue. The button will then display “Pause” until it is clicked again, or until processing stops again. A shorter alternative to clicking “Fix” followed by “Resume” is to use the “Accept” button; it applies the fix and resumes with a single action.

Of course, if you decide immediately after doing this, that it may not have been the best decision, it is easy to pause, and then use the sound bar to redo that (or any other) repair, provided it is still resident in the sound buffer. Information on how to navigate the sound buffer is given in a later chapter.

Audio Monitoring

Radio buttons in the “Sound Output” control group allow you to turn sound output on or off. Monitoring, particularly with a pair of headphones, or reasonable speakers, can provide invaluable feedback. These buttons may also be controlled from the menus.

Audio monitoring causes processing to be real time, which will be much slower for computers with even moderate processing power, when operating on files in reasonably good condition.

- Selecting “Out”, the default, the repaired file will be played as it is processed, which is some time after it is read into the sound buffer from file.
- Selecting “In” will send the unprocessed data to the audio system, which can be useful for “A–B” comparison, particularly when used with the “Play” button (detailed instructions are given later).
- Selecting “Noise” will enable you to monitor what is being removed by the program.

There should be no hint of the musical content; this would indicate that the settings are too severe, causing too many false detections. Instruments such as cymbals and brass, and certain types of singing, are among the more difficult music types, and may give rise to such problems.

- The sound can be turned off, removing the restriction to real-time processing. This is the fastest possible mode, and the most convenient provided you are confident of the result.

Sound Latency

Given that the computational delays are extremely variable, processed data is not sent to the sound system directly, rather to a second buffer, which then feeds the sound system. If this buffer empties, then the sound is broken up, as if the computer were stuttering.

The algorithms use techniques from advanced digital signal processing theory, and can be extremely taxing on processing power. I have attempted to make the coding as efficient as possible, but the algorithms have been chosen for their effectiveness more than their speed. Turning the sound off may be

the only viable setting with slower processors, in which case it is still possible to play the contents of the buffer from time to time, because the repairs have already been made.

The upshot of all this latency is that, if you hear a click, or other undesirable artifact, it will already be about 1–2 seconds in the past by the time you can click “Pause”, and the program can respond. Since the (main) sound buffer – the buffer displayed in the sound bar – is longer than this, the click is still accessible for repair. In order to go back and fix the problem, you will therefore have to navigate back, as described later.

It would be more convenient to go automatically to the exact point at which the problem was actually heard, in response to clicking the “Pause” button. This would involve much tighter synchronization of the processing loop with the user interface, resulting in a significant diversion of processor power to this task.

The Progress Indicator

The program runs without pausing while it is making repairs automatically, during which times the progress bar gives useful feedback, as follows:

- The blue bar shows graphically what fraction of the file has been processed so far.
- The current processing point is displayed in the form minutes:seconds:milliseconds.
- Statistics are shown in parentheses. They are the average fraction, of samples currently in the sound buffer, which have been repaired. (Left and right displayed separately.)

The repair rate depends on numerous factors, and there can be no simple rule. However, I offer the following suggestions when choosing sensitivity settings, as described in the next chapter:

- For a vinyl record, repairing more than 1 in 200 samples should be viewed with suspicion, although it might lead to results that are more acceptable.
- Unless your records are in really good condition, it is unlikely that the repair rate will fall below about 1 in 1000–2000 samples.
- For 78 rpm (shellac) records, anything goes if the result is an improvement in the perceived audio quality. Don't expect miracles!

6 Detection and Repair Settings

The “Detection and Repair” group contains controls with the following functions (explanations form the content of this chapter and Chapter 10):

- The DeClick and DeCrackle sliders set the sensitivity for detection of clicks and crackle.
- The Automatic slider sets the extent to which repairs are fully automatic.
- The “Pitch Protection” check-box turns on an algorithm which avoids periodic false repairs to the hard edge of high-energy pitched sound, such as brass instruments and some voice.
- The “Method” radio buttons give a choice of single- or dual-stream (wavelet) processing.
- The “Stereo” radio button is for normal stereo LP records. Do not select “Stereo>Mono” before reading Chapter 9.
- The drop-down list provides the convenience of saving and recalling combinations of settings.

It should be evident that there are a multitude of parameters which affect the processes of detection and repair, and it is tempting to allow these to be adjustable. This would be counter-productive without complete technical documentation of the mathematical algorithms, and the factors which affect their operation, in the hands of a technically expert user. During testing and development of the program, it became clear that less would be better than more; moreover I have read reviews of some audio repair software which complain about a plethora of obscure settings. So there are very few user-defined settings.

This chapter explains each of the settings in turn, and also how groups of settings may be saved with custom names, and then reloaded for reuse. First, however, it is important to understand that most of the detection parameters are dynamically generated by the algorithms built in to ClickRepair, and that the two sensitivity sliders, in particular, provide user input.

The numbers are arbitrary, in the range 0-100. A setting of 50 does not mean “remove 50%” – it is just less sensitive than a setting of 60. Moving the slider fully left switches the function “Off”.

Usually, suitable settings will be found by experimentation. For example, if you are trying to clean up a rather old record, you might find, by listening to the output, that higher sensitivity gives a more acceptable result, and, by listening to the noise, that the music seems unaffected. Another record of the same vintage, but a different genre, may exhibit brief bursts of musical tones in the noise output, and you will have a more difficult task in deciding on compromise sensitivity settings.

I have provided four defaults, “Default LP”, “Default Brass”, “Default Mono”, and “Default 78”. These are intended to serve two purposes:

- A quick fix for the impatient.
- A starting point for your own experimentation.

The DeClick Control

The most important parameters for click removal are set through the “DeClick” slider, which determines the sensitivity for the detection of clicks.

One of the factors which influences the detection of clicks, as opposed to crackle (see below), is that click detection is made on the basis of looking at short segments of consecutive bad samples. One really bad sample can trigger a click detection, but samples which are less obviously outliers will be analyzed as small groups – several consecutive samples which are moderately out-of-line may trigger detection.

It is impossible to choose a sensitivity which will see all clicks removed, no matter how small, with no false detections. The default values have been chosen to work well on many types of music which have been captured from records that are in a reasonable state of repair.

- Much of the time lower settings will remove all audible clicks.
- It may be possible to increase the sensitivity without doing significant damage, but this is not recommended unless there is a need.
- If the sensitivity is set too high, sudden changes in musical content may cause false detection, leading to increased incidence of undesirable false repairs.
- If the sensitivity is set too low, there is increased probability of not detecting audible clicks.

The bottom line is that the selection of the sensitivity is a trade off between click removal and musical degradation. The final decision will be governed by a number of factors, including:

- The degree of damage – older records may benefit from settings of 60–80 (or more).
- Type of music – some vocals are particularly difficult and may need settings of 30–40 (or less).
- Percussive music could be even more difficult and might need even lower settings.

The de-click algorithm is iterative – places where clicks are found are treated at least twice, sometimes more than that. Normally there should be no advantage in de-clicking the file a second time.

The DeCrackle Control

The detection/repair algorithms used for click removal are not particularly attuned to the removal of very short (1–3 sample), rapidly repeated, small clicks, which are usually heard as “crackle” or “buzz” (not “hiss”). This kind of damage is particularly prevalent with shellac (78) recordings.

Selecting this option switches on a “post process” which examines every sample individually and adjusts those which are sufficiently out of line. Click removal involves a more collective decision making process, making it more likely that small clicks could be overlooked when they are closely spaced. Note that:

- The sensitivity is set by the “DeCrackle” slider, which works in the same way as the “DeClick” slider.

- The default values are just suggestions, and not meant to replace personal experimentation.
- Be aware that aggressive de-crackling can take the edge off voice, cymbals, etc.

This feature should normally be used if click removal seems unable to deal with some forms of crackle. However, shellac (78) records usually have a great deal of crackle, which is why “Default 78” has “DeCrackle” set to 50.

All repairs made this way are indicated, in the normal manner, in the sound bar and display. They are distinguished by the use of a lighter red, which is useful feedback when treating shellac (78) records with relatively high de-crackle settings. Moreover, it is still possible to undo/redo repairs made this way, although that is not really a practical option.

Files with a significant amount of crackle may benefit from re-processing the output file a second time, perhaps with de-clicking and pitch protection switched off.

The Automatic Control

This controls the degree of automation. When moved fully to the left, no repairs will be made without manual supervision, when moved fully to the right, all repairs will be made automatically. The example shown in the previous chapter was at the setting of 30 – repairs up to 30 samples in length were carried out automatically, whereas repairs longer than this cause processing to halt.

When first using ClickRepair it may be a wise precaution to start with this control set to “none”. This will result in extremely slow progress, but will help in checking that the detection settings are chosen so as to maximize the ratio of desirable repairs to undesirable false detections.

Processing in semi-automatic mode will be quicker, with the control set to repair 30–60 samples automatically. There is a visual indication of the extent of automatic repairs to be seen in the sound bar.

In the end, the aim is to find reasonable settings, and then to let the program run in fully, or nearly fully, automatic mode. Clicking “Pause” within a few seconds of hearing a problem will still give the opportunity to go back and attempt to fix it manually.

Method – Simple versus Wavelet

It is clear from the interface that there are more methods available – x2 and x3. These more complex methods are the subject Chapter 10 and should not be used initially.

“Simple” means that each channel is processed as a single data stream of sound samples taken at the given sampling rate. This is the simplest and most efficient choice, and in some cases it is perfectly adequate, even beneficial. However, there can be significant advantages from splitting the signal into two streams, separating the high and low-frequency components, by selecting the “Wavelet” method. The theory of wavelets permits this to be combined with lossless down-

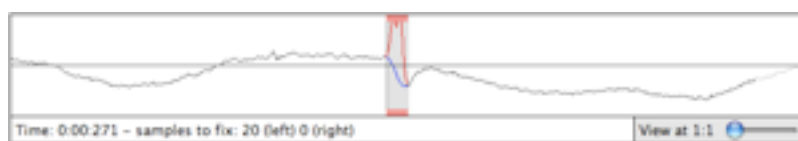
sampling, so that each of the two filtered streams is down-sampled to one-half of the given sampling rate. As a result of the down-sampling, processing the two streams is only a little slower than processing one, unless the repair rate is high.

I already explained, in Chapter 4, that wavelets are not equivalent to conventional filtering coupled with conventional re-sampling. Rather, the theory allows for perfect reconstruction of the data stream if no repairs are made in the wavelet streams. For records in good condition, this exceeds 99.5% of the file.

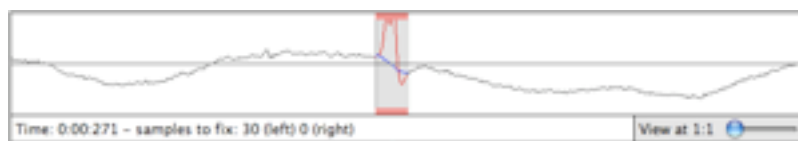
The question arises, which method to use. Here are some observations:

- The “exact reconstruction” algorithm means that a repair to just one sample of an underlying stream is spread over a minimum of four samples in the output stream. This implies that a bigger fraction of the original samples will be changed.
- On the other hand, wavelet processing will smooth small defects, so the actual modifications made to a smaller number of samples in a single stream are likely to carry at least as much energy as in the modifications made by the wavelet algorithm.
- The pitch protection and merge algorithms (see later) are able to use a longer time-scale on a down-sampled stream, which is beneficial.

To further assist in understanding the differences, here are two pictures showing Simple (top), and Wavelet (bottom), both at the default sensitivity of 50, in automatic operation on the same glitch.



Clearly, the wavelet method is better at these settings, although the Simple method does achieve an almost identical result if the sensitivity is increased to 60. The problem is that the higher sensitivity will almost certainly increase the number of false detections.



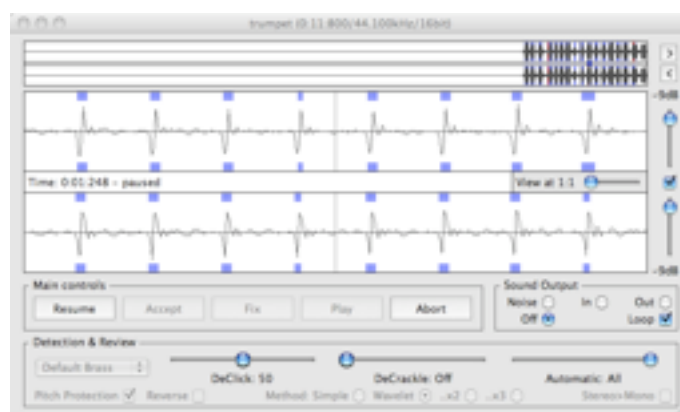
Here are some recommendations:

- For records in only moderate condition, and for shellac (78) records, “Wavelet” is a must.
- “Pitch Protection” works better in “Wavelet” mode, although it is available in “Simple” mode.
- For records in very good condition, and if highly pitched sound is not an issue, the “Simple” choice may be the most efficient.

Note that this check-box may not be changed while a file is being processed.

Pitch Protection

ClickRepair makes detection decisions based on quite short-term information. In fact, there are a number of conflicting factors involved in the decision making. As a result it is counter-productive to have the underlying algorithms look at correlation between repairs separated by more than a few score of samples. The problem arises that some music types, for example trumpet, have the appearance of a regular succession of clicks. The figure here shows just such an example of solo trumpet. The sharp pulses occur with perfect regularity for the duration of a single note – the rate at which they occur determines what note we hear.



Looking at the figure, notice that “Default Brass” settings are in use – this employs “Pitch Protection”. Every pulse has been identified as a potential click, which would lead to the destruction of the trumpet sound rather than the removal of noise. But they have also been identified as due to high-energy pitch, and no modifications made. This is indicated by the use of blue, rather than red, in the display and in the sound bar.

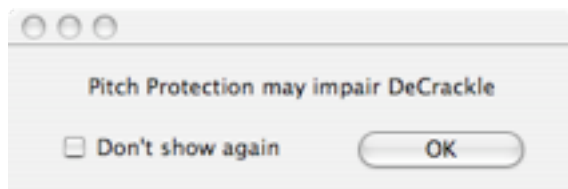
This check-box may be changed while a file is being processed (new in version 3.9).

When deciding whether or not to use “Pitch Protection”, consider:

- For a record in bad condition, there will be many repairs. The time required to look for longer-term correlation will increase exponentially, and processing may become painfully slow.
- Some records have periodic groove damage caused by stylus mis-tracking, exactly at the pitch of the sound. The results of periodic repair may be an improvement over the original.

Possible conflict between DeCrackle and Pitch Protection

Use of “Pitch Protection” together with “DeCrackle” may seriously impair de-crackling. This can happen on records with a lot of crackle, if some of it also has a distinct pitch. Whenever you turn on these two options together, a warning dialog will show as a reminder of the potential conflict.

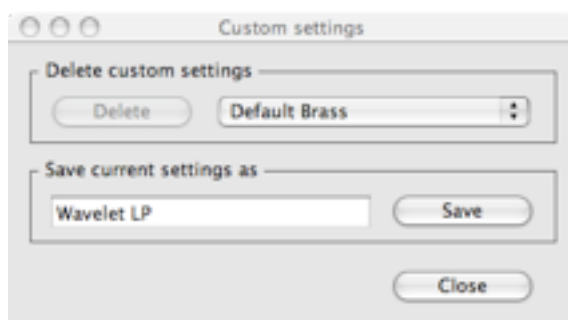


In some circumstances, if you want to use both options, it might be better to de-click first with “Pitch Protection” on, with or without de-crackling, then process the resulting file a second time with DeCrackle set to a suitably low value. In other circumstances, deploying the two simultaneously may not lead to any problem. Only experiment will tell where the balance lies between noise reduction and damage to the music itself.

Saving and Recalling Settings

Settings in the “Detection and Review” control group may be saved, with any chosen name (up to 32 characters in length), and recalled at a later time, using the drop-down list in the “Detection and Review” control group. I call this the “settings menu” in the following. All settings are remembered from one session to the next.

Note that the settings menu changes from the currently named settings to “---” as soon as manual adjustment are made to any of the settings. This is an indication that some settings have changed.



The settings menu comes with four presets, Default LP, Default Brass, Default Mono, Default 78. These are pre-programmed, and cannot be changed. If you have settings which you would like to add to the list, select “Custom...” on this list. A dialog will appear, shown above. Simply type in the name for your settings, then click “Save”. The current settings will then be saved with the chosen name. The Settings dialog may be used to manage the list, as follows:

- To add a set, type in the name and click “Save”.
- To change a set, type in a name which already exists, click “Save”, and confirm in the ensuing dialog that you want to update the settings of that name.
- To remove a set, select the unwanted settings in the drop-down list, click the “Delete” button, and confirm in the ensuing dialog.

It is important to note that the Settings dialog does not offer any possibility for adjusting the settings – its sole purpose is to save settings which are actually in use, and to manage that list. The only way to change an existing named setting is to save the new values with the same name. After the confirmation dialog, the old values will be replaced by the new ones.

The settings menu is disabled while a file is being processed.

The custom settings may also be saved to a file, using the “Export” item on the menu, which will result in a dialog allowing the file to be named, and its location chosen. Previously saved files may be opening using the “Import” item, and the settings they contain will be added to any existing custom settings.

Deciding on Appropriate Settings

A user wrote to me as follows:

“I ran an experiment whereby I processed the same track using different values of the DeClick slider - 10, 20, 30, 35, 40, 50 and the default 60. The program reported the number of fixes at the various settings. The number of fixes increases dramatically as the slider setting increases, but that should not be a surprise. Aurally, I found that at 30, a few isolated, faint clicks remained, but at 35 even these were gone. My untrained ear seems to detect, however, that as the slider setting increases, the music becomes very subtly duller and less bright. Therefore I'm thinking of adopting the strategy of using the lowest setting that removes all audible clicks. Is this a valid approach? Can I use the same setting for the same music genre or should I re-calibrate for each album/track?”

This may indeed be a good strategy, but beware of reducing the sensitivity too much. Half-removed clicks may sound like pops. You should visually examine some of the “false repairs” which are being made – they may be insignificant compared with the effect of half-removed real damage.

Since this is a time consuming activity, give the settings a name which will remind you of the “genre” and save them for future use. ClickRepair is a program for doing statistical analysis in a situation where there are no black and white answers. There is always a trade-off between removal of clicks and removal of desirable audio content.

7 Audio Monitoring and the Sound Buffer

Java Sound Output settings

Audio monitoring involves sending sound data to a sound data line which has been made available to Java applications. The number and functionality of such lines is determined jointly by the operating system and the Java run-time, and is outside the control of ClickRepair. Sometimes there is a choice of output lines; also these lines often have an associated “Master Gain” control to set the overall sound output level.

These features may be accessed from the “Mixer Settings” item on the “Sound” menu. There is a drop-down list, which will list all available output mixers, together with a “Master Gain” slider control, and a button marked “Play Test Tone”. Clicking on this button causes a three-second pure tone, at 440Hz and maximum allowable amplitude, to be sent to the selected line. If the gain is set too high, the tone will sound harsh, due to clipping. If it is set too low, the sound output from ClickRepair will be at too low a level.

I have chosen the default setting of +5dB, for the default Java Sound Audio Engine, to match what might be expected from typical applications such as Apple QuickTime or Windows Media player.

To change the settings, simply choose something which is suitable, then click the “OK” button.

The Sound Bar

The sound bar shows all the audio data which is still resident in the sound buffer. The default length of this buffer is 524288 samples, for files up to 48kHz – about 12 seconds in practice. This can be increased by a factor of 2x or 4x, to 24 or 48 seconds, using the “Preferences” dialog.

For files above 48kHz, the longest buffer setting may result in “Out of Memory” messages, since all workspaces get doubled or quadrupled in length – not just the sound buffer. Above 96kHz, sound output will be at half sample rate because of limitations with Java sound output.⁶

Remember that the sound bar represents more than 500,000 samples, so a single vertical red line represents typically 700 samples, and only one has to be repaired for the line to show as red. This means that it is possible for the sound bar to show a lot of red areas, without there being cause for alarm. Equally, there may be cause for alarm, which is why it is important to observe the statistics in the progress bar, and to do some investigation using the sound bar to navigate the sound buffer.

To review any part of the sound buffer, simply click on the vertical red line that marks the repair, and the display window will show the samples at about that place. If the point clicked is not exactly

⁶ Above 48kHz with Apple’s Java, because of a bug in their Sound implementation.

at the repair, which will be evident because the blue indicator does not coincide properly with the vertical red line, then use the right or left arrow keys to scroll the selection showing in the display.

First click “Pause” – the sound bar does not respond in automatic mode.

Finding the damage may require some detective work. There are two reasons why just examining the display can be unsatisfactory:

- Sometimes, quite a small “glitch” can lead to a surprisingly audible click.
- Sound files are not small, so it can be like looking for a needle in a haystack.



Listening to the Sound Buffer

Once processing has paused, the “Play” button may be clicked to play the selected contents of the sound buffer, but only after that selection has been made, as indicated by the grey section on the soundbar in the picture above. To select just a portion of the soundbar for playback, click and drag with the mouse. To select the whole of the sound buffer, double-click.

As explained above, a single click in the soundbar makes the display window scroll, for which reason a single click, followed by a shift-click, is not used as a selection mechanism for sound playback. This differs from a common behaviour pattern of many programs, but it avoids ambiguity in the meaning of a single click. Note that if a double-click is too slow, it will be treated as two single clicks.

There is a small blue indicator, in the central strip of the sound bar, that will track the sound as it is played. During this time, the button will display “Stop”. Notice that, if sound output is off, the processed sound is played as the default option. If you want to play back the original sound or the removed noise, select the appropriate radio buttons.

The shortcut for both “Play” and “Stop” is the spacebar. The “Loop” check-box results in an endless loop, until either you choose either to stop, or to resume processing. You may change the radio buttons while listening – but don’t forget that there is a lag before it affects the sound being heard.

Having heard where the problem is, click at that point and play the surrounding samples. Further refinement is made by using the left or right arrow keys to scroll the display, coupled with replaying the suspect passage.

Undoing Repairs in the Sound Buffer

Once you have navigated to the desired location in the sound buffer, it may be that a repair was already made which does not seem to be acceptable. The first step should be to undo this repair, which will reveal the full extent of the underlying damage.

To undo the repair for review, shift-click in a red rectangle.

If the repair still seems reasonable, click “Fix” again. Otherwise, it may be changed, using the mouse and arrow keys. This is explained more fully in the next chapter.

The Last Buffer

When the last samples have been processed, there is still a full sound buffer that may not have been auditioned, and which has not been committed to file. An opportunity for review may be set using the preferences: there can be a one second warning beep and/or an indefinite pause before finalising the file. The default is for a beep but no pause.

Press “Pause” during this time to pause the program in the usual way. You can then listen to the contents of the last buffer, and also navigate back and make changes if desired. On resumption, the remainder of the output file will be written, and the repair statistics displayed.

The final statistics pertain to the whole file, unlike the progress statistics which are a running average for the current contents of the sound buffer.

8 Using ClickRepairRT

The purpose of ClickRepairRT is to enable records to be played via an application which applies the declick algorithms directly to the audio stream. Before using ClickRepairRT, it would be useful to explore the operation of ClickRepair itself. That is why this chapter occurs relatively late.

If ClickRepair is in trial mode, then ClickRepairRT will be also be in trial mode. There is no mechanism for registering ClickRepairRT other than to register ClickRepair itself, after which ClickRepairRT will be automatically registered. They are bundled as a pair.

Audio Input and Output

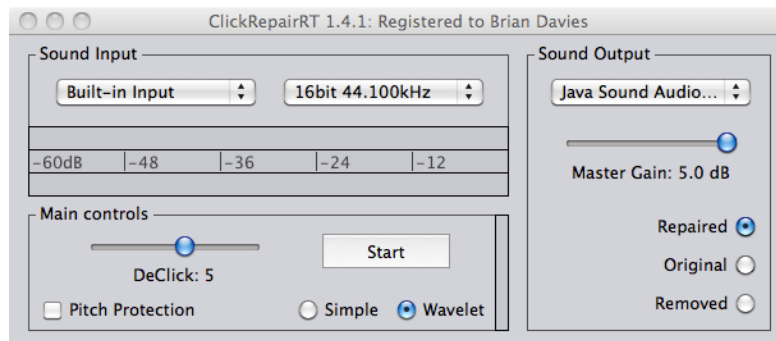
If ClickRepair has already been tried, the issue of audio output will have already been sorted out. The remaining hurdle is find a suitable set-up for audio input. The first thing to note is that the output from a turntable/pickup must be passed through a pre-amplifier to apply the necessary equalization; the pre-amplifier also brings the level up to that of a line input feed. On most computers, a 3.5mm audio input jack can accept audio from either:

- a microphone - this will be for a low-level signal (typically $\sim 5\text{mV}$).
- a line input feed - this will be for a high-level signal (typically $\sim 500\text{mV}$).

If the audio input, which is at line level, is fed into an input jack set for microphone input, severe clipping will occur which cannot be rectified later in the processing chain. Getting appropriate settings depends on the mechanisms provided by the operating system to control the available audio hardware. The following section applies only to the use of the audio input and output jacks which employ a built-in sound card provided with a particular computer.⁷

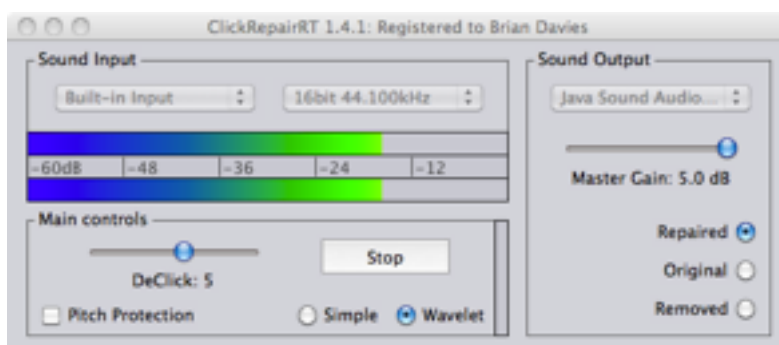
Using a Sound Input Jack

The first step is to start the application, which gives the window shown below. Notice the “Start” button. When this button is clicked, the application starts to process incoming audio, and the button changes to “Stop”. It is important to note that the pop-up menus for selecting sound input and



⁷ A viable alternative, which by-passes the inbuilt audio system entirely, is to purchase a USB audio interface.

sound output may only be changed while ClickRepairRT is not processing, so do not press “Start” yet! Rather, select the sound input and output source and sample rate from the pop-up menus. Then connect and start an audio source and (temporarily) mute the audio output.



Now press “Start” to process the incoming audio. The level input indicator should show that audio is being received by the application, and also indicate the input level. It should only go near the right-hand end of the scale, indicated by red, on the loudest passages, and should never go off-scale (beyond 0dB) at any time. This can not be adjusted within ClickRepairRT itself but must be adjusted in the audio subsystem of the computer itself.

- On Mac OS X, these settings can normally be accessed from System Preferences>Sound. Make certain that input is set to Line Input rather than Microphone. Finer control may be had from AudioMidiSetup in Applications>Utilities.
- On Windows systems the corresponding choice (between microphone and line input), may require turning off the “Microphone Boost” in an Advanced settings section. Depending on the hardware/OS combination, some experimentation will be needed.

Once the input has been set, undo the mute of the output and adjust the output level. Once everything is working, the settings will be remembered by ClickRepairRT from one use to the next.

Much of this depends on the individual Java Sound implementation, not ClickRepairRT itself.

Limitations and Differences

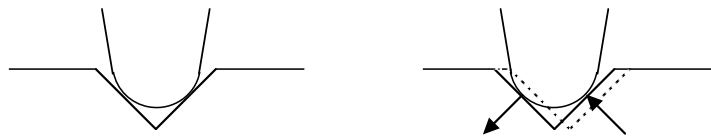
ClickRepairRT 1.4 is limited to sample rates of 96kHz. Processing corresponds to the “Reverse” mode of ClickRepair. The single sensitivity slider has no numbers corresponding to the slider in ClickRepair, and has fewer steps. The default (slightly more than half-way – there is no exact half-way setting) corresponds to setting of the DeClick slider in ClickRepair to 40. The “Simple/Wavelet” and “Pitch Protection” choices work in the same way as in ClickRepair;⁸ for most uses “Wavelet” is the best choice. It is also useful to remember that the listener is normally less critical when listening live, so it is better to err on the side of lower settings.

⁸ Switching between “Simple” and “Wavelet” will cause a sound glitch as internal processing is reset.

9 Processing Mono Records

Like stereo records, mono records have a V-shaped groove with a left- and right-hand wall. To understand the implications for processing of mono records, it is useful to consider first how stereo works, at least in a simplified way.

For a stereo record, each channel of the audio signal is recorded on just one of the walls. The movement of the stylus, in response to a signal in just one of the channels, is at an angle of 45° to the horizontal (and vertical). Relative to each other, these movements are at 90° . Ideally, therefore, the two signals may be recovered independently of each other. Damage to the groove walls can also be quite independent, although in practice both walls are often affected, but in a different way. In addition, phono cartridges are mechanical devices subject to mechanical limitations, and this ensures that a scratch in one wall will have consequences in both channels.



When stereo records were introduced, they had to be compatible with the existing mono players. These had settled on the standard of “lateral” recording, meaning that the signal is recorded as purely horizontal motion. Referring to the sketch above, if one wall moves down and to the left, while the other moves up and to the left, the net result is a purely horizontal movement of the stylus. Conversely, if the stylus is moving in response to tracking a mono recording, the response measured as movement of the two walls will have the same magnitude, and be at 90° to each other. Provided that phase conventions are chosen correctly, the electrical output which goes to the audio system will be the same in each channel, and the record will be heard in mono.

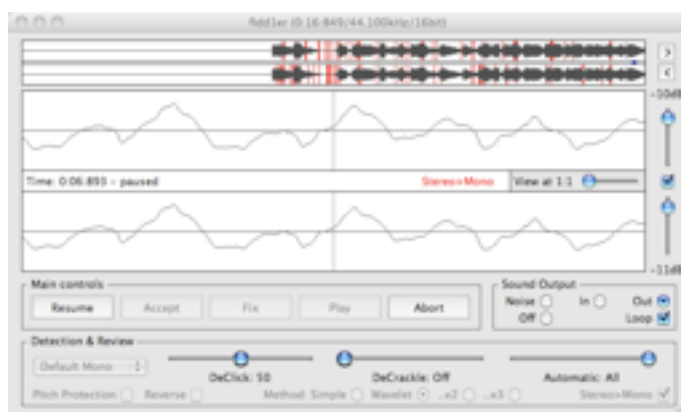
File Formats

Before explaining the implications of this information for the operation of ClickRepair, it is essential to be clear about the use of the word “mono”. The output from a typical record-playing system is stereo, regardless of whether the record is stereo or mono. When this output is captured to a sound file, the file will normally be in stereo format, unless the signal is mixed down to mono at that stage, and saved as a file with just one channel.

A mono file has just one channel of recorded data; a stereo file has two channels of recorded data. This does not depend on the source material – both channels might contain identical sound data.

The figure below shows a small section of a stereo capture of a mono recording. The two channels have an obvious high degree of correlation, but they are certainly not identical. One obvious difference is the fact that a small repair has been made in the left channel, which was not needed in the right channel. Another difference is evident from the scale sliders to the right of the display, ac-

ording to which the peak levels differ by about 1dB. (In fact, I examined this file quite carefully and found that the level difference was 1.4dB over the entire transfer.)



For mono records, I strongly recommend that the capture is done in stereo, to a stereo file. ClickRepair will automatically process the file as a stereo file. There is provision to merge the resulting output, as explained below, but the file format will remain stereo. This method gives much better results for mono recordings, both vinyl and shellac.

Even if the signal has been pre-mixed to mono, if the file has two channels of data, it is a stereo file, and will be processed as stereo by ClickRepair. Note, however, that if the two channels contain identical sound data, then only one is actually processed internally.

In summary:

- Better noise reduction may be achieved by capturing and processing mono material in stereo.
- The file format for a CD is stereo even if the source material is a mono record.

Users of previous versions of ClickRepair will notice that the “Mono” and “Shellac” check-boxes have disappeared. In fact both were associated with pre-mixing the incoming signal to mono. With ClickRepair 2, the only way to process a single channel is to use a sound file with only one channel.

The Stereo>Mono Option

Stereo files of mono records are processed in stereo – decisions on click detection and repair in the two channels are made independently. However, since the source material is mono, the final file should reflect this fact. There are a number of possibilities:

- Audition the left and right tracks separately, and choose the one which is better. One wall of the groove is often less damaged than the other.
- A variation of the above – splice together sections of the tracks to provide the final mono cut. This can be very time-consuming, but can produce good results.
- Merge the two channels to produce a single mono track, which is written to both channels of the stereo format file.

A problem for all of these techniques is one of “normalization”. It is nearly always the case that the two channels of data, even if they are highly correlated (as they should be), are at different recording levels. I have shown an example of this above. Each of the processes involved in capturing the record to a sound file can contribute to this lack of equalization.

The “Stereo>Mono” option in ClickRepair performs a merge of the two channels to provide a mono signal (option 3 in the above list). The same data is written to both channels of the output file. The merge automatically, and dynamically, adjusts the channel contributions to the mix (the idea behind options 1 and 2). An outline of how it works is as follows:

- The left- and right-hand channels are processed independently, in order to extract the maximum amount of information.
- The two channels are merged, using dynamically adjusted levels computed via a moving average.
- The moving average is calculated over a long scale, so as not to introduce audible effects.

All dynamic compression/expansion algorithms such as this introduce non-linear artifacts, which will be heard unless they are kept below auditory perception limits. The artifacts produced by the merge algorithm are at a very low level and in the frequency range 0-20Hz.

- The average level of the merged output is exactly the same as the average of the levels of the incoming channels. This means that the louder channel will be reduced, the softer one amplified.
- Significant level changes will normally be associated with major damage – for example a bad scratch. An example is shown below. Note the remaining “bump” in the left-hand channel.

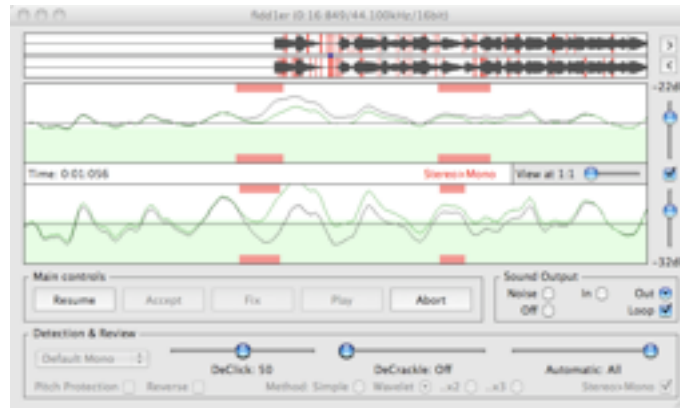
If you intend to process a file more than once – perhaps for a subsequent de-crackling – do not apply the merge option at any of the intermediate stages. This way each channel will be processed independently each time

Viewing Merge Details

It is possible to gain some information on how the merge algorithm is operating, as follows:

- First, pause processing.
- Use the sound bar to navigate for the desired place in the sound buffer.
- Finally, hold down the “Option/Alt” key.

In response to these actions, the display will change, as shown in the figure below. I have selected a spot where there was a major scratch, mainly in the left-hand channel. It is clear that there is a significant “bump” between the two repairs which were made automatically. This is witnessed by the fact that the peak levels, displayed by the two vertical sliders, suddenly differ by 10dB.

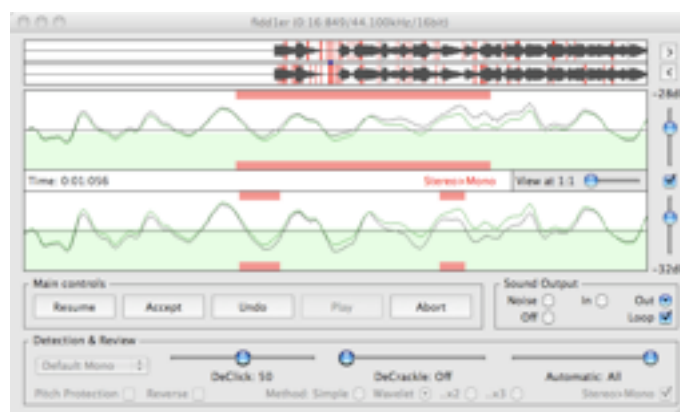


As before, the black traces show the repaired audio signal, with repairs indicated by red rectangles above and below. However, the level difference has been taken into account when making the merge. The information about this is displayed in green, as follows:

- The light green shading indicates the relative normalizations applied for the merge. It shows that the right-hand channel was amplified considerably, while the left-hand channel was reduced.

The range of this indication is from -6dB to +6dB. In the example shown, the left channel is reduced by about 1.5dB, the right channel is increased by about 2dB, to achieve the final mix.

- There is a second signal trace in dark green, which shows the merged and normalized signal, correlated with each channel individually. Where the traces coincide, it overwrites the black trace.
- Looking at the dark green traces, we see that these adjustments lead to a considerable improvement in the final result. Because greater weight has been given to the right channel, the “bump” has been reduced.
- Over the 1400 samples on display here, the relative levels appear to be constant, although they are in fact changing slowly. It is necessary to change the horizontal scale to a higher value in order to see the dynamics more clearly.



Significant damage will often benefit from manual adjustment. This is the subject of the next chapter, but to make this clear, here is the result of making a much longer repair in the left channel.

The “bump” is taken out of that channel and the relative level difference is reduced, making for a better overall result.

Shellac Records

Much of what is in this manual applies to material sourced from any record, vinyl or shellac. In this section I make a few brief comments on the challenges posed by the older shellac records.

The introduction of vinyl records coincided, more or less, with the introduction of full frequency range recording. Most shellac records were designed to play at 78 rpm (or thereabout), and their frequency range was restricted to about 8 kHz, because of technical limitations. Equalization schemes varied, although a common feature was the reduction of bass frequencies to limit groove excursion. Details may be found in various places, for example at www.rfwilmut.clara.net. The granular nature of the shellac recording material means that hiss is still quite prominent even after applying the proper equalization, which will lift the bass relative to the treble, possibly followed by a low-pass filter to further reduce the treble. On the other hand, most vinyl records play at 33 rpm and have a frequency range from subsonic to 20 kHz or more. Hiss is reduced to a low level by a combination of the smoother vinyl material, together with RIAA (or similar) pre-emphasis which extends to +20dB at the top, and -20dB at the extreme bass (again to limit groove excursions). The phono stage of a typical amplifier boosts the bass and reduces the high frequencies according to the RIAA standard. This implies that hiss, which is already lower because of the vinyl material, is reduced to an acceptable level, and should require no further treatment.

Given a shellac recording, the first challenge for many people is to find a suitable turntable and cartridge with which to capture it. It is essential to have a stylus with the larger tip radius required for shellac records, but it is not essential to have a turntable which will play at 78 rpm. The former problem can often be overcome by purchasing a second stylus for an existing cartridge, the latter problem can be circumvented by recording at 45 rpm and using an audio editor to re-sample. The same audio editor will be required to apply the necessary equalization and other operations.

In fact, I have an older turntable which will play at 78 rpm, but prefer the lower noise levels of a later turntable which only plays at 33 and 45 rpm. If the original record was designed to play at 80 rpm, resampling is required in either case.

In an earlier chapter I mentioned that the minimum useful sampling rate, for a vinyl record, was 44.1 kHz. This is because the highest theoretically available frequency is one-half of the sampling rate – known as the Nyquist frequency limit. But, if the source material is limited to about 10 kHz, then those comments no longer apply. In addition, a 78 rpm record played at 45 rpm and captured at 44.1 kHz is effectively sampled at 76.4 kHz. This means that there are many possibilities for how to proceed. One suggestion is as follows:

- Run the file through ClickRepair without resampling or equalization, possibly merging to mono. Merging will remove quite a lot of vertical low-frequency noise.

- Resample and equalize. Remember that, if you captured using an audio system which applies RIAA equalization, then after resampling the RIAA turn-over and cut-off frequencies will have shifted accordingly⁹, whereas the corresponding 78 rpm frequencies will only now be correct.
- Run the file through ClickRepair again, particularly to take advantage of de-crackling. If the file is still stereo, apply the merge at this time.
- After all this the remaining hiss will be continuous, and possibly obtrusive, but should be suitable for the application of de-hissing and/or further equalization, as desired.

An important point to note is that most de-hissing algorithms rest on the assumption that the noise has a “Normal” or “Gaussian” statistical distribution. Crackle is a complete misfit because the proportion of outlying data does not fall off in the assumed manner. De-crackling concentrates on removing the outliers, leaving a file which should be much more responsive to de-hissing.

There are many other recipes which could be suggested.

For example, if the record was played at 45 rpm, an audio editor can be used to change the sample rate to 76.4 kHz without resampling. ClickRepair will operate on this file differently, applying an extra level of wavelet decomposition: see the section on Sample Rates in Chapter 4. But, because the audio was not resampled, it is the identical data set which is being treated. I have found that this technique may improve de-clicking but degrade de-crackling, because the detection algorithm is working at half the sample rate.

The main point is that restoration of old recordings is as much an art as a science – a considerable amount of experiment and critical listening will be required to achieve good results. The only firm recommendation I can make is to do the initial de-clicking on the original data set, before any equalization and/or resampling.

⁹ For example, 500Hz will move to 867Hz when changing from 45 rpm to 78 rpm.

10 More Advanced Methods

The methods described in Chapters 6 and 9 were the basis of ClickRepair 2, and they remain the core of ClickRepair 3. Not only that, their operation is unchanged in the new version, so for clarity I have left their documentation separate and essentially unchanged. In many situations these are the only algorithms that will be needed.

This chapter describes the two additional algorithms which are the main new feature of ClickRepair 3, and the uses for which they these new algorithms are intended.

Pops and Bumps

Clicks and crackle are usually heard because each individual disturbance lasts for a very short time, and is perceived as having a high pitch. This is because such disturbance can only be represented using the higher frequencies, and hearing depends quite critically on frequency based aural processing. For the same reason, clicks will not be perceived if they are masked by sufficiently loud high frequency audio which overlaps in both the frequency and time domains. The science of psychoacoustics, which is well developed, is concerned with understanding which elements present in a given audio stimulus can actually be perceived, and which are masked by what is actually being perceived.

The problems this raises for the restoration of degraded audio material are immense. Some issues which are of relevance for de-clicking are:

- Repairing a click might leave a pop or bump. If such a residual artifact is masked by the desirable audio, there is no problem. If not, one problem is seemingly replaced by another.
- The mechanical or electrical cause of the click might have introduced a pop or bump, hardly discernible when masked by the click, but all too evident once the click is repaired.
- Pressing flaws on records often produce pops and bumps which have no associated click. De-clicking will not even find these.

ClickRepair has some methods for helping to address these problems. They revolve around a more extensive use of wavelets, and are therefore displayed in the interface as extensions of the Wavelet method. Specifically:

- Wavelet “x2” uses two levels of wavelet decomposition, analysis, and repair.
- Wavelet “x3” uses three levels.

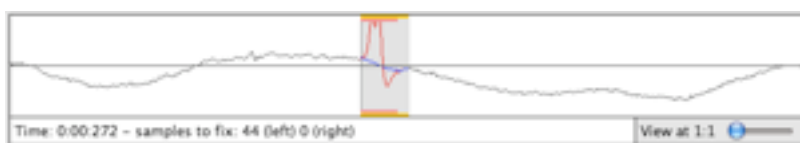
Wavelet ..x2 .. x3

These methods add extra layers of wavelet, or multi-resolution, analysis to the existing wavelet method. This is seen in the ClickRepair interface: the pair of radio buttons (Simple, Wavelet) of ClickRepair 2 has been augmented by two more radio buttons (..x2, ..x3) in ClickRepair 3. Repairs

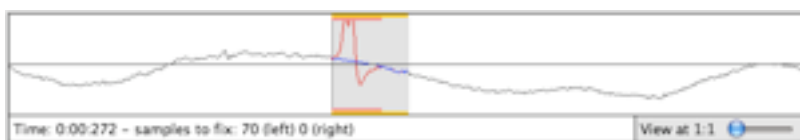
made with these new methods are “multi-layered”, and there is some indication of this when they are displayed. Two colours are used:

- That part of the repair detected and made in the extra wavelet layers is indicated in orange.
- That part of the repair made by the standard (wavelet) de-click algorithm is shown in red.

Notice that the two types of repair may overlap – see the examples shown below. This is always the case if the longer time-scale defect is associated with a detectable click. The sensitivity setting for all levels of detection and repair are set from the single “DeClick” slider, and normal de-crackling is still available in conjunction with these methods.



To gain some understanding of the different outcomes of these methods, I revisit the example shown in Chapter 6 comparing “Simple” and “Wavelet” repair methods. The two pictures here are at the same sensitivity, but now using “Wavelet x2” (above) and then “Wavelet x3” (below).



After examining a number of repairs made on this particular file, I decided to use “Wavelet x2” rather than “Wavelet”, because the results seemed a little better, both by visual inspection, and on listening. I rejected “x3” because it seemed to introduce extra processing of the audio without any extra benefit, but with an increased risk of degradation. However, it must be said that I could not hear any extra degradation with this particular file.

Caution: Each new layer of processing adds new possibilities for false detection, making careful experimentation and listening even more important when choosing between the available options.

Flattening of Manual Repairs

Manual adjustment of these repairs results in them being “flattened” to a normal wavelet repair. That is to say, there is no mechanism for manually constructing one of these “multi-layered” repairs. If processing stops because a repair exceeds the “Automatic” setting, or if you pause processing and manually undo a repair, it will be presented with the structure intact, as in the picture below. The “Fix” button will apply the structured repair, “Undo” will undo it again including its structure. But, once you make any manual adjustment whatever, the repair will revert to a normal wavelet repair and the original structured repair cannot be recovered.

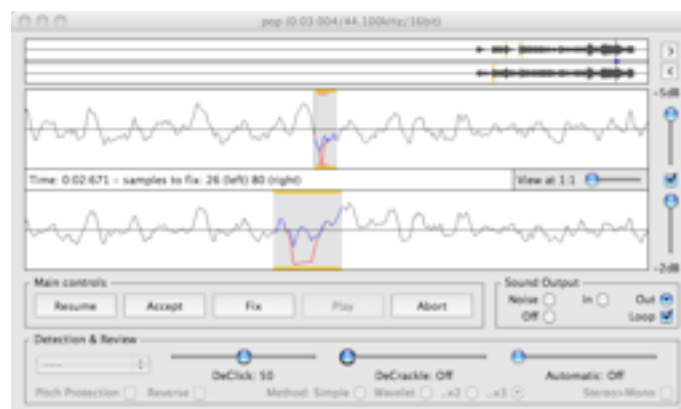


I illustrate here by showing what happens if the repair above is turned into a normal manual repair by the simple method of moving one edge of the repair by one sample and then back again, so that precisely the same set of samples are repaired, now using the standard wavelet method.



In this case there is not a lot of difference, because the surrounding audio happens to be so nearly periodic. Nevertheless, the question arises as to which of these two repairs is better. There is no definitive answer. Either of them represents the result of a statistical analysis to estimate what might have been there. Both of these estimates are made “blind” – that is, in the absence of any actual knowledge of what has been lost. The best one can hope for is that the difference between what might have been, and what has been reconstructed, are perceptually indistinguishable. Given the severity of the damage shown here, that is almost certainly not completely true, although either version does sound a great deal better than the badly damaged original.

A Genuine Pop



Here is a different kind of example – the automatic detection and repair of a pop that goes undetected using the normal wavelet algorithm (even at a sensitivity of 90). Clearly there is a significant pop in the right channel (bottom trace), and the fact that the automatically detected and generated repair has no red rectangle is due to the fact that there is no associated click. This is despite the fact that a click has been detected in the left channel (upper trace). Using previous versions of ClickRepair, this pop could only be repaired by finding and treating it manually, a very tedious and time-consuming process.

Manual repairs are the subject of the next chapter, since there are times when this is the only effective remedy available.

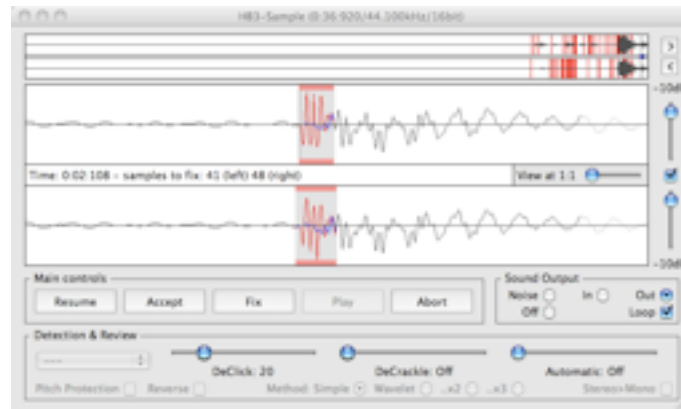
Time-reversed Processing

A number of users have noted that, with certain types of audio material for which false detection is a difficult problem, the situation can be much improved by reversing the audio file (so that it plays backwards), processing the reversed file with ClickRepair, then reversing the new file to make it play forwards again. The underlying reason is simple: some percussive sounds have a sharp attack and relatively slower decay, and when played backwards, that becomes a slow attack with a sudden end to the sound. The algorithms used in ClickRepair do not see these two in the same way, when attempting the difficult task of detecting clicks among the music.

ClickRepair 3 has provision for this type of “Reverse” processing, on the fly. This saves a great deal of time in situations where reverse processing is of significant advantage. The file, as a whole, will still be processed in a forward direction, and audio monitoring retains its normal operation.

What makes this possible is that the existing algorithms, as part of a process of iteration, actually treat the file in small overlapping segments, each in the forward direction. Selecting the “Reverse” checkbox causes overlapping segments to be individually processed in the reverse direction, without any need to time-reverse the entire file.

Below is a typical false detection and associated false repair made to a file which features a xylophone. The setting is DeClick 20 in Simple mode. Since this is not an isolated case, but happens repeatedly with this particular file, the sound degradation is unacceptable, completely altering the sound of the xylophone. Clearly, for this file, automatic de-clicking is just not useable with forward processing.



Selecting the “Reverse” check-box, only a couple of false detections are made at the DeClick setting of 20. Even at such a low sensitivity, most clicks due to actual record damage are still detected and repaired, even though the attack of the xylophone gets through unscathed.

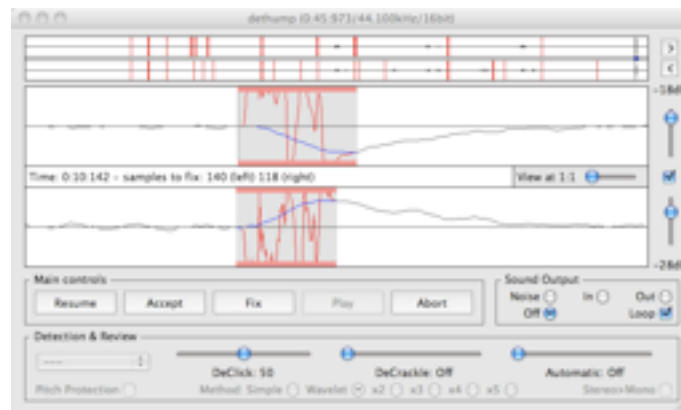
A new setting “Default Percussion” has been added to indicate the likely use of reverse mode.

Reverse processing is available in all of the wavelet modes as well. Its operation is also independent of the “Pitch Protection” and “Stereo>Mono” merge options, and it can be used in conjunction with any or all of those features. Note, however, that for the new “Default Percussion” setting I have chosen “Simple” mode, since these default settings are meant to be a starting point for individual experimentation.

11 Manually Adjusting a Repair

There are a number of reasons for manual intervention, even though this can be painfully slow compared with automatic mode. They include:

- ClickRepair may have missed a significant click or thump, and you may have to locate it manually.
- ClickRepair may have found a click, but under- or over-estimated its extent.
- Record damage can be very severe, ranging from a major scratch that extends across many tracks, to even worse scenarios, and may need individual attention to remove a residual thump.



Using the Mouse

Assuming you have the program stopped, with the offending samples firmly in view in the display window, you can initiate or adjust a repair using the mouse:

- To choose the start of the repair selection, left-click at the desired position.
- To choose the end of the repair selection, right-click.
- Either of these actions may be combined with dragging the mouse – that is, moving the mouse while keeping the button depressed.
- The size of the selected area is limited to 2048 samples.
- Each time you change the selection, a new repair will be constructed and displayed in blue.
- You can clear a selection by left-clicking to the right of the selected area or right-clicking to the left of the selected area.

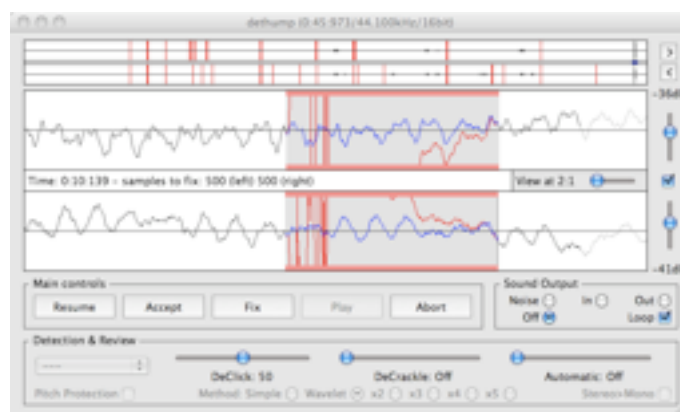
With a one button mouse on a Macintosh, click is left-click and control-click is right-click.

Using the Arrow Keys

The arrow keys may be used to refine the selections. To understand how this works, first notice that one edge of a selection rectangle displays as a darker vertical line, and that, on changing a selection using the mouse, this dark edge tracks the most recent change. This is to indicate that that edge may be moved, one sample at a time, using the left and right arrow keys.

- To move the selected edge left or right, use the left and right arrow keys.
- Holding down the shift key accelerates the arrow keys by a factor of 10 – the selected edge moves 10 samples at a time.
- To toggle between the right and left selected edges, use the up and down arrow keys.

The first figure (previous page) displays an example taken from a favorite LP that met with an accident. I got the program to stop by setting “Automatic” to 30, and then (for the purpose of this picture) I manually undid the work which ClickRepair had in progress. Clearly there is a heavy “thump”, which can be seen and heard; its reduction requires manual intervention as shown here. Since the damage is so extensive, it can be difficult to find all of it automatically, although it is unlikely that ClickRepair will miss it altogether.



The second figure shows an acceptable restoration, which was constructed, in each channel, by right-clicking with the mouse in order to extend the end points. The repair, and the original damage, were auditioned using the “Play” button. Because of the magnitude of the damage, several attempts at repair were auditioned and compared. Although this is time consuming, hopefully you will not have too much damage of this magnitude to deal with!

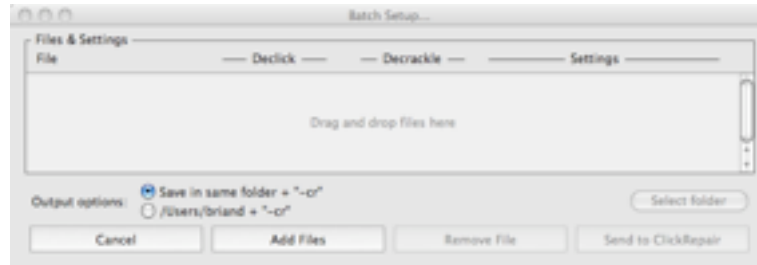
Because so much of the audio data has been lost, a repair such as this cannot reconstruct the original sound, although it can be a plausible interpolation. In the example shown, 500 samples have been interpolated, about 11.3 milliseconds of sound.

In fact, all but a few of the heavy cracks in this particular record were repaired satisfactorily without manual help, but the residual thump shown in this example was on too long a time scale, and too smooth, for further automatic progress..

Important: the first and last 1024 samples in the sound buffer cannot be repaired manually because the algorithms may require up to 1024 samples on either side to make a repair.

12 Batch Processing

To prepare to process a batch of files, choose “Batch Setup” from the “File” menu. This will bring up the dialog shown here.



Note that this dialog grows in height as more than 4 files are added, however, the drag and drop area will always be sufficient to display 4 files. The maximum number of files which can be added is 256, of which a maximum of 16 will be visible at any one time. This is controlled by the vertical scrollbar, which will change appearance once there are more than 16 files.

The display will be slow to update if a lot of files are selected at one time, since each file has to be checked, and a new skeleton file created.

Organisation

The first thing to decide is how to organise the output files in relation to the input files. This is controlled by two radio-buttons and a “Select folder” button. The first option is for each output file to be saved in the same folder as the corresponding input file, and named by adding “-cr” to the name of the input file (or another string if this has been changed in the “Preferences” dialog). The other radio-button allows the output file to be saved to a selected folder, the pathway to which is displayed alongside the radio-button.

In this mode, the name of the new file will be extended only if this is the selected preference in the “Preferences” dialog, whereas the file name is always extended when saving to the same folder.

When the second option is selected, the “Select folder” button will be enabled. It may be used to change the output folder if the currently displayed choice is not the desired destination. Only folders will be displayed in this dialog, since we are not yet to the point of choosing files. Navigate to, and select, the desired output folder, and click “Select”. All these settings are saved from one session to the next.

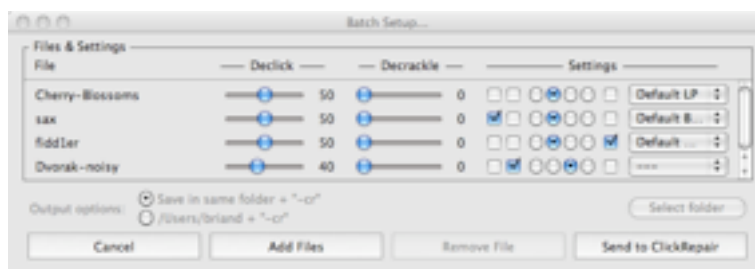
Note that the file dialogs used for batch setup are not the standard file dialogs provided by the operating system, rather they are Java “Swing” components, and have somewhat different behaviour.

The choice of destination may not be changed once files have been added to the batch.

Assembling a Batch

Once you are ready to add files, click the “Add Files” button in the “Batch Setup” dialog, or drag and drop files into the dialog window in the highlighted area.

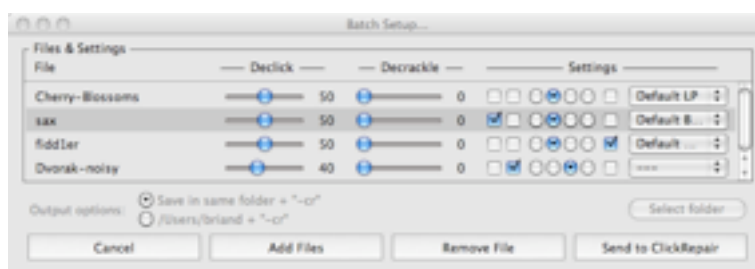
Clicking the “Add Files” button will bring up another file dialog, this time permitting only files to be selected. Select one or more files, and then click the “Select” button. The files will be added to the dialog, as shown here. To extend the selection in the file dialog, use the shift key. Random multiple selection may be made, using the “Apple” key on Macintosh, the “Control” key on Windows.



If you choose files which cannot be processed by ClickRepair, then an error dialog will appear for each such file, in turn. Also, if you try to organise the files so that an output file will over-write its input file, this will generate an error dialog, and the file will not be added.

Note that there will be no confirmation dialogs asking permission to overwrite existing files. It is possible to lose existing files if you choose to replace them in this way when batch processing. Note also that the new files are created as soon as the files are added to the “Batch Setup” dialog. Moving these files before processing is complete will cause unpredictable errors.

Individual detection and repair settings are provided for each file in the batch. Controls for these settings appear next to each file name. There is insufficient room to label the check-box and radio buttons, which correspond in left to right order to those appearing across the bottom of the main ClickRepair window. Note that the drop-down list has access to, and displays, any custom settings which you have saved.



To remove files from the batch, simply click on the name of a file in the “Batch Setup” dialog. The corresponding line will be highlighted, as shown here, and the “Remove File” button will be en-

abled, to allow removal. For removal, the shift key may be used to extend a selection; multiple random selection is not supported.

Processing

When you are completely satisfied with the batch settings, click “Send to ClickRepair”, which will dismiss the dialog and cause the “Open” button to change to “Start”. The “Automatic” slider will be set to “All”, although the previous setting will be saved. If you do not alter this setting during processing of the batch, it will be restored on completion.

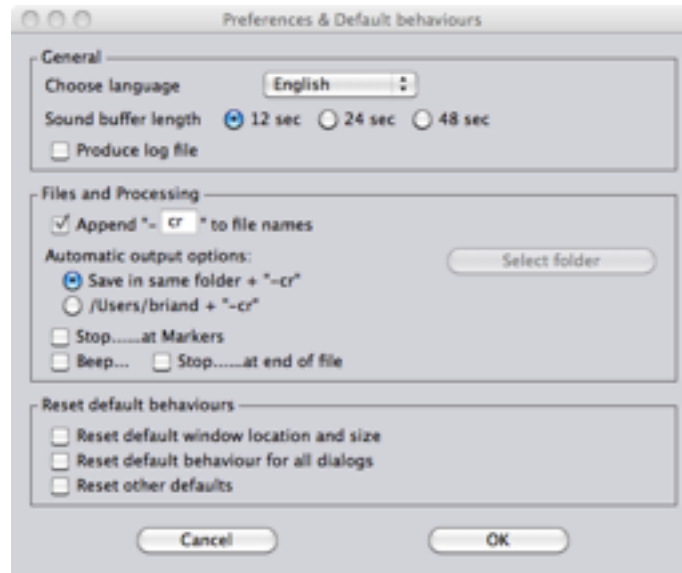
Processing will not start automatically, you must click the “Start” button to commence. Before you do, you might like to review the “Sound output” settings. If you don’t want to run in fully automatic mode, change the “Automatic” slider setting.

If you click the “Abort” button during batch processing, the action will apply only to the file currently being processed. If the action is confirmed, the window title will change to show the next file in the batch, the “Pause” button will change back to “Start”, and processing will be suspended. You must click “Start” to continue. You may also choose to abort files before processing begins, until the list of files is exhausted, and in this way abort all or part of a batch.

13 The Preferences Dialog

Preferences

Using the “Preferences” menu item on Mac OS X, or clicking the “Preferences” button on Windows, brings up the dialog shown here.



Language

ClickRepair has a choice of three languages in the user interface – English, French, and German. If your computer is set to French or German as language of choice, the program should automatically use that language from the first time it is used. To change the language in use, use the drop-down list in the preferences dialog.

The French translations were kindly provided by Michel Poulain; the German translations by Willy Kunz. Both are long-time users of the application.

Sound Buffer Length

Versions of ClickRepair previous to 3.8 had a set buffer length of 524,288 samples – approximately 12 seconds at 44.1kHz. In the new versions, the choice of a longer buffer is intended to make it more convenient to experiments with, and re-process, selected portions of the file while they are still resident in computer memory (RAM). The extra memory requirements are significant, and choosing the longest buffer length (2097152 samples) may cause “Out of Memory” dialogs on some systems. Because the necessary memory allocations are made as the file is opened, it is only possible to change these settings while no file is open.

Log Files

Logging may be turned on at any time, using the preferences dialog. At the start of each logged session, a file dialog will appear to allow the log file name and location to be chosen. Clicking the “Cancel” button in this dialog will turn logging off. It may also be turned off at any time using the preferences dialog. The log files are plain text for easy use. They record the file names, settings, and times taken.

Files and Processing

The “Append -cr to File Names” check-box determines the default behaviour for how the output file is named, and where it is saved, in relation to the input file. The default appended string “cr” may be changed by typing a new choice in its place. Remember:

- If the same name is used for both input and output files, they cannot be in the same folder.
- The default locations for each are the folders last used for input and output.
- When batch processing, the save options are controlled directly from the “Batch Setup” dialog.

The “Save in same folder + cr” checkbox determines the folder to which the new file is saved by default, that is, the way the save dialog is initially displayed. If this check-box is selected, the dialog will be set to save the output file in the same folder as the input file, with “-cr” appended. If this check-box is not selected, then the dialog will be set to the same folder as the last output file, with or without an appendage.

The “Beep” and “Pause on completion of processing” check-boxes allows you to enable or disable the warning beep at the completion of processing, and to pause before writing out the final buffer to file, if required. If you are listening to the audio, in real time, through a reasonable sound system, I recommend turning off the beep.

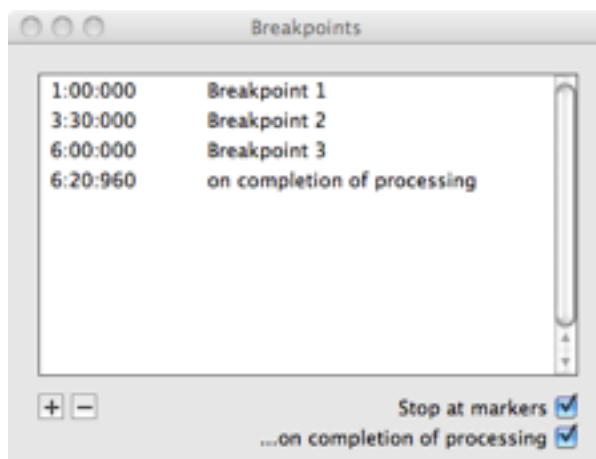
Breakpoints and Markers

Breakpoints (including Markers) can be extremely useful, allowing the user to choose places in the file where processing is paused so that manual settings adjustments and/or manual repairs can be made. ClickRepair 3.6 greatly expands this feature over previous versions, which only recognised markers placed in the file by selected Audio Editing applications.¹⁰

Starting with ClickRepair 3.6, when a file is opened, a “Breakpoints” dialog is available. This dialog will pick up any existing markers; more importantly, it may be used to set arbitrary breakpoints. The latter will not be added to the file itself; they are remembered only so long as the file remains open. The picture here shows three breakpoints that have been added by clicking on the “plus” button (which brings up yet another dialog for that purpose). Any of the breakpoints may be removed by

¹⁰ Almost certainly a Mac OS application, since the MARK chunk was originally defined for AIFF files. Although they can also be used in WAV files, the common Windows Audio Editors do not use them.

simply clicking on them, followed by clicking on the “minus” button. In the picture there is a fourth breakpoint set at the end of the file. This was done automatically because “Pause on completion of processing” is selected in the preferences. The two check-boxes at the lower right of the “Breakpoints” dialog are linked to the corresponding check-boxes in the “Preferences” dialog; selecting or de-selecting these check-boxes in the one dialog will carry over to the other, and also add or remove this final breakpoint. The breakpoint dialog will open automatically if “Stop at Markers” is selected in the preferences; otherwise it may be invoked from the “File” menu item “Breakpoints...”, once the file is open.



In using these breakpoints, remember that the “Abort” button can be used to restart processing from the beginning of the file. A consequence of having “Pause on completion of processing” as a marker is that processing can be restarted before the file is closed. Breakpoints may be inserted during or after processing. This can be useful if it is already too late to go back – simply insert a marker and then restart processing from the beginning. Remember, too, that with sound output turned off, processing is very fast when using the latest dual and multi-core computers.

Resetting Defaults

Normally, ClickRepair re-opens with the window in the same location, and at the same size, as when the program was last shut down. “Reset Default Window Location and Size” erases this persistent information, and resets the window immediately, but only after the “OK” button is clicked. The information will not be reset again until the Preferences dialog is again invoked.

“Reset Default Behaviour for all Dialogs” refers to the fact some of the dialogs have a “Don’t Show Again” button. This is to prevent such dialogs becoming annoying once you understand the flow of program use. If you wish to see any of these dialogs again, check this box and select “OK”.

“Reset other defaults” resets all check-boxes and radio-buttons not associated with the “Detection & Review” control group.

14 Other Features

Behaviour of Buttons

The “Main Controls” group consists of five buttons, but some buttons have more than one function. This has already been mentioned indirectly, in relation to the “Open” button, which became the “Start” button, then alternatively the “Pause” and “Resume” button. Other buttons with multiple use are the “Fix” button, which can alternate with “Undo, and the “Play” button, which can alternate with “Stop”. These buttons can be operated using fixed keyboard shortcuts, as described earlier, but they never receive the keyboard input focus, and cannot be clicked using the spacebar, or any other normal method for the particular platform. By contrast, the buttons in dialogs¹¹ may be operated in the default manner by first using the tab key to bring the desired button into focus, then clicking the space bar.

Abort Button

The “Abort” button does just that: it results in the output file being deleted, since it would not be readable. The immediate response is to pause processing and display a dialog asking for confirmation of the action. If you want to stop processing, but still finish copying the input file to the output file, move the “DeClick” and “DeCrackle” sliders to “Off ”, and wait for processing to finish.

File Dialogs

Java has two types of File Dialogs; ‘native’ dialogs as provided by the underlying operating system, and ‘Swing’ dialogs that are provided by the Java Virtual Machine (Java run-time system). Normally there is no need to worry about this, and ClickRepair will choose native dialogs whenever possible.¹² A small number of users have been plagued by random bugs in the native dialogs, as provided by the Java run-time, both on Apple Mac and Windows system. The “File Dialogs” item on the “File” menu provides a work-around in such situations, since the problem is not in the ClickRepair code itself.¹³ Another work-around is to use drag and drop for file selection, although this places some limitations on the destination of the new file.

¹¹ Note that the “Shortcuts” window is not a dialog. Its “OK” button cannot be focussed, and the window can be closed using the close button/box in the title-bar, unlike dialogs.

¹² The exception is that Swing dialogs are used for assembling batch jobs, because of some limitations imposed by native dialogs.

¹³ The symptom is a message saying that the selected file cannot be found, the name of the selected file having been corrupted as it was communicated to ClickRepair.

Keyboard Operation

Nearly all the menu items have menu shortcut keys, and may be selected in this way. In addition, the buttons in the “Main Controls” control group may be operated by various shortcut keys (without the command or control key); these may be viewed by choosing “Keyboard Shortcuts” – see Chapter 5 for details. This makes for speedy operation, but it also enables mistakes to be made more speedily, so beware!

In the main window, the “tab” key cycles the keyboard focus around the objects in the “Detection and Review” group; once an object has the focus it may be operated in the normal manner – the space bar is equivalent to clicking a check-box or radio-button, while the left and right arrow keys operate the sliders.

As mentioned earlier, the selection point for manual repairs may be manipulated using the arrow keys. Here is a list of these keyboard shortcuts:

- The left and right arrow keys move the currently selected edge of a repair one sample at a time.
- Holding down the shift key while using left and right arrow keys moves the repair 10 samples at a time.
- The up and down arrow keys move the selected edge to the next, or previous, unselected edge.
- When there is no selected repair in the display window, the left and right arrow keys are equivalent to clicking the left and right scroll buttons of the sound bar.

Mouse Actions

Here is a list of mouse actions:

- Left-click defines or moves the beginning of a repair.
- Right-click defines or moves the end of a repair.
- Left- or right-click, followed by dragging the mouse, also defines a repair.
- Shift click in a light red box undoes the corresponding repair.