

# A User's Guide to the *Sound Loom* and CDP Sound Transformation Software

By Trevor Wishart

Ed. & Published, January 2009 (AE)

A detailed review of the *Sound Loom* graphic user interface facilities for the Composers Desktop Project (CDP) sound transformation software, set in the context of practical compositional objectives.

© 2007-2009 Trevor Wishart, York, England

Copies may be made freely.

## Table of Contents

<b>Section 1: INTRODUCTION</b>	1
<b>General</b> .....	1
Spelling / Naming / ... and Backing up .....	1
<b>Getting Started</b> .....	3
<b>The Workspace</b> .....	3
The Workspace & ‘Help’ .....	3
The Workspace Directory Listing Panel (Right) .....	3
The Workspace Panel (Centre) .....	4
Backing up your files .....	4
The Chosen Files Panel (Left) .....	5
<b>The Process Page</b> .....	5
<b>The Parameters Page</b> .....	6
<b>The Run Page</b> .....	6
<b>Back at the Process Page</b> .....	6
Saving the output .....	7
What to do next .....	7
<b>Control Files</b> .....	7
<b>Sound View</b> .....	8
<b>Remembering and recalling what you have done</b> .....	9
At a local level .....	9
At a global level .....	9
Log Files .....	9
Patches .....	10
Favourites .....	10
<b>The Calculator</b> .....	10
<b>The Table Editor</b> .....	10
<b>Section 2: CLEANING UP SOURCES &amp; USING EDITING TOOLS</b>	11
<b>Cleaning and shaping your sources</b> .....	11
<b>Section 3: EXTENDING YOUR SOURCES</b>	13
<b>Time-stretching</b> .....	13
<b>Looping</b> .....	13
<b>Zig-zags and Drunken Walks</b> .....	14
<b>Brassage</b> .....	14
<b>In (Group) Brassage (Function) Brassage</b> .....	14
<b>Shredding</b> .....	15
<b>Section 4: WORKING ON THE SPECTRUM / MULTI-PROCESSES</b>	17
<b>Spectral Processes</b> .....	17
Inharmonic and Harmonic spectra / Spectral stretching .....	17
Spectral Tracing .....	18

Spectral Blurring .....	18
Spectral Freezing .....	18
Imposing Vowels .....	18
Morphing .....	19
<b>Multi-Processes (“Instruments”)</b> .....	19
Instruments with parameters .....	20
<b>Section 5: LOUDNESS ENVELOPES / SEPARATING &amp; RECONSTRUCTING A SEQUENCE OF EVENTS</b>	21
<b>Envelopes</b> .....	21
Extracting the envelope .....	21
Imposing the envelope .....	21
<b>Separating &amp; Reconstructing a sequence of events</b> .....	22
Saving the timings of the ‘footfalls’ .....	22
Separating the ‘footfalls’ .....	22
Dealing with reverberation .....	22
Reconstructing the original sequence .....	23
Modifying the rhythm .....	23
<b>Section 6: SPACE, TEXTURE / TABLE EDITOR &amp; CALCULATOR</b>	25
<b>Space</b> .....	25
Select the process Space / spatialisation / pan .....	25
Depth and other motion clues .....	25
Circular motion .....	26
<b>Texture</b> .....	27
Note data .....	27
Event packing .....	27
Scatter .....	28
Time grid unit .....	28
First / Last sound in list .....	28
Min / Max event gain .....	28
Min / Max event sustain; Play all of insound .....	28
Min / Max pitch .....	28
Overall attenuation .....	28
Spatial position .....	29
Spatial spread .....	29
<b>Table Editor (5 Panels)</b> .....	29
Basic mode of operation .....	29
<b>Calculator</b> .....	30
<b>Section 7: FILTERS, DISTORTION, GRAIN, PITCH &amp; FORMANT EXTRACTION</b>	31
<b>Filters</b> .....	31
Low Pass / High Pass .....	31
Filter Banks and ‘Q’ .....	31

Filters of varying harmony .....	32
Filters by spectral subtraction .....	33
<b>Wavelet Distortion</b> .....	33
<b>Grain</b> .....	34
<b>Pitch &amp; Formant Extraction</b> .....	34
 <b>Section 8: ASSEMBLING SOUND MATERIALS</b>	 35
<b>Mixing</b> .....	35
Purpose .....	35
Sound Mixing .....	35
Creating the <i>mixfile</i> .....	35
What is inside a <i>mixfile</i> .....	36
<b>Running the Mixfile</b> .....	37
Levels .....	37
Testing part of the mix .....	37
Modifying the mix .....	37
<b>Moving Between Mixing and Sound Modification Operations</b> .....	38
Extracting a sound from the mix .....	38
Replacing a sound in a mix .....	38
Adding sounds to an existing mix .....	38
<b>Constructing an Entire Piece</b> .....	38
 <b>Section 9: SOME FINAL REMARKS</b>	 39
<b>Finding things on the Workspace</b> .....	39
<b>Keeping the Workspace Manageable</b> .....	39
Removing files which are already backed up .....	39
Backing up newly generated files .....	40
Creating a new directory .....	40
Backing up files to an existing directory .....	40
Moving files to different directories .....	40
What to do with ‘intermediate’ files .....	40
<b>Remembering Your Best Ideas</b> .....	40



# A Beginner's Guide to the *Sound Loom* & CDP Sound Transformation Software

by Trevor Wishart

## Section 1: INTRODUCTION

### General

#### Spelling

The *Sound Loom* has some spelling quirks you need to remember.

Avoid filenames **and directory names**, which contain **spaces**. Although in many cases the *Sound Loom* will handle these, in particular situations it will not understand. To avoid problems, a name like “Alice Smith” can be changed to “Alice\_Smith” or “Alice-Smith”.

Never use “.” And avoid “#” in filenames.

The *Sound Loom* ignores case, so all filenames will be translated to lower case.

#### Naming

As you work you will produce lots of finished sounds, even more intermediate soundfiles, and probably analysis files, frequency files, formant files etc. together with various textfiles which, for example, control how your process parameters change through time.

As a matter of good working practice, you should give your sounds names that remind you of what they sound like. Once you have hundreds, or even thousands of files (which you can easily generate in a few days) you will quickly forget which sound is which if the names are not descriptive. A date-time, or a pure numbering system will not help you.

#### ... and Backing up

The sounds you make appear on the *Sound Loom* Workspace (explained below) as without a directory path e.g.,

“my\_sound.wav”

rather than e.g.,

“my\_directory\my\_subdirectory\my\_sound.wav” .

As a matter of good working practice, at the end of a session, or even during a session (if you have many files), you should move the files that you really want to keep to named directories (explanation of how to do this can be found below). You can use the names of directories to help sort your sounds into types or categories which are useful for your composition, e.g. “lowbells”, “strange\_voices” etc. You can always move sounds to new directories, or rename your directories at a later stage (explained below).

**Always keep important intermediate files**, so that you can go back and rework your material at a later stage.

## Getting Started

1. Double Click on the Sound Loom icon on the desktop.
2. If you are on a multi-user system you will be asked to enter your password.
3. On a new system, various “New User Help” screens will appear as you work. These give useful information about each page. You can eventually get rid of these (there is an “Abandon New User Help” button on each of these pages – you can restore “New User Help” from the “System State” menu on the Workspace.

## The Workspace

### The Workspace & ‘Help’

1. The first useful screen to appear is labelled (top left) “Workspace” and has 3 panels.
  - a) If you have worked on the system previously and you have created files (or loaded files onto the Workspace – see below), when you start your session, these files will be reloaded. If there are lots of them you may see a small vertical strip asking you to WAIT as it loads these files.
  - b) If you have previously written things in the “Notebook”, your last notebook entry will be displayed.
2. At the top left of the page is a button marked Help. If you click on this button, it disables the Workspace, if you then click on any Workspace button, a brief description of what it does will appear in the horizontal window at the top. Click on the ‘Help’ button again (it is now labelled ‘Quiet!’) to re-enable the Workspace. Most large pages in the Sound Loom have a similar help button in the same place.

### The Workspace Directory Listing Panel (Right)

1. The Directory Listing Panel on the **right** can be used to list any directory on the computer (or any network connected to the computer). To list a directory, go the **Find Directory** button above it, and select **Any Directory**. A panel will appear with small yellow folders, and their names. These are the directories on the main hard-disk drive. To select a different drive, go to the button at the very top labelled **Drive Selector**, click on it, and select one of the other listed drives. To locate a subdirectory, **double** click on one of the yellow folders, and its subdirectories will be listed (if it has any). To select a directory, click on it, and its name will appear in the box below. Then press **Select** (bottom left). The window will disappear and you will return to the Workspace.

**NB** The first time you select a directory in your session, you need to press the **List** button above the Directory Listing Panel, in order to see the contents of the directory you chose. After this, directories are listed automatically.

2. Once you have selected any directory, it is remembered by the Workspace. To list it again, go to the **Find Directory** button, and select **Recent Directories**. The most recent directories you have used are listed there. If you click on one of these, it is listed in the Directory Listing Panel. (Remember to use the **List** button if this is the first listed directory of your session).

## The Workspace Panel (Centre)

1. The central panel is the Workspace proper. Sounds (and other files) which you generate will be listed here. New sounds appear **at the top** of the Workspace. These sounds will appear **with no directory path** e.g., “my\_soundfile.wav” rather than “my\_directory/my\_soundfile.wav”
2. The Workspace is also the place to put the sounds (or other files) which you are going to work with in the session (you can add files to it as you go along). You can either **Grab** files, or **Copy** them. Copying a file makes a copy of your file in the Workspace directory (it will appear with no pathname). Grabbing files just lists your file (**with its pathname**) on the Workspace (no copy is made).
3. The Workspace is an **intelligent** listing. When a file is listed here, the *Sound Loom* interrogates it to find for example, the sampling rate, the number of channels, the duration, the loudness and any other properties of the file which might be useful to know about. (For this reason, very long files may take a little time to get onto the Workspace).
4. To **Grab** or **Copy** files to the Workspace, highlight them in the Directory Listing Panel, then go the **Grab** button above. Choose **Use on Workspace** to grab the file. Use **Copy and Use Copies** to copy the files.
5. You can **Remove** files from the Workspace **without deleting them**. (You can only do this with files which have a pathname). Select the files you want to remove, then go to the **Selected Files Only** button and choose **Clear Files from Workspace → Files which are already backed up**.
6. You can delete (i.e. destroy) files on the Workspace. Select the files, and, on the **Selected Files Only** button, choose (at the bottom) **Destroy any selected files**. You will always be asked to confirm that you really want to do this before the file is destroyed. Once a file is destroyed it cannot be retrieved!!
7. You can **Rename** a file. Select the file on the Workspace. Go to the **Selected Files Only** button. Choose **Rename → Rename**. Then type the new name in the box which appears.  
**NB** when you return to this menu, the ‘Rename’ function will be listed at the top of the menu, so you don’t have to search for it again. This applies to all the functions on this and other menus.
8. You can **Move files to a different directory**. Select the file on the Workspace. Put the directory name you want to use in the directory name box above the Directory Listing Panel. On the **Selected Files Only** button, choose **Move to New Directory**.

## Backing up your files.

9. To **back up** files (i.e. to put new files you have made into a directory), select the files on the Workspace. Select the directory you want to store them in, in the directory box above the Directory Listing Panel. You can
  - a) create a new directory by typing in its name.
  - b) select an existing directory via the **Find Directory** button.

- c) Create a new subdirectory by adding the subdirectory name to the directory name in the box.
- d) Then press **Backup selected New Files**. If you select **Store**, the file will be moved to the new directory, but will remain on the Workspace, with the directory (path) added to it. If you select **store and clear** it will be moved to the new directory and removed from the Workspace listing.

## The Chosen Files Panel (Left)

In order to process a sound you have to list it on the Chosen Files Panel (left). You may also want to process 2 or more sounds at once (e.g. to join them together by editing).

1. To add sounds to the Chosen Files panel, press the **Enter Chosen Files Mode** and the Chosen Files panel will change to the same colour as the Workspace panel. (If it is already the same colour, you are already in “Chosen Files Mode”). You can return to “Workspace Mode” by pressing the same button (now labelled **Return to Wkspace Mode**).
2. **In “Chosen Files Mode” .....**
  - a) selecting a file on Workspace Panel adds it to the Chosen Files Panel.
  - b) Selecting a file on the Chosen Files Panel removes it from that panel.
3. In Version 11 onwards, you can add files to the Chosen Files list without going into or out of “Chosen Files Mode” by using the buttons below the Workspace.
4. To process the file, click on the button **Process** above, top left.
5. It is also possible to apply the **same process** to many (single) files. To do this, select **Bulk Process**.

## The Process Page

Pressing one of the ‘Process’ buttons takes you to the process page. On this page there is an array of buttons behind which are the many sound processing routines of the CDP. Clicking on any button which is highlighted will show you a menu of processes (on the left). Only the processes which work with the particular kind of sound(s) you have selected will be highlighted (and active) here. If the process you want is not active, check you have selected the correct sound on the **Chosen Files List** on the Workspace (some things only work with mono files).

1. **To run a process**, select a process from the menu (or its submenu).
2. To find out what the processes on a menu do, press **Info** (above, middle). That button will be relabelled “Action”. Then select one of the array-buttons.
3. To find more details about a specific process, stay in “Info” mode and select **process** (above). Then press any array-button and select a program as if you were about to run it. A more complete explanation of the process will be displayed.
4. To get out of Information mode, press the original button (above, now labelled **Action**).

If you do not know which process to use, or where to find it, press **Which?** (above) and type in what you want to do (e.g. “time stretch”). Then, on the right, select “relevant” (i.e., relevant to

the particular file or files you have chosen to process) or “all”. You should now see a display of processes which do what you describe, or something similar, associated with the names of the menus on which you will find them. If you see nothing, try a different description.

## The Parameters Page

Selecting a process will take you to the Parameters Page. Here are displayed the **Parameters** of the process. You can find out what the parameters do by pressing the **Information** button.

You can (usually) enter parameter values in 4 ways.

1. Type a number into the number box (left).
2. Slide the slider bar.
3. Create a **control file** (see below) which describes how the parameter changes in time (button **Make File** on the right).
4. Get a file you have already created, which describes how the parameter changes in time (button **Get File** on the right).

The values you type (in the box or in a file) must be within the range (see left of parameter bar) or the Loom will complain. You can, however, often **extend the range** by pressing the **Range** button, on the left.

You can play your original sound by pressing **Play Source** (middle left, above). You can **see and play** the source from the **Sound View** button (middle, above) (more information below).

To run the process, click on the **Run** button (top left) (or hit **Escape**).

## The Run Page

Pressing the Run button takes you to the Run page. This page will display any messages which the process sends out. To run the process, click on the **Run** button (top left) (or hit **Escape**). If the process takes a long time you will (usually) see the time-bar advance at the bottom of the screen. If there are any messages from the process these will be displayed, and you must press **OK** (top left) to return to the Process Page. If no messages are displayed you will return automatically to the Process Page.

## Back at the process Page

Having run a process you will find yourself back at the process page. You will now find that the **Play** and **View** buttons on the top **right** have become active. You can simply **play the output** with the Play button. You can **see and play** the output with the View button.

You can also play the source sound from the **Play Source** button (top left) for comparison.

If you do not like the result, you can change the parameters and try again.

## Saving the output

Once you have created a file which you want to keep you **must give it a name**. If you do not, it will be deleted.

To save the file, press **Save As**. If no messages are displayed you will return automatically to the Process Page. The **Save List** will appear. You can name the file by

1. typing a name into the name box.
2. Selecting (by clicking on it) the name of the input file, listed in the 2<sup>nd</sup> panel below, under **Recent Source Names** and **modifying** the name (by typing extra characters or deleting some). **You cannot overwrite the input file here** (you can do this later if you want to).
3. Selecting a name you have used recently (and perhaps modifying it) from the lowest panel, labelled **Recent Names**.
4. Selecting a temporary name by pressing one of the buttons **a1** or **a2**.

If you select the name of an existing file, you will be asked if you really want to overwrite the existing file (as using the same name will destroy the existing file).

## What to do next

Once you save the file, the Play and View buttons will become inactive again. At this point you have 4 options.

1. Run the process again with different parameters.
2. Process the same file with a different process. Clicking on the **Get New Process** button will return you to the Process Page.
3. Process the *output* file. Clicking on the **Recycle Outfile** will also return you to the Process page, but you are now working on the (last) output file of the process you just ran.
4. Select a new file to process, by returning to the Workspace. To do this, click on the **To Wkspace: New Files** button.

## Control Files

A powerful feature of the CDP is that most parameters of most processes can be made to vary in time using a Control File, usually called a `breakpoint file'. These files consist of 2 columns of numbers e.g.

0	1
1	.3
3.456	.4
6	0

The first column is the time in seconds. The second column is the value of the parameter.

1. Times must be greater or equal to zero, and must increase. To make a sudden change, use two times that are nearly equal e.g. 3 and 3.0001
2. Values must be within range.

### To make a breakpoint file

- a) select the **Make File** button to the right of the parameter bar. A window will appear in which you can type values.
- b) In earlier versions of *Sound Loom*, if a small window appears asking you whether you want to use Graphics or Text, you should, perversely, select “Always Use Text”, in order to access the new ‘*Sound View*’ facility – see below.
3. You can create textfiles directly on the Workspace, or outside the CDP in a text file editor like *Notepad* or *WordPad* or even *Word* (**so long as you save them as Plain Text**).
4. In many cases, however, you will find a **Sound View** button at the top of the page. Clicking on this button produces a graphic display of the sound (from which you can play it). You can draw breakpoint values onto the sound display using the mouse (see instructions on screen).
5. You must then press **Output Data** to use the data you create. Once you **close the window** your values will appear as text in the text window.
6. Give your data a name, by typing a name into the **filename** box (top).
7. **Save** the control file. (Button, top left).

At this point (provided the range is correct etc.) you will return to the Parameters Page and the name of the file will appear in the relevant parameter box.

You can also **get an existing file** or **edit an existing file**.

1. Choose the **Get File** button. This brings up a window which lists all textfiles which could possibly apply to this particular parameter (the *Sound Loom* knows what the data range all textfiles is, and also what the range of the parameter you are using is ... it only lists relevant textfiles).
2. You can select a file with the mouse and click on **Use** (top left) which returns you to the parameters page where the file will now be in the entry box of the parameter.
3. Or you can click on **Edit** (top) which takes you to a page where you can edit the file, and save it (possibly with a new name, if you want to keep the original version). You will then return to the list of files and (**if it has the correct range**) your new file will appear in the selection list.
4. You can also get a file which is not on the list by pressing **All textfiles**. If you now select a file from the list and edit it so that it lies within the range of the parameter, save it, then select **See Possible**, your new file appear in the files for selection.....

### Sound View

Sound View buttons (usually pale green) allow you to

1. See the soundfile.
2. See its spectrum (press the **spectrum** button at the bottom)
3. Zoom in, timewise, down as far as the sample level.

4. Play the soundfile or any part of it - use the mouse to select areas (see instructions on window).
5. If the display has an **Output Data** button (bottom right) you can generate parameter values from lines, boxes or graphs you draw on the screen (see instructions in window), then output them directly to parameter boxes or to control files, by pressing the Output Data button and then **closing the window**.

You must close any existing Sound View window before you can end your *Sound Loom* session.

## Remembering and Recalling What You Have Done

The Sound Loom has several mechanisms to remember and recall what you have done. These are a permanent record, remembered from one session to the next and only deleted if you choose to destroy them.

### At a local level.

1. The previous process you ran can be recalled, using the **Use Process Again** button on the process page.
2. The previous parameter values used (even if these were used in a previous session) can be recalled using the **Penultimate Run Vals** button on the Parameters Page.
3. The last item selected on (many of the) menus is placed at the top of the menu, so that, when you go to the menu again, it is quickly retrievable.

### At a global level

1. The Workspace Listing and the Chosen Files Listing at the end of your session are remembered and restored when you start the next session.
2. Every process you run is stored permanently in a dated **log file**. You can recall any process from the current or previous sessions. If you have a crash during your session you will lose the log of that session, and possibly the sound you were creating at that moment time. Nothing else will be affected.
3. You can save a particular configuration of parameter values which you have used, and you wish to keep, as a **Patch**.
4. If you have a favourite process you can save it as a **Favourite**.
5. If you have a particular group of processes you want to save as a unit, you can do this as an **Instrument** or as a **Batch File**.

### Log files

As you work, the system keeps a record of what you do. You can **recall a previous process** from the current session, or any previous session. On the Workspace page, click on **Recall Action**.

1. If you have run some processes, you will see them listed. To recall a process select it and press Recall Action. Provided that the source sounds and any control files still exist you will be taken directly to the parameters page ready to rerun the process exactly as before (with the

same parameters). (If any files are missing you will be asked to provide substitutes, or allowed to abandon the recall).

2. To recall an action from a previous session, press the **Get Log** button, and choose **Previous Log**, **Next Log**, or **Choose a Log**. In the latter case you will be presented with a list of date-stamped log files to choose from.

After a number of sessions you will be asked if you want to delete any of your log files. In general you should answer **No** to this enquiry, unless you have a specific reason to delete a particular log.

## Patches

On the parameters page, once you have entered your parameters, go to the right hand panel (**Patches**). Type a name for the patch in the box above, then click on **Save**. The patch will be listed in the panel below and will be available whenever you run this process again. To load the patch, simply select it with the mouse, and click on **Use**.

## Favourites

Once you have run a process and saved the output, you can save it as a **Favourite**. Go to the process page, and on the right hand panel labelled **Favourites**, select **Add Last Process**. The process will be listed in the panel below. You can now select it with the mouse and click **Use** to run it.

## The Calculator

On Several pages you will find a **Calculator** button. The calculator enables you to do simple mathematics, but, more importantly, to convert between different musical units. To use it:

1. Enter values into the **Value (V)** box by clicking on the buttons in the panels on the left.
2. As you do so certain buttons on the **Input Units** panel will be highlighted. Select one of these e.g. MIDI.
3. When you make that selection, a number of buttons on the **Output Units** will be highlighted. Select one of these, e.g. FRQ.
4. The calculator will calculate the frequency corresponding to the MIDI value and place the result in the box on the bottom right.

**If you call the calculator from the Parameters page....**

1. You can get a (numeric) parameter value into the Calculator value box using the **Get Param** button (below the Numeric panel, but only when called from the Parameters page).
2. Insert the Calculator output into a parameter value box on the Parameters page, by pressing the **Use as Param** button (top right).

## The Table Editor

This is a powerful tool for editing tables of values in textfiles (see p.29).

## Section 2: CLEANING UP SOURCES, & USING EDITING TOOLS

### Cleaning and shaping your sources

The *Sound Loom* processes sounds you have recorded or copied onto the hard disk. You should begin by ensuring your sounds are cleaned of any extraneous noise or silence, and that they are at maximum level (you can always reduce their level when you need to).

#### 1. Get rid of silence at the start and end of sounds.

Use **Housekeep** → **Select & Clean** → **Top & Tail**. This cuts off silence at the start and end of your file. You can see where there is silence (or low level noise) using the **Sound View** button. The silence does not necessarily have zero level so you have to tell the program at what level it should reject sounds by entering a value between 0 and 1 in the **Gate Level** parameter. A value of 1 will gate out (delete) the entire sound. Try zero first. You can see what effect this has using the **View** button (top right). If it this does not cut away all the ‘silence’, try a higher gate level … if it cuts off too much, try a lower value.

#### 2. Keep a specific portion from a file.

**Edit** → **Cut out and keep**. Look at the sound with **Sound View**. Use the mouse to select a portion of the sound, and **Play** the portion to check it is what you want. Press **Output Data** and then **Close the Sound View** window. The edit parameters should now have been written in the boxes on the parameters page. Run the process, and save the output.

#### 3. Cut recording with many events separated by ‘silence’ to separate files.

Use **Housekeep** → **Select & Clean** → **Gated Extraction**. The important parameters are

- a) **Gate Level** and **Endgate Level**, which do the same job “Gate Level” as in “Top & Tail” but apply to each segment that is cut from the source.
- b) **Threshold** which means, how loud does the chunk have to be before I keep it. This avoids keeping low level bits of noise (unwanted paper rustling, footsteps etc.).
- c) **Minimum Duration**. This sets the minimum length of any chunk you want to keep. Avoids keeping short transient events (kicking the microphone stand!).

#### 4. To shape the ends of recordings which begin or end suddenly.

Use **Envelope** → **Dovetail** and set the duration of the dovetails (fade in and fade out) of your sound.

#### 5. To Shape the end of your sound.

**Envelope** → **Curtail** allows you to define the start and end of a fade-out for your sound.

Both the above programmes come in “normal strength” with a choice of “linear” or “exponential” (steeper), or “double strength” (even steeper).

Cleverer ways to clean up sounds can be found on the **Music Testbed** → **Cleaning Kit** on the Workspace page.

### 6. Normalising your files.

Always try to work with materials at maximum level. You can boost the level of your materials using **Loudness → Normalise** (and keeping the default level of 0.9). Bear in mind that, if you have made a very low level recording (e.g. by microphone), you cannot *improve* it (i.e. boost the level of the signal relative to any background noise) by normalising it, so always be sure to make your initial recordings at as high a level as possible (but without distortion).

## Section 3: EXTENDING YOUR SOURCES

### Time-stretching

1. Always try playing the sound at a lower “tape speed”.

**PITCH:SPEED → pitch → tape transpose by semitones.**

This process either raises the pitch while making the sound shorter, or lowers the pitch by making the sound longer. Slowing the sound down often reveals complex gestural detail (and lowering the pitch can reveal complexity hidden in the upper reaches of the spectrum).

2. You can also slow down or speed up a sound with **Brassage → timesqueeze**, bearing in mind that to make a sound twice as long you need to squeeze it by 1/2. For more complicated values, use the **Calculator** and calculate **1/V** (far right of Calculator).
3. Alternatively, for a mono sound, you can use **spectral time-stretching**.

- a) Select **PVOC → analysis** and keep the default values of the parameters. This creates a file of the time-changing spectrum of the sound.

If your sound is stereo, but doesn’t need to be stereo, use **Channels → extract/convert channels → stereo to mono** to convert it to mono. Alternatively you can separate the 2 channels using **Channels → extract/convert channels → extract all channels**).

- b) When you submit the output file from the Phase Vocoder to the process page, you will find that the processes below PVOC (the spectral processes) are now highlighted (whereas many of the processes above, which apply only to sound files, are no longer highlighted).
- c) Select **Stretch → time → do time-stretch**. Enter a time-stretch value (here, 2 does mean make it twice as long). Recycle the output.
- d) Select **PVOC → synthesis**. You can now hear the output.

### Looping

1. **Extend → loop → give loop repetitions**. The loop length will need to be slightly shorter (at least) than the duration of the sound. The **loop length** determines the duration of the repeating sound-segment. You can make the loop length shorter than the sound then get it to advance gradually through the sound on each repetition by setting a value for **advance between loops**.
2. A more ‘naturalistic’ looping can be obtained with **Extend → iterate**. Here you must specify the **delay** between one entry of the sound and the next (effectively the same as the loop length, but now in seconds, not milliseconds). But here you can slightly or completely randomise the timing (making it slightly or very irregular) with **randomisation of delay**, make the repeating elements differ in pitch either slightly or crazily (specified as a maximum range up & down, in semitones, by **pitch scatter**), and differ in loudness (**amplitude scatter**).

## Zig-zags and Drunken Walks

3. We can extend a sound by reading it forwards and backwards to and from random places in the file (or places that you specify in a data file). This does not change the pitch of the sound. Use **Extend → zigzag → random** (or “user specified”). As there’s no reason for this process to end, you must specify a **minimum duration**, as well as a place (time) in the source to start and end zigzagging. You must also specify the maximum (**max zig length**) and minimum (**min zig length**) of the zigs/zags. Remember that in the places where the sound is read backwards it may be radically altered.
4. We can extend a sound by leaping around in it, but always reading it forwards in time. This does not change the pitch of the sound. This is **extend → drunkwalk**.
  - a) The locus defines where we are in the file – this can be at a specific time, or defined by a **breakpoint file** to move forward or backwards, or any which way, through the file at any, possibly varying, speed. You can even read from beginning to end (say 0 to 10 secs) in the exact duration of the file (10 secs) so, with all other parameters set to their defaults you simply hear the original file recreated. Imagine a lamp post to which is tethered a drunken man. If the lamppost moves, it drags the drunken man with it.
  - b) At any locus (say a lamppost) you can stagger away from it, only by a certain maximum amount. Imagine the drunken man is tethered to the lamppost by a rope. The **ambitus** determines the length of the rope.
  - c) Each stagger can be of a certain size, determined by **step**.
  - d) At each step we move to a new place in the file and begin to read from there (forwards). How often we make these steps is determined by the **clocktick**.

All of these parameters can be made to vary through time (using breakpoint files from the **Make File** button). You have to play with this program to get some idea of what it does.

5. In recent versions of *Sound Loom* you will find an extra button on the Parameters Page of Zigzag which allows you to describe the type of drunken motion you want to achieve, and which will calculate appropriate parameter values for this motion.

## Brassage

Brassage is a process that cuts a file into tiny segments, then puts them back together (usually) in order, to reconstruct the file. Depending on just how we cut and reassemble the file we can make the file longer or shorter (discussed above). We can also **pitch-shift** the segments but retain the duration. We can produce **reverberation-like effects**, **granular effects**, and **randomisation of the order** of the sound.

### In (Group) Brassage (Function) Brassage:

- a) **timeshrink** does what it says.
- b) A **density** less than one will introduce silences between the grains.

Changing the **grainsize** makes a large difference. The default (50 ms) works well for time-stretching and pitchshifting. Smaller grains tend to change the sound quality (eventually

making it gritty), large grains produce more of a collage effect, as you hear out the individual chunks you are cutting from the source.

- c) **Pitchshift** does what it says (in **brassage** → **full monty** you can set a maximum and minimum pitch-shift and the grains are transposed **at random** between these two limits).
- d) **Spatial Position** defines left as 0 and right as 1, in the stereo space. It is more interesting to change this over time (breakpoint file).
- e) The **Scatter** randomises the position of the output grains.
- f) **Searchrange** is more radical. Set at zero, grains in the output follow the order of grains in the input. If the search range is large, the output grain is chosen from a search area around the corresponding position in the input file. If the search range is high the grains are being chosen from everywhere in the file up to and including the ‘now’ in the input file, so the sound gets scrambled. This is most interesting with sounds with a definite sequence e.g., speech, melody or any set of disjunct and obviously different events.

## Shredding

Any sound can be ‘scrambled’ using **Radical** → **Shred**. The program cuts up your sound into segments, then shuffles them and reassembles them. You can then repeat the process *on the new sound*, i.e., the already cut-up sound is cut-up again in different places, and so on. As you make more and more shreds the sound is cut into (on average) smaller and smaller fragments, and randomised in order on each shred.

This process works best on sounds with some recognisable initial order e.g., speech or melody, or disjunct events edited together in a specific order.

You can specify how many shreds (i.e. how many times you repeat this shred process), the (average) size of the cut segments, and the randomisation of their sizes. **Always** randomise the sizes (at least a little).

After 1000 shreds, any sound will be reduced to low-level noise.



## Section 4: WORKING ON THE SPECTRUM / MULTI-PROCESSES

A sound can either be represented as a waveform (the shape of the pressure-wave in the air), or as a spectrum (representing e.g. the harmonics in a piano tone). The spectrum tells us where the energy is in the sound (e.g. in speech, relatively high if you hold your nose while you speak, relatively low if you have a heavy cold). The spectrum consists of a number of “channels” corresponding to lower or higher frequencies.

The spectrum contains no time-information, so if our sound changes over time (all non-synthetic sounds change over time!), we need to divide the sound into a series of tiny “windows”, like the frames of a film, but much briefer in time, and specify the spectrum in each of these windows. Each window contains a number of channels (or points) of information.

On the computer, soundfiles (“.wav”, “.aiff”) contain a mathematical representation of the waveform, which you can look at with “Sound View”. You can convert these to spectral files (also called “analysis files”) using the **Phase Vocoder (PVOC)**. You can also look at the (time-changing) spectrum of a sound with *Sound View* by pressing the **spectrum** button (bottom left of the *Sound View* display).

PVOC only works on mono files, but you can split a stereo file into two mono files (1 for each stereo channel) (**Channels → extract all channels**) and work on these, recombining them to stereo once you have finished (**Channels → merge channels**).

### Spectral Processes

Once you place a spectral (analysis) file on the **Chosen Files List** and go to the process page, you will find that most of the buttons below PVOC have become active, and most of those above PVOC are inactive. These active menus contain processes you can apply (only) to spectral files.

#### Inharmonic and Harmonic spectra / Spectral stretching

If we look at the spectrum of a sound with a definite pitch, the peaks of the spectrum will have a simple relation to one another. The frequencies of the upper peaks should be (at least approximately) simple whole-number multiples of the frequency of the lowest peak. So, if the lowest peak has frequency 400, higher peaks should have some or all of the frequencies 800, 1200, 1600 etc. When we say the spectrum is harmonic, we imply that the sound itself has a **single** pitch. (This is not directly related to the traditional notion of “harmony” in Western notated music). If the spectral peaks are not (even close to) simple multiples of the lowest peak, the spectrum is said to be inharmonic. In this case, we may hear several pitches (try striking a suspended metal rod) or a focused but unpitched sound (some drums). A sound may have a sliding pitch, so that the simple relationship between the spectral peaks is preserved as they slide around. The spectrum may also vary e.g. between an inharmonic attack (which we may experience merely as a bang or a bright attack) and a clearly pitched tail, or it may be entirely random (in which case we hear noise).

You can stretch the spectrum of a sound with **Stretch → spectrum**. To understand what this does, begin with a clearly pitched sound (which has a “harmonic” spectrum). You can stretch everything above (or below) a specified frequency. To stretch the whole spectrum select “above”

and set “*Frq Divide*” to its minimum value. **Top of spectrum stretch** determines how much the topmost frequencies of the spectrum are transposed upwards. If the whole spectrum is transposed by this amount, the sound will merely be pitch-shifted. A pitched sound will move up in pitch. If you change the **Stretch Exponent** to a value that is not 1, different parts of the spectrum will be transposed by different amounts. In general this causes sounds with pitch (with a harmonic spectrum) to become inharmonic and several pitches may be heard in the output sound.

## Spectral Tracing

You can choose to keep only the most prominent features in the spectrum, with **Highlight → Tracery** (e.g. → **Trace All**). This process retains only a given number of the most prominent channels (frequency bands) in each time-window of the spectrum. To hear this clearly, enter a complex sound, and choose the value “1”.

With a complex sound, which channels are the most prominent changes from moment to moment, so trace can have the effect of selecting ‘melodies’ of partials from your original sound. This effect is heightened if you first spectrally timestepstretch the analysis-file (**stretch → time**).

## Spectral Blurring

In contrast, the process **Blur → Blur** keeps some windows (the “film-frame” time-slices) and throws away others. If you enter the parameter “16”, only 1 in every 16 windows is retained. The intervening windows are replaced by values which glide between the values in the retained windows. In this way, lots of information about how the spectrum changes through time is lost. With a spectral blur of around 32, human speech sounds slurred, as if the speaker is drunk. Much longer blurs turn speech into semi-metallic gliding inharmonic sounds. You can also do both with **Highlight → Blur & Trace**.

## Spectral Freezing

**Focus → Freeze** will grab the spectrum at specified moments and hold it until the next time-marker is reached. You have to enter a special file of times with “before” and “after” markers. See the **Information** button on the Parameters page.

You can also capture a single spectral window and create a sustained sound from it. Use **Simple → Magnify Window**.

## Imposing Vowels

**Highlight → impose Vowels** can be used to impose sequences of vowel sounds on a spectral file. This will only work if you begin with a relatively noisy or “rich” sound (a sound that has energy all over the spectrum, like a large pack of barking dogs) – a pure sine-wave will definitely not work. The process uses a special data file consisting of “time – vowel” pairs. (For more information, try the “*Information*” button on the Parameters page, or the **Standard Features** button on the text-editing page that appears when you press “*Make File*”). Note that the vowels are interpolated, so if you enter

u  
ee

you produce a slow diphthong sliding from “u” to “ee” over 1 second.

## Morphing

You can morph from one sound into another using **Morph → morph**. This is not as easy as it sounds and depends very much on the nature of the two sounds you choose (e.g. if they are clearly pitched, they should be at the same pitch), the duration of the morphing process (if it is too fast, no morphing transformation will be experienced), and the morphing parameters you choose to use. A morph will obviously require two sounds to be selected on the Chosen Files list.

In a morph, the values in the spectral windows of one sound are made to move gradually towards those in the spectral windows of the 2<sup>nd</sup> sound, until the spectrum becomes entirely that of the 2<sup>nd</sup> sound. The channels of the spectral windows contain both Frequency and Amplitude (loudness) information, and you can slide the amplitude values, and the frequency values (of the 2 spectra) independently. (Specify the starting and ending time of these slidings).

The exponent(s) determine the shape of the trajectory of the morph (cosinusoidal goes slowly at first, fast in the middle and slowly towards the end). Changing the exponent warps this trajectory (experiment to find the best for your particular sounds).

## Multi-Processes (“Instruments”)

As PVOC takes only mono files, to work on a stereo file you would first split it into two mono sources, then do a (PVOC) analysis of each of these. These 3 processes can be combined as one multi-process, called an **Instrument**. In fact any number of processes can be combined in this way. But remember, **instruments cannot be edited**, so design them carefully in the first place, possibly combining just a few processes in each.

To make this stereo analysis instrument, we would first put a stereo soundfile on the Chosen Files list (on the Workspace page), then select **Instrument** (top centre) instead of “*Process*”. This takes us to the **Instrument page** where our selected soundfile is shown on the left. (We can select other sounds from the Workspace with the **Choose Files** button here).

We then press **Make Instrument** (top left) which takes us to the Process Page where we proceed as normally. In this example case we select *Channels → Extract All Channels* and run it. Once the process has run we find that the “*Save As*” button has been replaced by a **Keep Process** button. If we press this, it returns us to the Instrument Page, where we see a **colour-coded diagram** of the input and output files, and the name of the process used.

We can now select a new file to process. In particular, we can select an **output file** (or an input file) from the last process by **clicking on it in the diagram**, (when you will see it appear in the list on the left). In our example there will be 2 output files shown on the diagram, and we click on one of these to select it.

We then press **Continue** (above left) which takes us back to the process page, where (in this example) we can choose PVOC, to analyse the sound .... and so on. In this way we can assemble as many processes as we want into a single Instrument, whose colour-coded diagram will appear and grow on the Instrument page.

When we have finished with this chain of processes, we select **Conclude**, (instead of “*Continue*”). This will return to the Instrument page where we will find (on the left) a list of all the files we have made (plus the diagram, so we can tell which file is which). Some of these will definitely be **output files** (i.e. those we want to keep) while others will be merely **intermediate files** (i.e. those we do not need to keep).

To complete the process we must

1. Name **at least one** file from the list, using **Save As**. In fact we must name all the files that we want to keep (output files). Any files we do NOT name will be assumed to be intermediate files, and will be deleted. This also tells the instrument we are making that, when we run it again, these intermediate files made during the various processes will *always* be deleted.
2. Press **Conclude** (top right), which takes us to a new window where we must...
3. Enter a **name for the instrument**.

If we now return to the Process Page, in the panel on the far right (“Instruments”) we will see our newly-named Instrument listed. Clicking on “See” will show the colour-coded Instrument diagram. Clicking on “Run” will start the instrument (so long as we have selected the correct files on the “Chosen Files List” on the Workspace Page).

When you run an instrument it looks just like a normal process, except that (at the bottom left) there is a box in which appear, in sequence, the several processes contained in your Instrument.

## Instruments with parameters

If we make a 2<sup>nd</sup> instrument to time-stretch the two analysis files and then resynthesize them into a stereo soundfile, we would use the process **Stretch → Time Stretch** (twice – once for each analysis file) followed by **PVOC → synthesis** (twice), then (selecting both the soundfiles synthesized) **Channels → merge channels** (to reconstruct a stereo file). However, when we use the completed instrument on other occasions we will probably want to vary the value of the time-stretch parameter. We therefore need “*time-stretch*” to be an accessible parameter of our Instrument.

When you are building any Instrument you will notice, at the right hand side of each parameter display, a new small box labelled “**variable**”. If you tick this box, that parameter will appear on the parameter page of the completed Instrument when you run it, and you can enter a parameter value there. In contrast, you may always use the PVOC process with the parameters (e.g.) Analysis Points = 512 and Analwindow Overlap = 3. Therefore, you would prefer not to enter these parameters every time you use the Instrument. So enter these values when you are *making* the Instrument but do **not** tick the “*variable*” boxes for these parameters. These parameters will then **not** appear as variables of the instrument. (Your standard values - 512, 3 - will be assigned behind the scenes).

## Section 5: LOUDNESS ENVELOPES / SEPARATING & RECONSTRUCTING A SEQUENCE OF EVENTS

### Envelopes

The loudness “envelope” (or contour) of a sound can be extracted and used. For example you can shape the loudness contour of one sound with the loudness contour of another.

#### **Extracting the envelope**

The loudness envelope of a sound is extracted using **Envelope → extract**. You can extract the envelope as a textfile (so you can read and edit it) or as a “binary file” (which you can’t see or edit, but has some other uses).

The most important parameter is the **Windowsize**. With a small window the envelope “sees” the small loudness fluctuations in a sound. For example with a barking dog which has growly barks, a small windowsize will see the “grain” (the “rrrr”) of the individual barks. If you just want to pick out the whole bark events, you would use a larger *windowsize*, which would not see the “rrr” grain.

When you run the program, if you use the “text output” option, you can see the output (either as list of values, or sometimes as a graph) by pressing the “Read” button (top right of the parameters page). As usual, to save the envelope you must “Save As” and give it a name.

#### **Imposing the envelope**

You can now impose this envelope on another sound. There are 2 ways to do this, **Impose** and **Replace**.

Imagine you have two recordings which both change between being quiet and loud. If you **Impose** the envelope of one on the other, if a quiet bit in the envelop of the 1<sup>st</sup> sound corresponds with a quiet bit in the 2<sup>nd</sup> sound, the output there will be **doubly** quiet, i.e., the low level in the envelope is applied to the low level in the 2<sup>nd</sup> sound, so the output is even lower in level. If you **replace** the envelope, the process **forces** the level of the 2<sup>nd</sup> sound to the level of the 1<sup>st</sup> - so at our quiet point the output is only as quiet as the envelope. (This can produce anomalies at points where there is zero level in the 2<sup>nd</sup> sound).

You can impose one envelope on another by placing the sound you want to alter on the “Chosen Files” list (on the Workspace), followed by the envelope you want to apply to it, then, on the Process page, selecting **Envelope → Impose → env in textfile**.

You can also do this whole process in a single manoeuvre. Place the sound you want to alter on the Chosen Files list, followed by the **sound** whose envelope you want to use. Then, on the process page, chose **Envelope → Impose → env from other sndfile**.

It is possible to modify envelopes either by editing them (if they are textfiles) or using various processes (**envelope → replot**, **envelope → reshape**).

## Separating and Reconstructing a Sequence of Events

Imagine you have a sequence of footsteps in a reverberant passage and you want to change the rhythm of the events. You must first separate the footfalls.

### Saving the timings of the “footfalls”

First however, it's useful to know the timing of the footfalls.

Put the recording on the “Chosen Files” list, then go to the **Table Editor**. Select the **Button Tables** and choose **Create Data → Create Ouput Data from Soundview Display**. This will show you a plot of the waveform of the sound. You can play this or any part of it (drag out a playbox with **Shift : Mouse-drag**) to find the footfalls. You can mark the start of a footfall by clicking the mouse. Once you have marked the position of all the footfalls, press **Output Data** on the Sound View display. Back at the Table Editor, the times of the footfalls will appear in the large panel to the right, below.

To save these times, enter a name in the **Enter Name** box on the right, then press **All**. Quit the Table Editor, and your list of times can now be found (in a text file) at the top of the Workspace. Let's call this file “walkrhythm”.

### Separating the “footfalls”

We can now separate the footfalls. With the recording on the Chosen Files list, go to **Process** and select **Edit → cut out and keep many → time in seconds**. On the parameters page, select **Make File** (to right of the parameter bar). On the text-editing window which next appears, select the **Sound View** button at the top.

On the Sound View display of the waveform, you can select an area of the sound (Shift-Mouse Drag) to play. Try to select the whole sound from the very start of a footfall to just before the very start of the next. Once you have selected exactly what you want, press **Output Data**. Then proceed to select the next footfall, and so on.

When you have selected all the footfalls you want, close the window (use the red button, top right). The times of the selected portions will now appear in the text window. You must save this list of values by giving it a name (in the box, top right), and pressing **Save File** (top left). Once you do this you will find yourself back on the Parameters page, and the filename will have been entered in the value box of the parameter **Segment start and End Times**. You can now run the program.

As this process outputs several files, when you “Save As” you will be asked “**Generic Output Name?**” i.e., do you want to give all the files the same (generic) name. If you decide to do this, and choose the name e.g. “footfall”, the output files will be named “footfall0”, “footfall1”, “footfall2” .....

### Dealing with reverberation

If your recording was very reverberant, cutting the footfalls in this way will probably produce an artificial sounding cut-off at the end (where the persisting reverberation suddenly ends). You can get around this in two ways

1. Add further reverberation to mask the existing reverb (and extend the sound past the silent end). This can be done using **Reverb : Echo → Rev/Echo → stadium**, followed by mixing the reverberated output, at low level, with the original sound. (Note that the reverb output is stereo). The simplest way to do this is with a *mixfile* (see below).
2. Fade out the existing reverberation using **Envelope → Dovetailing → double strength**. This fades out the sound at its end, so it falls to zero before you reach the abrupt cutoff at the end.

## Reconstructing the original sequence

You can now put the original footfalls back together. Select them all, in correct sequence, on the Chosen Files list. On the process page select, **Mix → create mixfile on timegrid**. Then, on the parameters page, select **Get File**, and from the list provided, choose your file “walkrhythm”. Running this program will create a **mixfile** for your footfalls, using the timings in “walkrhythm”.

(A *mixfile* is a set of instructions on how to mix several soundfiles. It is a textfile, and can be edited).

Save the *mixfile* output, and **Recycle** it (or just place it on the Chosen Files list), and, on the Process Page, choose **Mix → Mix from Mixfile**. Run the process, and check that it has **not overloaded** by pressing the **Maxsamp** button (top right). If the mix has overloaded, the **Attenuation** parameter (on the parameters page) will have been reset to a value less than 1. In this case, run the mix again.

The output sound should be your original sequence of footfalls.

## Modifying the rhythm

You can create new sounds with the same footfalls, but with different rhythms, by changing the times specified in the mixfile. You can directly edit the mixfile (on the Workspace **Selected Files Of Type → Any Textfile → Read/Edit Selected File**) but there are more powerful ways to do this using either the **Table Editor** (see below) or the **Qikedit** button which you will find on the parameters page when you run “**Mix → Mix from mixfile**”.



## Section 6: SPACE, TEXTURE TABLE-EDITOR & CALCULATOR

### Space

A mono soundfile can be made to move around the stereo file by describing its movement in a “panfile”.

#### Select the process Space → spatialisation → pan.

On the parameters page there are 2 parameters. **Pan position** is defined by a number. -1 puts the sound in the left speaker only, (+)1 in the right only, and 0 locates the sound in the centre between the loudspeakers. Values less than -1 (or greater than 1) gradually attenuate (turn down) the sound in the left (right) speaker so it appears to be even more distant.

When you pan a sound across from left to right, there is a tendency for the sound to appear loudest when it is in a single loudspeaker (-1, or 1). The pan function compensates for this by making the sound grow louder as it approaches the centre (0) position. If your sound is already very loud this may cause it to overload. The **prescaling** parameter allows you to turn down the input sound to avoid overload (clipping). Always try this at value 1.0 (no attenuation) first.

If your sound moves back and forth between the loudspeakers in some way it's often best to avoid values of (-)1, so that the sound is never entirely in one loudspeaker (until, perhaps, the very end of a motion trajectory where the sound might “exit” left or right).

A panfile is a list of times and values (see above) describing how the position of the output sound changes over time. A panfile which crosses the space once per second might look like this

0	- .9
1	.9
2	- .9
3	.9
4	- .9
5	.....

The position of the sound at other times (e.g. .5) are **interpolated** from the values in your table. So, in the table above, at time .5 the position will be 0.

#### Depth and other motion clues

To create a sense of motion away from the listener or towards the listener we can use a number of “motion clues”. As sounds recede they get quieter, so we can create the illusion of movement away from the listener by gradually turning down the level (use an envelope e.g.). Distant sounds also have less high frequency information, so we might use a low-pass filter to help with this. Distant sounds, *in some environments only* (e.g. forests), are more reverberant, so we can use reverberation as another clue. Sounds passing from left to right (or vice versa) at high speed in a large space, appear to fall in pitch as they pass us (at centre 0), the Doppler effect. (However, a sound moving from left to right without Doppler Effect may be perceived as moving in a very small space which has been magnified, e.g. rolling around a small bowl).

As an example let's consider circular motion.

## Circular motion

Imagine an object on a flat table in front of view, moving in a circle. You are hearing it from the level of the table top, the sound will appear to move left→right→left→ etc. but will appear to move faster in the centre (where it is moving across left-right, and slower at the edges (where it is moving back or forth, without much left-right movement). Ideally, this motion is described by a sine-wave, but the ear's position discrimination is not very accurate. For practical purposes we can create a file which moves at constant speed from right→left (or vice versa).

```
0    -.9
1     .9
2    -.9
....
```

But lingers near the edges of the space e.g.,

```
0    -.9
.3   -.75
.7    .75
1     .9
1.3   .75
1.7   -.75
2     -.9
2.3   -.75
....
```

To enhance the feeling that the sound circles towards us, then away from us, we can add a loudness envelope to the output which is at a maximum when the sound passes the centre the first time, and at a minimum when it passes the 2<sup>nd</sup> time E.g.,

```
.5    1    (near us)
1.5   .3   (far from us)
2.5   1    (near us)
```

Either in addition, or alternatively, we can make a filtered version of the sound, using **Filter → lopass/hipass → bands as frq**. To ensure this is a **low pass** filter (i.e., it passes the low energy, and filters out the high energy), we must set the Passband **lower than** the Stopband. E.g., we might set the Passband to 400 and the Stopband to 600. This will produce a muffled version of our original sound. We can then mix together the original sound and its muffled version, so that (for example) the original is most prominent (and therefore the sound apparently close to us) when the sound passes Left → Right, but the muffled more prominent (and therefore the sound further away) when it passes Right → Left. This implies that the sound is circling.

Place the filtered sound and the original sound (in that order) on the Chosen Files list, and select **Mix → balance between 2 sounds**. Press the Make File button and create a balance file.

If the balance is 1 we hear only the first (filtered) sound, and if it is 0 only the 2<sup>nd</sup> (original) sound. If it is .5, the 2 sounds are mixed in equal proportions. The balance file has a list of time – balance pairs, and to make the same circling effect as above we might write

```
.5    0    (near us, the filtered sound is not heard)
1.5   1   (far from us, only the filtered sound is heard)
```

## 2.5 0 (near us, the filtered sound is **not** heard)

The balance values at other times are interpolated from the existing values, so at in-between times (1.25, 1.75 etc.) the 2 sounds are equally mixed, and at these times the sound is also to the left or the right of the space (that is how we panned them).

# Texture

It is possible to produce sequences or masses of sound from a single sound. This can be done using the **Texture** process. A texture consists of the original sound played back over and over again, where you can specify the timing, transposition, loudness and spatial position of the events.

Start with a single sound, which can be of any duration, but a short mono sound is best to experiment with (stereo sounds are reduced to mono by the process). Choose **Texture → Simple → Neutral**. Neutral means that the pitches used in the texture will not lie in any specific harmonic field. In particular they will not lie on the notes of the tempered scale, but will be have random frequencies between prescribed limits (described below).

The **Texture Set** began many years ago as ‘Remix’, and specifies most of its values as MIDI values.

## Note data

The first parameter of the texture process is a special data file which gives information about the pitch(es) of the input files (and, in other texture programs, information about harmonic fields etc.). The process uses the pitch information to decide how to transpose the input sound in the output texture. If we tell it that the input sound is middle C (midi 60), and tell the process to transpose it to the E above middle C (midi 64) it will do that (transposing it up a major 3<sup>rd</sup>). However, you do not have to use a pitched sound (and you can lie about the pitch!). Thus if you state that the pitch is 60 and ask for it to be transposed to the 64, the sound will be transposed up a major 3<sup>rd</sup> **regardless of what its true pitch is**. So, if in doubt, you should simply say that the pitch of your sound is 60.

For a simple, neutral texture, with one file, the data file needs just one number specifying the “pitch” of that input file. Go to **Make File** and on the text window that appears you can either type in the exact pitch of your source (if you know it) or you can press the **Standard Features** button, which will print a “60” for you, and give a default name “Notedata” to the file. You must now save this file.

You can run texture programs with several input files, and in that case, the data file needs a pitch value for each input file. (The standard features button will generate a string of “60” of the correct length).

## Event packing

This parameter describes the (average) time step between each new play of the source sound. If the time step is small, the plays will overlap one another to make a dense texture. If the step is large, the plays will not overlap (or not so much) and the texture will be more sparse.

Note that this, and almost every other parameter of texture can be **varied through time** by creating a breakpoint file (via the **Make File**) button.

## Scatter

This determines how regular the event-placing in time is. With a scatter of 0 the events in the texture will come at a standard time-step, creating a rigid pulse. With a scatter of 1 the events will be randomised in time, with no feeling of pulse.

Greater than 1 produces even more randomisation. Small values (e.g. 0.05) may produce or pulsed output with a more “human” feel.

## Time grid unit

If you want to quantise the time, enter a value here. This forces any events, even if randomly scattered, to occur only at particular places on a time-grid: e.g., only at some or all of 0.0, 0.2, 0.4, 0.6, 0.8 etc., but never anywhere else. Otherwise leave it as zero.

## First / Last sound in list

If you are using only one sound, both these parameters have the value 1 (there is only one sound in the Chosen Files list). If you are using many sounds, you can select which contiguous group of sounds (from your list of entered sounds) to use. The important feature here is that you can vary these parameters over time, for example gradually increasing the number of sounds you introduce into the texture.

## Min / Max event gain

These define the minimum and maximum possible loudness of each event. Each event’s level is set, at random, between these 2 limits. Values are MIDI (0 – 127) corresponding to a gain range of 0 – 1.

## Min / Max event sustain; Play all of insound

The events can be shortened but cannot be lengthened. Note that, as an event is transposed upwards it will become shorter (and longer if transposed downwards). These parameters can be used to define limits for the duration of the individual events **but you cannot force sounds to be longer than they are**. You can override these parameters, to force each event to play **all** the original sound, by ticking the box at the bottom, labelled **Play All of Insound**.

## Min / Max pitch

These parameters specify (as MIDI pitches) the upper and lower limits of the range of pitches that the events can take. As explained before, these are defined **relative to** the pitches you specified for the input sound(s) in the data file (first parameter). They thus define the outer limits of the **transposition** of the input sound(s). These range-limits can be made to vary over time.

## Overall attenuation

If the texture becomes very dense, it may overload. If it does it will automatically reset this parameter (to turn itself down). If this happens, run the process again (and again, if it still overloads).

## Spatial position

This defines the centre of the (possible) spread of positions of the events in the texture. 0 is left, 1 is right.

## Spatial spread

This defines the spatial spread of the texture events. 1 means fill the stereo space. 0 means all events are at the same position (defined by **spatial position**).

Spatial Position and Spread can vary over time.

The texture program can be used both to make sound-masses, create events which flow from regular pulsation to random scatter, or to develop entire phrases (or even pieces) by creating time-varying parameters.

## Table Editor

The Table Editor is a powerful tool for manipulating tables of data, like control files or *mixfiles*. The table Editor is accessible from many places in the Loom. In particular, if you access it from the parameters pages, you can send its output (a table of values) directly to a parameter box on that Parameter Page.

**There are 5 panels on the Table Editor page.**

### Basic mode of operation

1. Select a file from panel 1 (left) - its contents will be listed in panel 2.
2. Select a column from the data in column 2, by entering a column number in the box below “Get Column”. The column data will be shown in panel 3.
3. Do some operation on the values in the column. The result will appear as a column of values in panel 4.
  - a) The operations are listed under the various buttons above the panels.
  - b) You will often need to enter a parameter (N) in the N-box (top right), and sometimes a “threshold” value in the threshold-box (to right of N-box). With other processes, a special value-entry box will pop-up.
4. In the simplest case, the new column you have created will have the same number of entries as the original column you chose. In this case, just press OK at the foot of the area marked “Keep Output Column?”. This will create a new table, with your new column of values replacing the original column.
5. You can now save the new table. Enter a name (right) and press “All”. You should see your new table appear in the list in Panel 1. You will also find it on the Workspace.
6. If you have called the Table Editor from a parameters page, once you have saved your output file, you can send it to the parameters page with the button **Outfile as Param**.

An example would be to take a *mixfile*, extract the times column from it, modify the times (e.g. make it twice as slow by setting N as 2, and choosing **Maths → Multiply by N**), and then save

the new *mixfile*, which changes the tempo of your *mixfile* without altering the duration of the individual sounds being mixed.

The Table Editor has a great number of other uses e.g. you can enter rhythms by Tapping them: (**Create → Create Times → Tap Out a Rhythm**) or enter MIDI data from a keyboard (use the small **piano-keyboard button**), do algebraic operations on tables, or modify whole sequences of processes (batchfiles). For more details of how the Table Editor can be used, consult the **How To Use** button on the top right.

## Calculator

This is a calculator designed for musical use. It will convert between musical units of various kinds and also do maths calculations (including simple algebra).

The basic mode of operation is to use the buttons on the left to enter values to the **VALUE** box in the centre. Choose what your number represents (e.g. semitones, frequency, milliseconds) on the **INPUT UNITS** panel (centre right), then select a button from the **OUTPUT UNITS** panel (next right) and see the conversion or calculation in the output box (bottom right).

Results can be **recycled** to the **VALUE** box or **stored** on the **STORE** box. Calculations involving samples and time require you to enter a **sampling rate** (panel on right). You can also convert bars-beats-tempo to time (etc.) ... follow the instructions on the highlighted buttons as you attempt to do this.

## Section 7: FILTERS, DISTORTION, GRAIN & PITCH EXTRACTION

### Filters

Filters are devices to enhance or remove the energy at a particular place in the spectrum. If you cover your mouth with a hand while you speak, you will cut out the high frequencies in the spectrum and as a result your speech will be muffled or incomprehensible. If you speak nasally and pinch your nose you will cut out the low frequencies, and your voice will sound “squeaky” or “pinched”. Note that neither process changes the pitch of the sound,

#### Low Pass / High Pass

A low pass filter will pass the low frequencies (remove the high frequencies) like the hand over the mouth. A high pass filter will pass the high frequencies (remove the low) like the pinched nose. In the Sound Loom these are both represented in the same filter at **Filter → lopass/hipass**. If you set the **Stopband** higher than the **Passband** it will act as a lowpass filter. Conversely, setting the Stopband *lower* than the Passband will force it to act as a high pass filter. The attenuation (in decibels) determines by how much the filtered part of the spectrum is reduced in level. With decibels, 0 means no reduction, while -96 means the maximum reduction in level.

When you filter a sound, because you are removing energy, the output sound will have a lower level. For this reason you may want to add some gain to the input. Do this with **Prescale**, but be sure to check that your output has not overloaded (distorted) by using the **Maxsamp** button (top right – only active once you have run the process, and before you save the output file).

The sound from a filter will often resonate a little (or a lot), making the output longer than the input sound. To be sure to catch all of this resonance (and therefore not cut the sound off abruptly) you can extend the **Decay tail duration**.

A **bandpass filter** is a filter that reduces the level of everything that lies outside the band you specify (a **notch filter** reduces the level *inside* the band you specify).

#### Filter Banks and ‘Q’

You can also send your sound through a bank of different bandpass filters, using **Filter → bank**.

The outputs of these filters are added together to create the final sound. The filter output may therefore **overflow** (you will get a message in the Run page window). The filter is self-correcting (if it is about to get too loud it turns itself down) but if you get an overflow message it is wise to run the process again and turn down the **Gain** parameter.

The filter banks have parameters **Low Frq** and **High Frq**. These determine the lowest and highest frequencies of the bands in your bank of filters. How these are interpreted depends on which filter you choose. For example, if we choose the limits 400 and 2200.....

**Filter → bank → harmonics** creates a set of bands on the harmonics of 400, up as far as 2200 Hz, i.e. 400, 800, 1200, 1600, 2000.

**Filter → bank → alternate harmonics** creates a set of bands on the alternate harmonics i.e. 400, 1200, 2000.

**Filter → bank → subharmonics** creates a set of bands on the subharmonics of higher frequency (2200) i.e. it divides the higher frequency by whole numbers only to give 2200, 1100, 733.33, 550, 440

**Filter → bank → harmonics with offset** creates a set of bands by calculating the harmonics of 400, then adding an offset specified with **Frq Offset**. If we set the offset to 55Hz, we would first generate 400, 800, 1200, 1600, 2000 and then offset these by 55Hz, ending up with 455, 855, 1255, 1655, 2055. A large offset makes the filters have an inharmonic relation to one another - you will hear several non-tempered pitches in the output sound (see Section 4 for an explanation of Inharmonic and Harmonic spectra).

**Filter → bank → fixed number of bands** creates a set of evenly spaced bands between the low and high frq values. Here you must specify the **number of filters**. In general these filters will form an inharmonic set. E.g., with 8 bands, we generate frequencies of 400, 657.143, 914.284, 1171.427 etc. If we specify 7 bands, we will end up with filters of value 400, 700, 1000, 1300, 1600, 1900, 2200, which are inharmonically related except for 400 and 1600, whereas if we specify 10, we end up with the frequencies 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200 which are all harmonically related to an absent lower frq of 200Hz.

The first parameter of these filter banks is **Q**. This determines the narrowness of the filter. A higher Q means a narrower filter. With a very narrow filter, all the energy is concentrated around the frequencies you specify, and the output can sound like a chord from a synthesizer. With a Q around 5, the filtering effect will be subtle or inaudible (values below this are not recommended). You can make the Q vary over time using the “Make File” button to creating a control file in the usual way.

As you increase the Q value, the amount of energy transmitted through the filter decreases so the output sound gets quieter. This can be a problem if you create a control file which increases the Q over time. However you can correct for by replacing the envelope of the output (filtered) sound with the envelope of the input sound. Place the output (filtered) sound on the Chosen Files list, followed by the input (unfiltered) sound, and on the process page select **Envelope → Replace → env from other sndfile**.

## Filters of varying harmony

You can also build a bank of pitched filters where the pitches change over time. You can specify the pitches as MIDI values (which don't have to be whole numbers, so you don't have to use the tempered scale) or as frequencies. On the process page select **Filter → varibank**. The filter has a **Q** value (which you can vary through time with “Make File”), and a **Gain** (if you have a large Q you might want to increase the Gain so that the output has a decent level – close to, but less than, 1). You can also send the sound through the filter twice, for a stronger filtering effect (select **Double Filtering** below).

The main parameter for this process is **Timelist of pitch & amps of filts**. This is a file of data describing your bands which you make by pressing **Make File** to go to a text writing window. At the top of this window, the button **Standard Features** will display the format you need for this data. You need to specify 2 or more lines of data. Each line starts with a time and is followed by a list of pitches and their (relative) amplitudes. These pitch/amplitude pairs will create bandpass filters at the specified pitches, at the specified time. Between each specified time, the filter bands will change in pitch (and amplitude) gradually between the values you have specified. (You must put the same number of values in each line .... to get rid of a band, just set its amplitude to zero).

When you have finished writing the data, hit the “Standard Features” button again (it has changed its name to **Delete Features**) and the top (instructions) line will disappear.

Back on the parameters page, you can also specify the **Number of Harmonics** (of the pitches you've specified) you would like to use. If you set this to 4, for example, a filter band will be assigned to each of the first 4 harmonics of each of your pitches. Hitting the **Max Harmonic** button (above centre) will enter the maximum number of harmonics available (limited by the highest pitch you have specified) into the “Number of Harmonics” box, provided that this is within the range specified for the ‘Number of Harmonics’ parameter. Particularly, with low-pitched sounds, if the value in the box is not being set, try extending the range by pressing the ‘Range’ button – on the left of the parameter bar – to allow a greater range of harmonics to be entered, and press the ‘Max Harmonic’ button again.

The **Roll off** specifies how loud the harmonics are relative to the base pitch. A roll-off of minus 3 (decibels) means that each harmonic is 3dB quieter than the previous (lower) one as you go up the spectrum.

## Filtering by spectral subtraction

An elegant way to remove (consistent) noise from a sound is to subtract the spectrum of the noise from the signal. To do this, first cut out a segment of the source (no matter how small) which contains **only** the background noise. Then analyse the source and the noise fragment with **PVOC → analysis**. Then put the 2 analysis files (the source first) on the Chosen Files list, and select **Simple → Clean by subtraction**. (The default value for **persistence** is probably optimal). For more detailed work of this kind, see the ‘Cleaning Kit’ on the **Music Testbed**.

## Waveset Distortion

The **waveset distortion** process divides your sound into pieces by counting the number of times the wave crosses the zero line (you can see this happening in the Sound View window if you zoom in down to the sample level). A simple sine wave crosses the zero just twice in each wavecycle, but more complex waves cross the zero several times during a wavecycle. If you can isolate the wavecycle and then repeat each wavecycle, you would be able to time-stretch the sound without changing its spectrum. However, these distortion programs simply count 2 zero-crossings, and use that part of the signal (the “waveset”) as a unit. This may or may not correspond to the real wavecycle. As a result, when we manipulate the “waveset” we produce somewhat unpredictable “artefacts” – the sound, or parts of it, may be transposed down or up, and noise, or grittiness may be introduced.

Repeating the wavecycles 16 or more times will produce a stream of brief pitched sounds, as each repeated waveset is like a sustained synthetic sound of a particular colour (produced by the particular shape of the waveset).

We can also choose a group of wavesets (say 3) and repeat the whole group.

There are numerous processes that manipulate wavesets in the **Distort** menu.

In particular, we can create repeating loudness envelopes on the sound whose duration is determined by counting the wavesets. In a complex sound, the duration of (say) 16 wavesets can vary enormously with time e.g. in speech, the sibilants have a noise spectrum which crosses the zero very many times in rapid succession, so the envelope during sibilants will be very brief.

Inside vowels however the envelope duration will be much longer as the vowel signal crosses the zero much less frequently. The audible effect of the envelopes will therefore vary as the nature of the sound itself changes.

## Grain

Any sound which has a definite graininess (we can hear individual attacks separated by low level signal or silence) can be manipulated by processes on the **Grain** menu. These programs begin by trying to isolate the grains in your sound (if there are no grains, they will not work). To do this they must follow the loudness envelope of your sound, and must judge how big to make the envelope window (see Section 5) to correctly isolate the grains. The process **grain → assess max no of grains** will search for the best setting for this window size. If you run this process then proceed **directly** to another grain program, you will find the appropriate *windowsize* preset in the parameter box for the new process.

You can now duplicate, omit, transpose, or shift in time the individual grains in your source sound. You can also **timewarp** the sound **without** timewarping the individual grains (like inserting rests between notes, or overlapping them, in a score).

## Pitch and Formant Extraction

Providing your sound has a definite (possibly moving) pitch, you can attempt to extract it. Analyse it first with PVOC, then use the analysis file and go to **Repitch → extract pitch from analysis data → to binary file**. The output of this process is two files. The first is an analysis file which you can resynthesize (with PVOC) to hear if the pitch-extraction is successful. The 2<sup>nd</sup> is a pitch data file (with filename extension “.frq”).

Similarly, if your file has a changing spectrum (like speech), you can extract this from the analysis file using **Formants → extract**.

You can then manipulate the pitch data (see the **Repitch** menu) and combine the new pitch with the original formants.

These processes work best where the data has very clear pitch or “formant” properties (formants are peaks in the spectrum, and are often used to describe vowels).

## Section 8: ASSEMBLING SOUND MATERIALS

### Mixing

#### Purpose

The **Mix** programs in the CDP/Sound Loom can be used ...

1. To create new sounds, by combining existing sounds.
2. To create small gestures or larger phrases, combining existing sounds.
3. To assemble phrases into sections of a piece, and sections into a whole piece.

Remember that, at every stage, the output is simply another sound file, which can become the source-material for the next stage of your composing.

#### Sound Mixing

Mixing sounds on the Sound Loom takes place in 2 stages.

1. The sounds are used to create a ***Mixfile***. A *mixfile* is a textfile which describes how the sounds are to be mixed together. Simple mixfiles can be generated automatically by the CDP/Sound Loom.
2. The *mixfile* is “run” to generate the output sound.

#### Creating the ***Mixfile***

1. Choose the sounds which you wish to mix together, placing them on the “Chosen Files” lists.
2. On the process page select, **Mix**.
3. There are 4 (initial options)
  - a) **Mix → create a mixfile → superimposed.** This creates a mixfile where all your sounds start at time zero.
  - b) **Mix → create a mixfile → end to end.** This creates a mixfile where your sounds succeed one another in the order you placed them on the Chosen Files list. The result is the same as using “edit → join” except that you can now manipulate the mixfile itself (see below).
  - c) **Mix → create mixfile with timestep.** This creates a mixfile in which each file enters N seconds later than the previous file, where N is the timestep (in seconds) that you define.
  - d) **Mix → create mixfile on timegrid.** This creates a mixfile in which the files enter at the times you specify in your “timegrid” file.

Each of these processes produces a textfile which describes the mixing process. You should save this textfile, and then “run” it (see below).

## What is inside a *mixfile* ?

Inside the *mixfile* a **typical line for a mono file** will look like this.

```
mymonosound.wav    0.0000    1    1.00      C
```

The first entry is the name of the soundfile to be mixed.

The second entry is the time in the mix when it starts.

The third entry is the number of channels (1 for a mono file)

The fourth entry (which should lie between 0 and 1) is the level (loudness) of the sound in the mix. If you want the level of your sound to vary over time, you should run it through **loudness → gain** with a breakpoint envelope, to create the loudness shape you want. You can then use the output of that process as the file inside the mix.

The fifth entry is the spatial position of the sound in the output mix. Here “L” means left, “R” right and “C” centre. These values can be specified more precisely by using numbers. In this case -1 is left, 1 is right, and 0 is the centre, but any number in the range -1 to 1 can be used to specify spatial position more precisely. (If you want your sound to move about, you should first use **space → pan** to create a sound moving in stereo, then use the stereo file in the mix).

### For example:

<b>sound</b>	<b>time</b>	<b>chans</b>	<b>level</b>	<b>position</b>
mymonosound.wav	0.00	1	1.00	C

For a stereo file, a typical line will look like this

```
mystereosound.wav    0.00  2    1.00
```

OR

```
mystereosound.wav    0.00  2    1.00 R    1.00 L
```

The first case is similar to the mono sound except

- The entry for “Channels” is now 2, for a stereo file.
- There is no value for position, as we cannot “position” a stereo file (it already takes up the whole stereo spread).

In the second case, in the last 4 entries we have specified the **level and position for each input channel** of the stereo file. The mixing program reads the first of these number-pairs as the data for the left input channel, and the 2<sup>nd</sup> pair as the data for the right input channel. In this example we have told the left channel to go to the right (R) and the right channel to go to the left (L) thereby creating a reversal of the stereo image of the input sound.

### For example:

<b>sound</b>	<b>time</b>	<b>chans</b>	<b>level</b>
mystereosound.wav	0.00	2	1.00

OR

<b>sound</b>	<b>time</b>	<b>chans</b>	<b>level-ch1</b>	<b>pos-ch1</b>	<b>level-ch2</b>	<b>pos-ch2</b>
mystereosound.wav	0.00	2	1.00	R	1.00	L

Once you have saved the *mixfile* output, you can edit it. (The textfile editor understands the syntax of mixfiles, and will tell you if you make a mistake!). However, the simplest way to do this is to edit it as you “run” it.

## Running the Mixfile

We now put the mixfile we have created on the “Chosen Files” list (or just “Recycle” the output of the mix-creation process) and on the Process Page choose **Mix → MIX FROM MIXFILE**. On the parameters page, hit **Run**.

### Levels

Typically, the sounds in your mix will overlay with one another to some extent (but not necessarily): e.g., in case 3a (above) they will always overlay one another. The mix process adds together the sound signals where they overlap, so you have to be careful about **the output level**. Once you have run the mix (and returned to the parameters page), select **MaxSamp** (top centre-right). In the window that appears you can see if the output has overloaded. If it has, the necessary **Attenuation** value will be placed automatically in the Attenuation parameter box on the parameters page. You can then re-run the mix and be sure, on this occasion, that it has not overloaded.

### Testing part of the mix

You can test parts of the mix by setting the **Mixing Starttime** and **Mixing Endtime** values. Note, however, that the output of these part-mixes may have clicks on the start or end ... the outputs are only meant for testing (auditioning) purposes.

### Modifying the mix

Most importantly, you can now modify the mix, from the parameters page, in many ways.

1. Using the **Edit Mix** button you can edit the *mixfile* in a text window. The text editor knows the syntax for mixfiles and will tell you about any errors you make.
2. The **QikEdit** button gives access to a whole range of mix modifying processes, via an array of buttons.
  - a) You can **modify the timing** of individual lines, or groups of lines by first highlighting them with the mouse, then selecting one of the time-modifying buttons. For some of these processes you need to enter a value in the box at the top right. For example to move the last 3 sounds in a mix to be 2 seconds later, enter two in the box, highlight the 3 lines to move, then press **Move Time --> By**.
  - b) You can **modify the loudness**, or the **spatial position** of individual lines or groups of lines in a similar fashion.
  - c) You can **reverse the order of times** (e.g. turn an accelerating rate of entry into a decelerating rate of entry) or **reverse the order of sounds** (e.g. retaining the accelerating rate of entry but with the files in the opposite order), **randomise** the file order (“Randomise Order”), slightly or hugely **randomise the entry-timings** of the sounds (“Shake Times By”), and perform numerous other global operations on the mixfile.

In general, “Edit Mix” is not destructive (i.e. does not destroy your original mix) **if** you give your edited mixfile a new name. In contrast, “QikEdit” overwrites your original mix. However, when you change a mixfile in QikEdit, the original values are **commented out** (with a semi-colon at the start of the line). This means that you can return to your original idea if you make an error (**Undo**) or even after you have used the new mix and it proves not to be better than the old (**Get Previous Mix**).

## Moving between Mixing and Sound Modification Operations

Typically you might make a mix and then decide one or other of the sounds in the mix needs to be modified.

### Extracting a sound from the mix

1. On the “QikEdit” page you can select a line in the mix, then hit **Sound to Wkspace**, which will place the sound in the selected line on the “Chosen Files” list on the Workspace, where you can then modify it (generating a new soundfile).
2. Alternatively, on the Workspace, with the *mixfile* on the “Chosen Files” list, you can use **Get/Mod → Get Soundfiles from Mixfile**, which will replace the *mixfile* with a list of the soundfiles it contains. You can then choose one of these to modify.

### Replacing a sound in a mix

You can replace one sound with another (e.g. a modified version of that sound) in a mixfile by placing the original file, the new file and the *mixfile* on the “Chosen Files” list, and selecting **Get/Mod → Replace One Sndfile With Other in Mixfile**.

### Adding sounds to an existing mix

Place on the “Chosen Files” list the mixfile and the soundfiles you wish to add.

Select **Get/Mod → Add Sndfiles to Mixfile**.

## Constructing an Entire Piece

You can use mixing to gradually build up the structure of your piece. Small events can be arranged in some kind of time-sequence using a *mixfile*. The output “phrases” can then be combined to make sections of the piece, using another *mixfile*, and the sections combined to make the entire piece with a further *mixfile*. At each stage phrases, or sections, can be reshaped (e.g. in amplitude envelope) appropriately to the context in which they are to be placed.

More importantly, if you find a particular phrase or section seems to have the wrong proportions when you start to put your piece together, you can go back to a lower level and redo the mix at the level. More radically, using the “Recall Action” button, you can reconstruct individual sounds, provided you still have the source sounds from which they were made. When sounds are moved to different directories (e.g. when you back to named directories) or even renamed, mixfiles keep track of where the sound has been moved, so you do not have to re-edit the *mixfile*. “Recall Action” attempts to do the same, but in the course of a big piece, where you may have renamed sources several times, you may need a little more help to find your original sources (it will offer prompts to which you can respond).

## Section 9: SOME FINAL REMARKS

At this stage you are probably becoming more familiar with the Sound Loom / CDP, and have perhaps generated many sounds. You will realise that it is easy to fill the Workspace with sound experiments, and the number of sounds can quickly become unmanageable. Also, you may have made some really good sounds but cannot quite remember how you did it. Here are some final comments that may help.

### Finding things on the Workspace

You should try to keep as few files as possible listed on the Workspace. But it is easy to generate a multitude of sounds and intermediate files.

1. **You can search** for files with the **Search** button (top right of Workspace panel), entering some part of the name in the box which appears.
2. **You can sort** the Workspace in all sorts of ways (by name, by type –e.g. all soundfiles to the top, by the date-time order in which they were made, and so on). **Any/All Files → Sort Workspace**. You can also sort specific groups of files: highlight them and choose **Selected Files Only → Sort**.
3. **You can highlight a block of files**. Choose the top (or bottom) file with the mouse in the normal way, then click on the bottom (or top) file while holding down the **shift key**. This will select the whole block of files.
4. **You can find all the Workspace-listed files in a given directory**. Highlight just one file in that directory then choose **Any/All Files → (Select Specific Files) Other Specific Files → Files in Directory of Selected File**.

### Keeping the Workspace Manageable

The files you generate all end up on the Workspace. Once you have a large number of files, the small counter boxes (top left and right of the Workspace) will change colour to alert you. In order to keep on top of all the files you will generate it is important to keep on the Workspace only those sounds you are either working with (as sources, or intermediate files) or are in the process of making.

### Removing files which are already backed up

The Workspace will probably list files which already have pathnames. These may be files you have already placed in directories, and possibly source sounds from other directories.

You can remove any of these from the Workspace (you can always recover them later by typing the directory name into the directory box and hitting **List** above the directory panel (right) then grabbing them onto the Workspace).

To do this, highlight those files on the Workspace then go to **Selected Files Only → (Clear Files from Workspace) Files which are already backed up**. This will remove the highlighted files from the Workspace listing.

## Backing up newly generated files

In order to know where you have been and where you might be going, you should sort new sounds into categories (e.g first experiments, really good sounds, sounds you may well develop further, sounds for a particular section of the piece, sounds you may keep for another piece, and so on) and store each category in a named directory. It is worth spending a few minutes at the end of every session to do this (even better to do it as you go along).

## Creating a new directory

With new files (those that have **no** pathname in the Workspace listing) you can do this by writing the name of a directory in the directory-name box (top right), then selecting the files you want to put in it, then pressing **Backup Selected New Files** (middle above Workspace panel). On the small menu that appears, if you select **Store** the files will be placed in the directory, but will remain on the Workspace (with the directory name added in front of each of them). However, if you want to also clear them from the Workspace, so they are no longer listed there, select **Store and Clear**. You can always retrieve the files at a later time.

## Backing up files to an existing directory

Do this in exactly the same way, or, if you have used the directory recently, you will find its name by going to **Find Directory → Recent Directories** where you can select it from a list by clicking on its name.

## Moving Files to different directories

Your choice of which directory you keep a file in is not irrevocable. You can always move files to a different directory (this tends to happen increasingly as a piece progresses, as there are more and more subcategories of material to keep a hold on). Put the name of the new directory in the directory-name box, select the file(s), then go to **Selected Files Only → Move to New Directory**.

## What to do with “intermediate” files

Usually you will have many intermediate sounds (e.g. analysis files made on the way to creating some transformation of a source) that may not be wanted for the piece, but some may be useful to use in your ongoing work. But **if in any doubt, err on the side of caution** and keep the files in a “possibly-useful” directory.

Things you definitely don’t want Can be **destroyed** (this is irrevocable!). Highlight the items and hit **Control-d**. (You will get an “Are You Sure ?” message, so it’s never too late to change your mind!).

## Remembering Your Best Ideas

When you began working it probably did not seem worth keeping a note of what you were doing, but as you progress the need to remember where you’ve been and what you’ve done will become clearer. If you make a particularly striking sound, it is worth make a note in the **Notebook** about how you did it. If you forget to do this, you can use **Recall Action** to find out exactly what you did. In particular, you can

1. Highlight on the Workspace the file you want information about.
2. Go to **Recall Action** (top left) and in the window which appears select **Search All Logs** (top left).
3. In the next window (“**Search Relevant Logs**”) hit the button **Preselected Text** and choose **Workspace File** from the menu. This will put the name of your Workspace-selected into the name box on the left. (Alternatively you can just type it into this box).
4. Now press **SEARCH**. The system will search for any logs which contain the filename (i.e. any actions in which the file was generated or used).
5. You should then get a list of logs in the top panel (if you have lots of logs you may have to wait a while).
6. If you now choose a log (highlight it in the panel) the relevant lines from the log will appear in the lower window. From these you should be able to reconstruct how you made or used the sound.

You can even recall the act of creation to the Workspace. Highlight the relevant log in the top panel, then press **Get Selected Log** – once the log appears, select the line you want and press **Recall Action**. If some of the intermediate files have been destroyed, or renamed, you may have problems (though there are ways to recover actions in many circumstances), but if you have been working only for a short time, the **Recall Action** option may well work immediately, taking you immediately to the parameters page ready to remake the sound (or at least see how you made it).

For even more detailed management and sorting of your sounds or sources, you can use ‘Property Files’ (on the **Music Testbed**) to categorise your sounds in terms of the properties you feel are important for your composing. The properties are defined by you and might be commonly used musical features such as “pitch-motion”, or purely subjective properties such as “ugly”, “zippy” etc.). Property values can be commonly used values (such as “C#” for pitch), numerical values (e.g. “0.7”) or names of your own choosing (e.g. “very”).

Managing your material and work is a very important part of constructing a piece in this medium, simply to avoid becoming overwhelmed by sounds and information!!

Good composing!