Introducing the CDP Sound Transformation Programs

Release 7 (MAC / PC)

By Archer Endrich

Music is a vast cornucopia of wonders. There are many approaches and ways of talking about it. This document aims to set forth our approach and terminology to help orient you the user. We may refer to familiar concepts, but with unfamiliar terminology, or you may already be close to this way of thinking and working. We hope that this Introduction will help you to make a connection between the CDP software and your current way of working.

CONTENTS

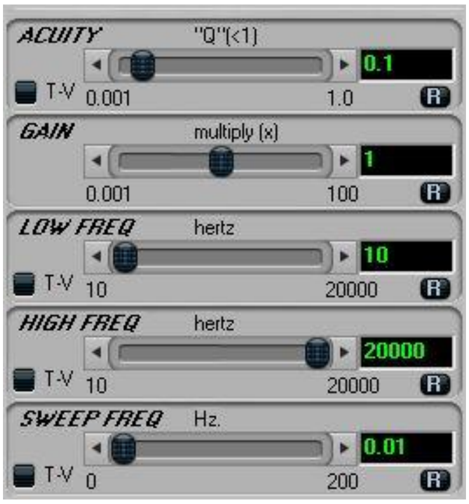
SECTION 1 – Introduction

The CDP System is designed for the in-depth exploration of the possibilities for sound transformation. You can use the CDP software via a graphic user interface (GUI) or via a Command line interface (CLI), depending on preference and the particular task in hand. The GUIs available are *Sound* Loom on PC & MAC (free), and on PC only *SoundshaperLITE* (free) or *Soundshaper Pro* (a purchase). Settings used can be retained as **Presets** (*Soundshaper*) or **Instruments** (*Sound Loom*) or as **batch files** (command line) and re-loaded at another time. The GUIs have mechanisms to keep a text file record of what you do in each working session. Thus the emphasis is on creativity, innovation, awareness of detail, careful form-building and accurate record keeping.

The CDP software is all offline processing of soundfiles, not 'realtime', generally from pre-existing 'sampled' sound. It therefore belongs to the *musique concrète* type of electroacoustic music. Much very complex processing is therefore accomplished in just a few seconds. The emphasis is on the design of sounds, either for fully electroacoustic music compositions or as sound bites or tracks in songs or other media. Many use CDP alongside other software in a hybrid studio environment.

Robert Fraser has written an excellent survey of the various CDP processing functions. See this file on his Website: **http://www.ensemble-software.net/cdp.html**.

Soundshaper Parameter Dialogue:



ACUITY "Q"(<1) 0.1 (0.001 – 1.0)
GAIN multiply (x) 1 (0.001 – 100)
LOW FREQ hertz 10 (10 – 20000)
HIGH FREQ hertz 20000 (10 – 20000)
SWEEP FREQ Hz. 0.01 (0 – 200)

PARAMETERS

The CDP software provides access to a multitude of detailed sound transformation parameters, parameters which can be set individually and their values known. These parameters can be set with a single **fixed** value, or with **variable** values, either selected randomly within a specified range, or changing over time: time-varying 'breakpoint' data (also known as 'automation'). CDP's **breakpoint** files are text files in which values can be precisely defined for *time* and one or two other types of value. They can be created in a text editor or by using facilities in the GUIs.

The ability to think about music in terms of 'parameters' is an important skill. In order to work with music on the computer, various features and components of sound have had to be identified and given a quantified range of values. For example, while musicians think of a sound in terms of its *loudness* or *intensity*, in digital audio the primary term is *amplitude*. This refers both to the raw shape of a waveform (the height of the peaks and troughs) and also, more importantly, to the range of numbers with which that waveform is represented in the computer. There are many ways of doing this, but in all cases the two key numbers are those which represent silence and the peak (maximum) amplitude. Together these define the *amplitude range* of signal. To express amplitude, the CDP software uses either the **0 to 1** numerical range or standard Decibels. The value **1.0** defines the peak level, and the value **0.0** signifies silence. Expressed in decibels, this range is **0dB** (full amplitude) or **-96dB** (silence). With 24bit or 32bit audio, dB values can be even lower than -96dB.

Our Gain-dB Chart (*gadbchrt.htm*) shown on page 4 gives an overview of the numerical relationships between gain and dB, and may prove helpful. The file is in **\docs\html**.

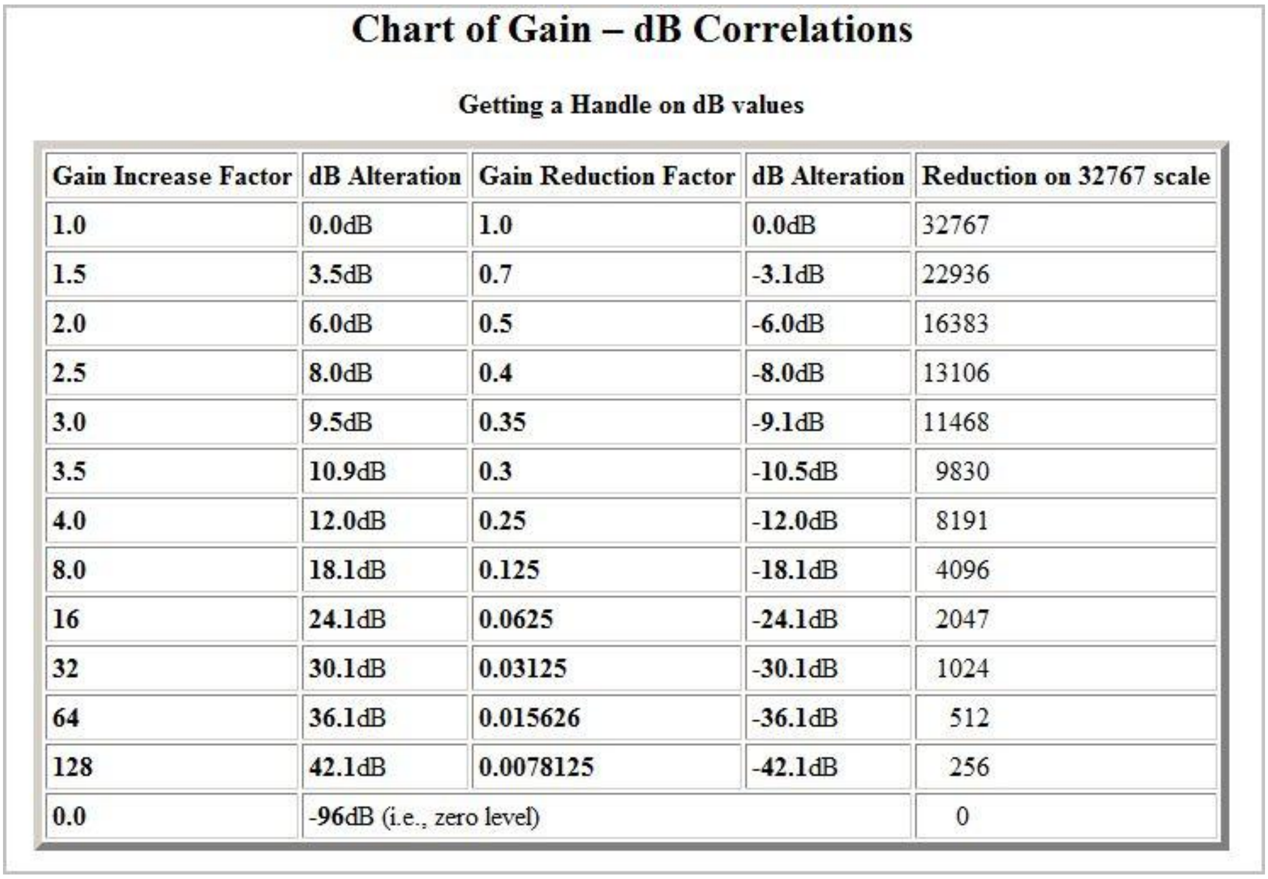


Chart of Gain – dB Correlations

Getting a Handle on dB values

Gain Increase Factor	dB Alteration	Gain Reduction Factor	dB Alteration	Reduction on 32767 scale
1.0	0.0dB	1.0	0.0dB	32767
1.5	3.5dB	0.7	-3.1dB	22936
2.0	6.0dB	0.5	-6.0dB	16383
2.5	8.0dB	0.4	-8.0dB	13106
3.0	9.5dB	0.35	-9.1dB	11468
3.5	10.9dB	0.3	-10.5dB	9830
4.0	12.0dB	0.25	-12.0dB	8191
8.0	18.1dB	0.125	-18.1dB	4096
16	24.1dB	0.0625	-24.1dB	2047
32	30.1dB	0.03125	-30.1dB	1024
64	36.1dB	0.015626	-36.1dB	512
128	42.1dB	0.0078125	-42.1dB	256
0.0	-96dB (i.e., zero level)			0

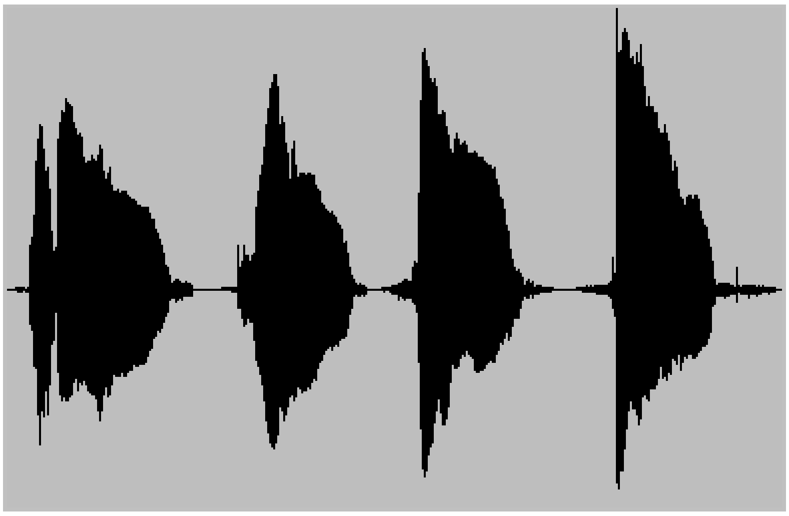
KEY FEATURES OF A SOUND

After a while, it starts to feel natural to think in terms of parameters. They specify features of a sound that we want to alter. The following information summarises the key features of sound and the core terminology associated with them.

- '**Parameters**': *para* + *metron*: beside/subsidiary + measure. They model the components of sound and form the basis of parametric thinking, a new way of dealing with musical ingredients. It can be as simple as UP & DOWN, NEAR & FAR, HOW MUCH & HOW LITTLE along scales of values. Working on music with computers is actually a very physical, experiential activity – and it is the accumulation of experience about what happens that gradually brings it all to life.
- '**Parametric thinking**' means discriminating and understanding the physical measures of the features of a sound as it changes over time. Initially this is done in numbers, but as one becomes familiar with sonic features and their effect on us, the thinking becomes more directly poetic and dramatic: they are understood as *gesture* and as *psychological impact*.

A *time domain* display of the sound shows **samples**: time units and amplitude levels at that time point. The graphic display shows *time* on the horizontal axis and *amplitude* on the vertical axis, positive values above the zero line (speaker cone moving forwards) and negative below (speaker cone moving towards the back). This type of information and display is used for processing **soundfiles**.

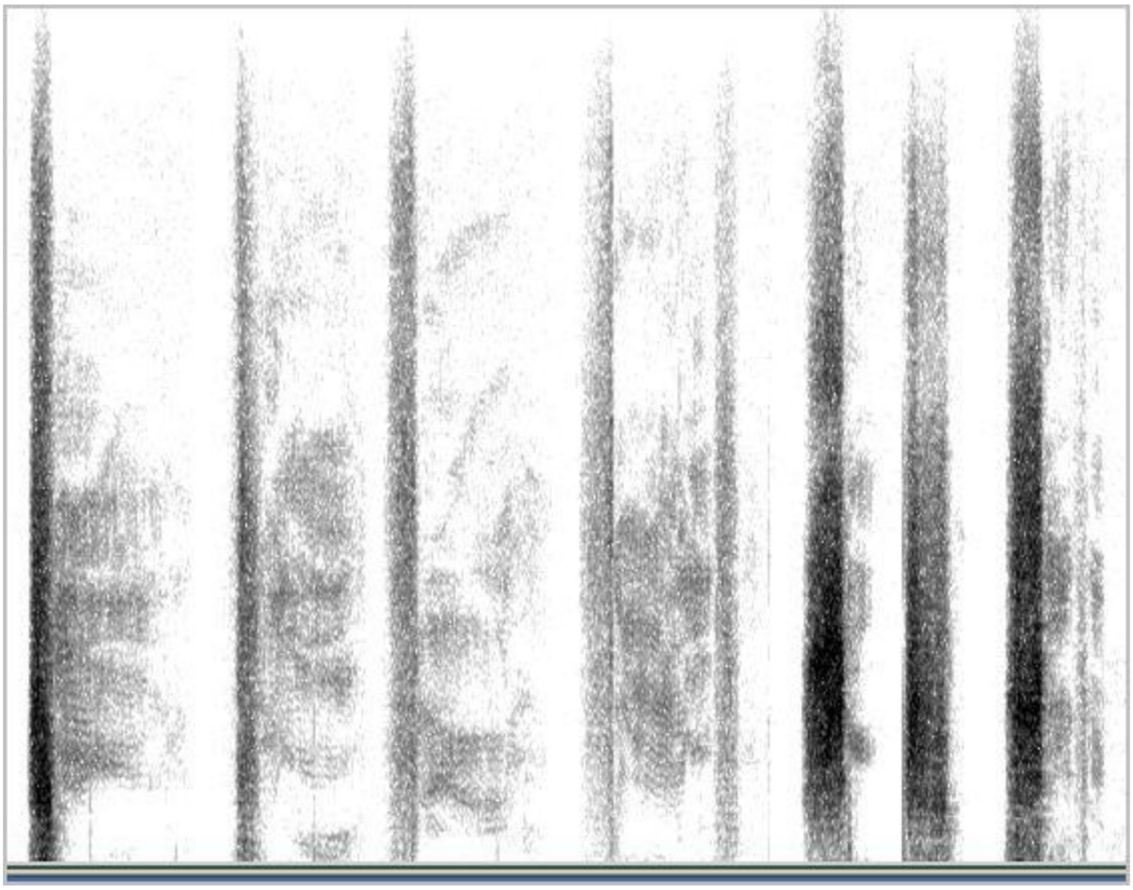
The image below shows a time domain signal display.



Time (horizontal) + amplitude of samples (vertical)

A *frequency domain* (FFT) display shows **partials** (frequencies) and their **amplitude** levels (processing analysis files)., e.g. in a 3-D display or with colours. The *sonogram display* below, from *Sound Loom*, shows **time** along the horizontal axis and the **frequencies** along the vertical axis, with amplitude of the bins shown in a greyscale. This type of information and display is used for processing **analysis files**. Another term for the frequency domain is the *spectral domain*. Analysis files are *converted* from and back to soundfiles with PVOC Analyse and PVOC Synth.

The image below shows a sonogram, a spectral domain display in *Sound Loom*.

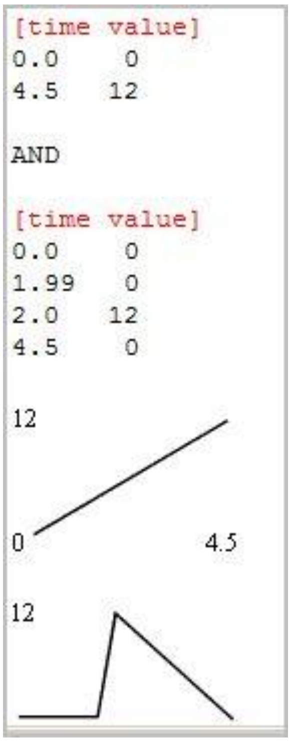


Time (horizontal) + frequencies (vertical)

You will find CDP functions for both the *time* and the *spectral* domains. Sometimes they are referred to with these terms and sometimes as **soundfile** or **analysis file** functions.

The spacing of partials in a sonogram depends on whether a linear or a log vertical scale is used (*Audacity* offers both). Partials are **harmonic** if they are integer multiples of the fundamental. The partials are bound together as it were into what is perceived as a single pitch. Evenly spaced harmonic partials will only be seen in the log display, corresponding to musical notes. **Inharmonic** partials are fractional (or decimal) multiples of the fundamental and can be perceived more individually, like the different tones in a bell or a gong. The makeup of the partials in a sound and how they come and go over time (the **spectral envelope**) determines the **timbre** of a sound.

Amplitude changes constantly over time, forming the amplitude **envelope**. This term 'envelope' refers not to the value of each digital sample, but to the overall amplitude shape of a signal as it changes over time. This can be represented as a simple *time value* **breakpoint file** – a succession of straight lines over time: a time and a value for every sample. To control the action of parameters over time, far fewer time-points are needed. Thus the term **control rate,** and it is for this that most CDP breakpoint files are used. This will be familiar to many users of DAW editing tools as an 'automation signal'.



```
[time value]
0.0     0
4.5    12

AND

[time value]
0.0     0
1.99    0
2.0    12
4.5     0
```

Two breakpoint files are shown on the left. They control the movement of pitch in semitones. The CDP software linearly interpolates intermediate values between the user-specified values at different times, creating a ramp between the two values. These examples involve pitch, so the ramp is heard as a glissando, except in the second file between time 1.99 and 2.00, so close in time that the pitch change is virtually simultaneous, i.e., heard as an intervallic leap.

Much of the power of the CDP system comes from the use of such breakpoint files to control some or all parameters of a process, to create a sound that changes over time in a wide variety of ways.

KEY TERMINOLOGY

Sample rate: number of **samples** per second with which the physical sound is represented. CD quality is 44100 samples per second. For the highest quality "pro-audio" production, higher sample rates are in common use, such as 96000. In the CDP software, output soundfiles are given the same sample rate as the input soundfile, whatever it is. There is also software to change the sample rate of a soundfile. Rates less than 44100 are not recommended. **Digital noise** occurs when there are significant differences of amplitude from one sample to the next. Higher sample rates minimise this by sampling the signal more times per second.

Amplitude: our basic measure of loudness. **Gain** is the degree of change to amplitude, and can similarly be expressed as a linear amplitude change or as a dB change. A **Gain Factor** of 0.5 (i.e., where every sample value is multiplied by 0.5) corresponds to a change of -6dB. Excessive use of *gain* can result in digital noise, and distortion can occur when the signal level rises above the maximum, e.g., > 1.0. This can happen very easily when mixing soundfiles (the amplitudes are summed), so the *mixfile* allows you to specify the amplitude of each component soundfile.

Zero crossings are when the waveform crosses the zero line. The **Attack Transient** is the rate and amount of amplitude increase at the start of the sound – both a natural physical characteristic of many sounds and an important musical *gesture*. Often it is only from the attack that we recognise what is making the sound. It also affects what we perceive as the **timbre** or tone colour of the sound. **Decay** is how the sound ends, the rate at which the sound returns to silence. Another use of zero crossings occurs in CDP's DISTORT Set of programs, based on the irregular patterns formed when a complex waveform crosses the zero point.

Frequency/pitch/partials: oscillation components of the sound. **Frequency** is a general term for (wave)cycles per second, measured in *Hertz*. The standard orchestral tuning standard is defined as Concert A = 440Hz. "Middle C" on the piano has a frequency of approximately 261.6 Hz. Like amplitude, frequency is measured on a linear scale, while musical '**Pitch**' is defined logarithmically in terms of *intervals* – such as the semitone or the octave. Thus the two most important properties of a sound can be represented either linearly (amplitude, frequency) or logarithmically (dB, pitch).

Partials is a general term used for all frequency components making up a sound, whether **Harmonic** or **Inharmonic**. **Noise** is an irregular complex of partials, and **Distortion** produces added *noise* components in various ways.

Timbre is the perceived 'tone colour' the (changing) partials produce, the sum total of all the partials, known as the **Spectrum** of the sound. **Density** is the closeness of partials (or note events).

SECTION 2 – Your working environment will/may include:

- **Sounds** in **.wav** or **.aiff** format. In the process of composition you may work with mono inputs much of the time. Some functions which produce stereo outputs will *only* accept mono inputs, e.g., PAN. In other cases soundfiles with 2 or more channels are now supported. *Soundshaper* is more forgiving in this respect because it often handles these matters internally. Check the documentation for PAPLAY or PVPLAY (in SYSUTILS) for more details about soundfile support. **NB:** When calling a play program on the command line, you now must specify the soundfile extension. *If you have problems with sounds made with other software, they can sometimes be resolved by making a copy of the sound with CDP's* COPYSFX *program.* This will re-write the sound with standard **wav** or **aiff** header information, which will be recognised by the CDP software. The problem can arise when other software does not write a header that is fully compliant with the standard.

- **Executable programs** are in e.g. **C:/cdpr7/_cdp/_cdprogs,** whether using a MAC, via *Sound Loom*) or a PC with *Soundshaper* or *Sound Loom*. If on PC, you are also recommended to install Rajmil Fishman's AL-ERWIN. AL is an algorithmic compositional environment that *can be seen as a hybrid mixing program which combines track-based and sound-design type mixing*. "It is designed for the creation and manipulation of sonic events and their organisation into a structured musical work." ERWIN is a plug-in for advanced granular synthesis: it "applies the solutions of Schrödinger's equation for an atomic potential with radial symmetry (a well-known equation from Quantum Mechanics) to the creation of granular clouds".

- **Graphic User Interface(s)**: *Soundshaper* (PC only) &/or *Sound Loom* (PC and MAC). ***SoundshaperLITE* comes with Release 7, but it only supports Release 5 and parts of Release 6.** *Soundshaper Pro* supports the full set of Release 7 programs. On the PC there are also some other graphic programs: *GrainMill* (granular manipulation: graphic version of the BRASSAGE program), *Brkedit* (breakpoint editor with additional facilities for data reduction and for comparing two breakpoint files; it can be accessed directly or from within *GrainMill*), and *Viewsf* (a very simple soundfile display program, but it does zoom down to the single sample level and display individual sample values).

- You are recommended to consider adding additional third-party graphic editing tools, such as: AUDACITY (http://www.audacity.sourceforge.net) and/or WAVOSAUR (http://www.wavosaur.com/download.php – NB: Windows-only) to your overall system. AUDACITY is also useful for wav to mp3 conversions (for licensing reasons you will need to install the mp3 module separately – see the Audacity website for details). Both are free, but donations are requested.

- **A plain text editor** — you will need this to create or edit plain text files (i.e., no formatting information), such as **note data files** for TEXTURE and quite a few other types of textfile. A frequent need is to create and edit **breakpoint files** (as described above, e.g. *time value* data) to define time-varying changes (**.brk** or **.txt**), and another key file type is the **mixfile,** which can be for mono or stereo files (**.mix**) or for multi-channel files **.mmx).** Typical plain text editors on PC are Notepad++, Wordpad, Emacs, Textpad and Emacs. The standard text editor on the MAC is **TextEdit,** but several others are available**,** including the classic UNIX version of Emacs and TextWrangler. The Emacs text editor has powerful macro facilities and the extension can be anything. You will find that Wordpad sometimes adds '.txt' when you SAVE AS with a '.brk' extension, so you have to rename it to get rid of the '.txt' A quick guide to Emacs (*useemacs.htm*) is included in the CDP documentation. (See **http://ftp.gnu.org/gnu/emacs/windows** for the current version.)

- **A guide to the different types of text file format is provided**. It is referred to as *Files & Formats* in the HTML documentation. You can also access it directly by creating a desktop shortcut to *filesfrm.htm* in \docs\html – this is recommended.

- **Breakpoint files** can also be created with one or other of the graphic editors provided with the CDP System: BRKEDIT.EXE or the graphic breakpoint editors in the respective GUIs. BRKEDIT provides a facility to audition a breakpoint file as time-varying amplitude or frequency.

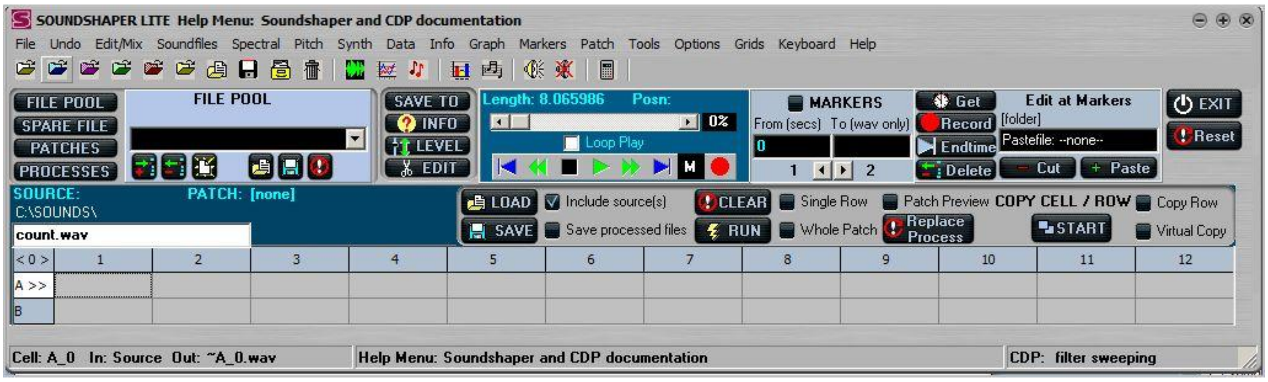
- **CDP HTML Reference Documentation**: the main index page is **index.html**, which actually links to the previous main index, **ccdpndex.htm**. It is recommended that you create a Desktop Shortcut for **index.html** as well as for *Sound Loom* and/or *Soundshaper*. You can access the CDP documentation from within *Soundshaper* under the menu **HELP\CDP Help**. Also note that *Soundshaper* gives the name of the CDP program being used, at the bottom right of its window. The actual *Soundshaper* reference manual is in **\docs\Soundshaper.**

- You may also want to have your favourite **soundfile editor** available (see above), as well as an **audio sequencer** into which you can put the results of CDP's extensive sound transformation processes. *Cubase*, *Pro-Tools*, *Logic Pro* and *Sonar* (among others) all offer fantastic facilities (we assume that most users will have hybrid systems), as well as a huge range of synthesis possibilities. Soundfile editing and mixing can also be done within the CDP software.

- **Optional:** One can also use the CDP programs from a **command line interface (CLI)**: the 'Terminal' on MAC or an **MS-DOS window** on PC. This can be useful for several reasons. You can access CDP system programs such as DIRSF and COPYSFX, you can create shell scripts or batch files (**.sh** on MAC, **.bat** on PC) which can be used to run several processes at once (see the SDBATS demo for examples) and build up a 'library' of your most effective process sequences. Other scripting languages that can be run from a command shell include Perl, Python, TCL/Tk, and *Tabula Vigilans*. We are also exploring the Raspberry Pi and hope to compile CDP on this platform. These are suited to the control of CDP programs in a more algorithmic way, whether simple or complex. These can all be programmed and run from a specialist text editor such as EMACS. Some of the most advanced work with the CDP System, such as data sonification, is being done *via* the CLI and scripting languages.

- Finally, you may also be interested in exploring ***ProcessPack***, a TCL/Tk 'umbrella' program under development since 2007 by Richard Orton and Archer Endrich working as Wellspring Music. It is now available via the CDP Website. It hosts a growing number of modules designed as high-level composition tools, i.e. more directly concerned with shaping passages of music, with algorithmic-based shaping tools. It is very complementary to CDP, which also has a significant algorithmic dimension. In addition to new facilities, they also make use of a number of CDP programs as internal components, thus integrating them into a broader composition context.

To launch ***Soundshaper*** you just double-click on the program name or its shortcut icon. To launch *Sound Loom* you just double-click on *soundloom.app* (Mac) *or soundloom.exe* (PC) or its shortcut icon. The image below shows the main page of *SoundshaperLITE.*



NB: The graphic interfaces construct command lines from the information you provide in their dialogue boxes. They then pass this command line to the underlying command prompt for processing that CDP function. Very little processing is built into *Soundshaper* or *Sound Loom* – but do note the very powerful MUSIC TESTBED facilities in *Sound Loom*.

The Time Domain functions take soundfiles as input and produce soundfiles as output. The Spectral (Frequency Domain) functions take analysis files as input and produce analysis files as output – except PVOC ANAL which will analyse a soundfile and PVOC SYNTH which re-synthesises an analysis file back into a soundfile. In *Soundshaper*, select SPECTRAL > CONVERT, which takes you to **Analyse** or **Synth**(esise).

Other graphic programs such as *GrainMill* or *Brkedit* can be run by double-clicking either on their name/icon or on their shortcut on the Desktop.

When working with the CLI, you enter program names (and command lines) at the prompt. When using this facility, it is important that the path to the CDP programs be set for the computer system as a whole, so that the interpreter can find the programs from any working directory. Note that we have provided a document about using command lines accessible via the CDP index with the name *Using MSDOS.*

SECTION 3 – Additional Documents and Charts

Especially in the initial stages of using the system, the full documentation is useful for explaining the meaning of the parameters in greater detail and providing some background on what sound transformations the program can be expected to achieve.

As mentioned above, you can access the Reference and Tutorial Documentation in various ways: via **index.html** (shortcut on the Desktop!), **HELP\CDP Help** in *Soundshaper*, INFO buttons in *Sound Loom*, and information about programs and the meaning of parameters provided at the bottom of the screen on the Status Bar. **Index.html is the Main Index for the whole of the Release 7 System**. Once you call up this Main Index, you can access most parts of the documentation from it, returning to the Main Index either with 'Back' on your browser, or with a 'Return' link within the document.

The CDP documentation has a **button index** (*cbutnndx.htm*) which takes you to the reference documentation for the function name on the button, an **alphabetical index** (*calphndx.htm*) with which you can look for a certain type of process, and a **descriptive index** (*cdscrndx.htm*) that goes through the functions group by group. All these can be accessed via **index.html.** When wondering what facilities might be available to achieve a certain musical goal, it is often most useful to peruse the **CDP Desk Reference**. This is in PDF format and is available for you to print out, if you wish. It is located in **\docs\DeskReference**. Printed and wire-bound copies may also be available for purchase.

We have not built the documentation into the programs themselves mainly for flexibility of use and inter-platform portability, except for *GrainMill,* which has built-in Help as well as (the more complete) HTML documentation.

The CDP System is presented in a series of Program Groups designed to bring you mentally closer to the function you need for a specific compositional task. Many of the newer programs are actually separate executables, indicated in the Desk Reference by being enclosed in square brackets (e.g. [Filtrage]). However they are placed within the listing and manuals for the functional groups in order to show what they are for. The 'Musical Applications' sections of the reference manuals for most of these new, separate programs has been left blank for the time being. We need more experience with the programs and hopefully, some feedback from users, before filling these in sensibly.

There are therefore several ways to orient yourself regarding the functions available in Release 7 and what they are for. Besides the various indices described above, please note the following:

- CDP-LITE links in *index.html* provide a 3-level guide to about 40 basic functions arranged by musical task: Level 1 is a very general description, Level 2 provides More Information, and Level 3 is the full Reference Manual.
- There are three major overviews of the CDP software which you can use to orient yourself in the very extensive range of processing functions: the *Getting Started with CDP in 12 Steps* tutorial by A Endrich (with sound examples), *A User's Guide to the Sound Loom and CDP Sound Transformation Software* by T Wishart, and *CDP: A Guided Tour* by R Fraser. (These are named more succinctly in *index.html*.)
- The various CDP tutorial documents and sound example files are useful until you become more experienced with the software. Several are offered with the CDP System and are described below. There are also at present two CDP Tutorial Workshops by Archer Endrich which are also being made a free download with Release 7. They offer a wealth of information and sound examples. *Tutorial Workshop CD 1* goes through over 170 worked examples of CDP sound transformation processes. *Tutorial Workshop CD 2* is designed for experienced users ready to delve more deeply into the TEXTURE Set programs. It seeks to connect the program set's facilities with musical objectives.
- We would welcome any tutorial contributions from users, especially videos.
- Please note the presence of two glossaries, one for technical terms (*glosstfrm.htm*) and one for musical terms (*glossmfrm.htm*). The latter seeks to make connections between the facilities of the CDP software and a range of possible musical composition objectives. Both of these glossaries can be accessed from *index.html*.

There are a number of **Reference Charts** that can be accessed from *index.html*. These charts are separate HTML files in the \html folder. You might want to print out the ones you find you use the most and have them laminated, for permanent easy-access desk reference. They include:

- **Notechrt.htm -** Equivalent Pitch Notations: MIDI, *Csound* and Frequencies. A special version of this chart is provided for printing neatly on 2 sheets (single or double-sided).
- **Gadbchrt.htm** – Chart of Gain – dB Correlations (as illustrated above)
- **Trnschrt.htm** – Chart of Transposition Ratios
- **Ndfchart.htm** – Note Data File Chart for the Texture Functions
- **Txcharts.htm** – 2 Useful Reference Charts for Texture
- **Rpchchrt.htm** – Transposition & Shifting Inputs and Outputs

SECTION 4 – Additional Materials for Background Information

We provide a number of additional tutorial documents intended for general information, orientation and getting started with the software. These can also be accessed via *index.html* (the files are in **\docs\htmltuts**). They take a look at the CDP software from various perspectives. You may find that one or the other suits the way you look at things.

- **CUSEFRAM.HTM** – This gives an introduction to sound design, presents a number of strategies with which to approach this kind of work, and defines a few basic concepts.
- **CKEYFRAM.HTM** – Using the 'keyhole' idea, this takes you through several specific parts of the system, generally moving from basic to more complex functions.
- **LANDYEAM.HTM** – about Sound Transformations in Electroacoustic Music, by Leigh Landy.
- **OPERPVOC.HTM** – *The Operation of the Phase Vocoder*. This is a technical document by Richard Dobson explaining some of the intricacies of the Phase Vocoder analysis software.
- **SDMANUAL.HTM** – This 'Study Guide' by Archer Endrich places the CDP software in the broader context of sound design for film. It was created in 2000 as part of the work for a Millennium Award to give workshops in sound design.

- **ADTOCDP.HTM** – Cross-references Trevor Wishart's *Audible Design* with the corresponding CDP functions.
- **CMA1FRAM.HTM** – This tutorial provides background information relevant mostly to command line use of the System.
- **CMSDOS.HTM** –for users new to running the command prompt CLI, or new to the whole concept of paths, this document goes through the basic operation of this kind of environment.

SECTION 5 – Demo and Tutorial Materials in greater detail

We are gradually building up more materials that show how sounds are made with the CDP software and cover some of the more advanced parts of the software. These include:

DEMO – The following materials can be accessed via *index.html*. In Release 7, the source sounds are in wav format and the examples are in mp3 format.

  - **Demo of CDP Sound Processing:** ('DiceDemo') – *indexpgs.htm* - Examples from Archer's composition, *Crossing the Dark Rift*, several of which use The TEXTURE Set. (The *GrainMill* sound example 'Bigone' is also here.) This tutorial goes through the construction of the final result in a step-by-step manner, thus showing how the CDP software can be used to gradually sculpt a sound towards the sound imagined by the composer.
  - **Worked Examples in 9 Categories:** *CDPExamples.htm* – source and result sounds, input text files used, parameters and commentary.
  - **Multi-process Batch Files**: – Six 'Sound-Builder' Templates with an initial set of examples: *reentryS-B1.htm*, *churblesS-B2.htm*, *grnflowS-B3.htm*, *glissswpS-B4.htm*, *rmrepeatS-B5.htm* and *widentrnS-B6.htm*. The idea of these examples is to illustrate chains of processes: the output of one process becomes the input of another. This is where the real power of the CDP software is to be discovered. (These include batch files with which you can use your own sound inputs to run through the same process sequences.). Batching mechanisms are also available within the GUIs. Building your own libraries of batched modules will refine your working methods and speed up future work.

- **GrainMill** – When getting started with *GrainMill*, we recommend going through:
  - **Grntutor.htm** – a marvellous tutorial introduction by Philippos Theocharidis (*\philippos*), written when he was at Newcastle University. His excellent examples give a good indication of what is possible with this granular manipulation program.
  - **Rdplayex.htm** – The *\rwd* folder contains Richard Dobson's examples, which are useful to show how to use *GrainMill* for very specific and controlled effects.
  - **Aebigone.htm** – in *\ae*, Archer's 'AEBIGONE' example illustrates the use of multiple breakpoint files timed to expand the original sound into an ever-changing granular texture.

OTHER TUTORIAL DOCUMENTS

- **Texture Pack** – is supplied within the Main Reference documentation (**\docs\htmltuts\Txpack**) to help you get started with the complex but enticingly powerful programs of the Texture Group. It includes all the breakpoint and note data files needed to run the examples discussed in the HTML reference document for Texture, *cgrotexf.htm*. This package includes batch files to make and delete the examples, but you can also use the **Preset** mechanism built into *Soundshaper* or the **Patches** supplied with *Sound Loom*. The text form of the batch files provides a concise listing of all the examples and their files. *Readtxtu.txt* provides an overview of the examples and their associated files.
  - **Drunk – (\docs\htmltuts\Drunk)** – This is a set of 6 examples that illustrate creating sounds with EXTEND DRUNK (the one in the Soundfiles/Groucho section), with extensive use of breakpoint files. This adaptable program, based on a Miller Puckett original, makes use of time-varying breakpoint files for a number of parameters, making the interactions of the parameters rather complex. This tutorial takes you through a number of pattern types.

- **3 Csound Tutorials** - *Csound* orchestra and score files, along with introductory material, by Andy Hunt. If you are new to *Csound*, these form a great introduction that will help you to get started using it.

- Also please recall the availability of the two Tutorial Workshop CDs mentioned above.

SECTION 6 — ADDITIONAL HELP

Along with the distribution of CDP Release 7 as a free download, we are also starting a CDP Online Forum at : **http://unstablesound.net/cdpforum**. We expect this to be the first port of call for any questions about the using the software or about problems encountered. Please report any problems providing the following information:

- program name,
- parameters used
- type and length of the input file
- and any error messages received.

If you post queries & problems to the CDP Discussion Group, a wide range of expertise may be brought to bear on solving your difficulty. And we shall also respond when needed.

- Information & *Soundshaper* updates at: **http://www.ensemble-software.net**
- Information & *Sound Loom* updates at: **http://www.trevorwishart.co.uk**
- Check the CDP Website: **http://www.composersdesktop.com**
- You can find the latest version of and information about ***Csound*** at **http://www.csounds.com**.

> Note the fantastic new learning tools *The Csound Book* and *The Csound Catalog*, both edited by Richard Boulanger. *The Csound Book* has several contributions from prominent CDP users. Richard Dobson has also contributed about 150 pages to *The Audio Programming Book*, ed. Boulanger & Lazzarini.

SECTION 7 — ABOUT CDP

The CDP System has been designed by and for composers as a tool with which to push their work with sound towards ever more adventurous realms of innovation. There have been many contributors over the years, but most of the sound transformation software has now been conceived and written by composer-programmer Trevor Wishart, hence our calling this software the '**CDP-Wishart**' libraries. There has also been much technical input and maintenance by Richard Dobson, documentation written and/or edited by Archer Endrich, and GUI front-ends by Trevor himself (*Sound Loom* – MAC & PC) and Robert Fraser (*SoundshaperLITE* or *Soundshaper Pro* (PC only). We hope you'll have lots of fun and success exploring the world of advanced sound design – a core music technology for the 21st century.