CDP GUIDE: 5. OTHER FUNCTIONS

CATEGORIES:

5.1 SYNTHESIS   (CDP group: SYNTH)

CDP has only a small number of synthesis programs, being predominantly a sound-processing system. However, some of these are very powerful and the results can be played by **Sequence2**, provide waveforms for a sampler, or have envelopes or spectral envelopes imposed on them.

- **Wave** creates several traditional waveshapes: sine (single freq.), square (odd harmonics), triangle, ramp. Nowadays these are probably most useful as test tones or controlling waveshapes.

- **Chord** generates a chord of sine tones from a datafile of (possibly fractional) MIDI pitch values. The components cannot be weighted in amplitude; the intention is to allow you to hear what a particular combination of pitches might sound like (e.g. before setting pitches in **Stack** or **Tune**).
  However, because it can hard to distinguish one tone from another in such a mix, a keyboard or online **virtual keyboard** may be more useful.

- **Noise** generates white noise: a random selection of all fequencies. Use filtering to create "pink" noise and other filtered variants.

- **Silence** generates a silent file of any length. Although other functions can insert silence, it is still useful to be able to create silence of any length, for example to splice sounds separated by silence.

- **Spectra** synthesises a spectral band centred around a chosen frequency. This sounds like a warbled tone of varying pitch, within the band.

- **Newsynth 1/2** (Addsynth) is an additive synthesiser, generating a waveform with a time-varying user-defined spectrum of any number of (harmonic or inharmonic) partials. Mode 2 allows each wave-cycle envelope to be shaped, to a degree.
- **Newsynth 3** (Multichannel) generates pulsed selections of the user-defined spectrum, sent to a multi-channel output. Each output event is a random selection of partials from the overall spectrum and subject to octave-transposition over a specified pitch range.
- **Newsynth 4 (R8)** (Fractal) generates a buzzy tone made by fractalising the waveform. Selective filtering can be applied to the output to generate a range opf sounds.
- **Newsynth 5 (R8)** (Duffing) is an experimental Duffing oscillator.

- **SynFilt (R8)** sculpts filtered sound from noise, creating a single pitch or chord. The chord filtering is done using the very versatile time-varying VARIBANK format. Because white noise is randomly generated, the output is more naturally varied than some synthesis methods. You also specify the number of harmonics to generate for each pitch.

- **Impulse (R8)** creates a pitched pulse wave (a stream of impulses). The impulses can be gapped and altered in pitch, steepness of slope, level, number of peaks and degree of glissing, all time-variable. To create a series of audible clicks, raise the slope, keep the MIDI pitch below 20 and optionally raise the gap size.

- **Pulser Synth (R8)**: closely related to **Pulser/Multi (R8)**, this synthesises enveloped "sound packets" (short bursts of sound) at regular intervals to give a pulsed effect. The average time between packets controls whether a pulsating or continuous sound is heard. The synthesized waveforms are additive (up to 64 partials) and the spectrum can be static, time-varying or random (though the last two were not working at the time of writing.)

- **Multiosc (R8)**: an FM-style synth process with an one oscillator modulated by another, itself optionally modulated by third one, and that one by a fourth. Envelopes are sadly lacking.

- **SynSpline (R8)** synthesises a steady tone made from smoothly joined randomly-generated points. This is surprisingly successful, giving a gently fluctuating tone of a rather bright timbre (which can then be filtered).

- **Clicks** (*Soundshaper: ClickTrack*) creates a click-track from a datafile, described in the Reference and in **File Formats**
  If you simply wish to record a basic metronome track, there are readily available tools on the Web such as this **online metronome**.
- **Clicknew (R8)** produces clicks of very short pulses, from a simple set of times. For a continuous click-track, one would need a looping player (as in Soundshaper, for example), or a very long list of times. Any rhythm can be played, of course, as long as one can work out the times.

OTHER FUNCTIONS (Release 8)

- **Multisynth (R8)** uses a special score format to trigger and mix a limited set of given "instrumental" sounds, optionally across multi-channel space.
  The sounds are of poor quality and do not compare with the many free high-quality instrumental samples available.

- **Newscales (R8)** outputs a series of fixed-length (0.5 sec) tones (to different files), each defined by frequency and amplitude, and by timbre. The spectrum can be different for each note, or the same for all notes. However, a number of calls to NEWSYNTH Mode 1 would do the same job and add a DURATION parameter, while NEWSYNTH Mode 2 would allow the sounds to be enveloped.

Outside the **SYNTH** category (see **EDIT** functions), but not unrelated, are:

- **Waveform (R8)** extracts a segment to make a wavetable. Mode 1 etracts a group of half-*wavesets*, Mode 2 etracts any short segment (up to 10"), while Mode 3 superimposes the segment on a sine wave. In each case, the cut segment is abutted to its inversion, making a symmetrical wavecycle and doubling its duration.

- **Packet** isolates or generates one or more sound packets (short enveloped sound grains), suitable for use within the TEXTURE suite.

5.2 GENERATE DATA
(CDP groups: FILTER, MODIFY, REVERB)

Some utility programs to generate data files for various processes. Soundshaper's **Graph-Edit** is also a general-purpose page for generating data, and has several presets as well. (More will be developed in the future.)

- **BankFreqs** creates a list of frequencies for use with **Filter Userbank**. Amplitudes have to be added to the frequencies. (Soundshaper's **Graph-Edit** has similar presets giving a set of frequencies to which you can add the amplitudes.)

- **VFilters** generates a basic textfile for use with **Filter Varibank**.

- **Filtrage** generates random-frequency filterbank data for use with **Filter Varibank**, (the degree of randomness being under user control).

- **Roomresp** generates an early-reflection datafile for use with **Reverb / Roomverb**.

- **Spaceform** (*Soundshaper: Sinedata*) creates a sinewave data file, intended for use with Pan, but it can be used wherever a sinusoidal LFO is required.

- **DShift** calculates the necessary transposition data for a Doppler pitch-shift.

5.3 INFORMATION
(Main CDP groups: SNDINFO, SPECINFO, PITCHINFO)

CDP has a large number of Information functions which either report to the console or output a text file. A console report can be redirected (in Windows at least) to a text file using the re-direction operator **>**, e.g. *sndinfo props myfile.wav >myfile.txt*. A double operator (**>>**) appends the output to the specified file.

SOUNDFILES (CDP groups: SNDINFO, MCTOOLS, HOUSEKEEP)

These functions give information about a soundfile -- the general functions **FileInfo** and **FileProps** can report on most CDP file types. Some functions are specific to certain processes, notably GRAIN and DISTORT. (**GrnCount** or **GrnAssess** are especially important for use with Grain functions.)

General:

- **Sndinfo Props** displays properties of a soundfile, or other CDP binary file (**.ana, .evl, .for, .frq, .trn**). For a soundfile it displays:
  file-type, sample-count, sample-rate, sample-type (bits), channel-count, peak data per channel (if present)
- **Sfprops** similarly displays the properties of a soundfile, or other CDP binary file.
  file-type, format (e.g. standard or Wave_Ex)**, duration, sample-count, sample-rate, sample-type (bits), channel-count, peak data per channel** (if present)
  For WAVE_EX, **Sfprops** will identify 7.1 and cube speaker layouts.
- **Len** displays the duration of a soundfile, or other CDP binary file (**.ana, .evl, .for, .frq, .trn**).
- **Maxsamp/Maxsamp2** display the maximum level found: in the whole file (Maxsamp), or within a specified range of the file (Maxsamp2). The level is given on a scale of 0-1 and the maximum possible gain (to raise to 100%) is also given.
- **RMSInfo** reports the overall power level in each channel, displaying two calculations – RMS (Root-Mean-Square) and raw average (normalised sum of all samples) – as well as the average level of any DC. A range within the file can also be specified.

- **Findhole** reports the longest time that the signal falls below a given threshold level (if at all). Since several functions use a gate level – e.g. **Gate**, **Onsets**, **Gate-extract** (*housekeep extract 1*) – this helps to determine the threshold level at which a gate might take effect. (Raise the threshold if no holes are found.)
- **Onsets** (*housekeep extract 6*) lists in a textfile the onset times that the signal is above a given threshold level. These times can be used to extract segments (e.g. in other modes of **housekeep extract**), insert silences, retime events, etc. (Suitable parameter values may be: **gate** 0.035 and **minlength** 0.05).
- **Peakfind** lists in a textfile the times of peaks in a soundfile (needed for **Retime**). A threshold level is needed, to determine what constitutes a peak.
- **Find Start** (*retime 12*) finds the start-time of the sound (the first non-zero sample). This is useful for **RETIME** functions and mixing.
- **SigStart** compares multiple infiles and displays the earliest start-time in the set, when there is some signal. Sadly it does not say which of the sounds starts first.
- **Find durations** (*retime 11*) finds the longest and shortest silence-separated events in a soundfile, useful for **RETIME** and **MCHANPAN**. Events may be clearly visible in a soundfile viewer, but may not have absolute silence between them (especially where reverb is present) – it may be necessary to create silences using **Gate** or **Corrugate** (*envel warp 11*).
- **ZCross** displays the proportion of zero-crossings, with a higher proportion indicating more noise. The information may be useful for waveset **DISTORT** functions.

Comparative:

- **Loudest** (*modify loudness 7*) finds the loudest of a number of soundfiles.
- **Maxi** similarly displays the maximum level in several files, but outputs the levels (only) to a textfile. (Alterntively, in Soundshaper, you could select the files and invoke MIX: the **Maxsamp** level of each file in the input list is displayed. Hit CANCEL to leave MIX.)
- **LoudChan** displays which of the channels in a stereo or multi-channel file is the loudest. (Just displaying the levels would be more helpful!)
- **Lens** displays the lengths of several soundfiles (but not other CDP binary files).
- **Sumlen** sums the durations of several soundfiles and displays this.
- **Timediff** displays the time difference between two soundfiles. It assumes you know which one is longer.
- **Sndinfo Diff** compares two soundfiles (or other CDP binary files) and displays differences.
- **Chandiff** compares the differences between channels in a stereo file.

Process-specific:

- **Cyclecnt** [DISTORT] counts the number of pseudo-wavecycles in a soundfile. Distort functions often have number-of-cycles as a parameter.
- **Grain Count** [GRAIN] counts grains according to a given threshold, etc.
- **Grain Assess** [GRAIN] a friendlier count of grains at a suitable built-in gate level and window-count.
- **Grain Find** [GRAIN] aims to find grains in a grainy file. (Possibly faulty? On a test run, this gave a blank report.)

- **Grev 6** [GRAIN] reports grain times (in samples!). These can be used as input to **Put Grains** (*grain grev 7*)
- **Findpan** [PAN] finds the pan position at a given time in a stereo soundfile, by comparing the levels in each channel.
- **Stretcha** [STRETCH TIME] calculates the stretch factor for timestretch or shrink.

SPECTRAL (CDP groups: SPECINFO, SNDINFO)

These functions give information about frequency analysis files (**.ana**). Relating channels (frequency bands) to frequency (and vice versa) is particularly important, as is the Window count. There are also functions to track time-varying spectral peaks in the sound.

- **Sndinfo Props** displays properties of a frequency analysis file **.ana**, or other CDP binary file (**.evl, .for, .frq, .trn**). Note that the display lists the number of analysis *points*, but calls them channels. **SFProps** counts the channels correctly.
- **Sfprops** similarly displays the properties of an analysis file **.ana**, or other CDP binary file. For analysis files it displays:
  file-type, channel-format, orig. sample-rate, window-size, analysis channels, window overlap, analysis rate, frame count, data size, duration These parameters are explained in the **PVOC Manual** and *The Operation of the Phase Vocoder*.

- **Len** displays the duration of an analysis file, or other CDP binary file (**.evl, .for, .frq, .trn**). The length will not be quite the same as the original soundfile.
- **Windowcnt** displays length as the number of (time-)windows in the analysis file. This is important for those functions like **Blur** and **Weave**, which manipulate spectral data over time. Analysis windows overlap in time, so the window count depends on degree of overlap, as well as the duration of the file.
- **Freq to Chan** (*specinfo channel*) displays the relevant analysis channel number for a given frequency.
- **Chan to Freq** (*soecinfo frequency*) displays the centre frequency of a given analysis channel number. Analysis channels have a fixed width which depends on the analysis settings.

- **Octvu** displays the time varying amplitude of the spectrum, within octave bands.
- **Peak** (*Soundshaper: Peakamp*) locates the time-varying energy centres of the spectrum (the spectral peaks) and reports these to a textfile.
- **Report** (*Soundshaper: Peakfrq*) gives a textfile report of spectral peaks: of frequency and time, loudness and time, frequency only or loudness only.
- **Newmorph2 Mode 1** (*Soundshaper: Peakfreqs*) extracts the peak frequencies from a given analysis file, in order of prominence, to provide a pitch template for Newmorph2. It lists the frequencies only, not their amplitude.
- **Peak extract** extracts the spectral peaks from an analysis file and writes these to a textfile.
- **Get Partials** (*Soundshaper: GetHmcs*) extracts the relative amplitudes of harmonics in a pitched sound and writes these to a textfile.
- **Level** (*Soundshaper: Levelvu*) displays the level of an analysis file as a pseudo-soundfile (DO NOT PLAY THIS!)
- **Formants See** displays a formant file (**.for**) as a pseudo-soundfile to view. (DO NOT PLAY THIS!)
  GetSee does the same, but takes an analysis file as input.
- **Print** prints spectral data to a textfile. This generates *huge* amounts of data – try it with a single window first. (c.f. **PrntSnd**)

PITCH (CDP groups: SNDINFO, PITCHINFO)

These functions give information about pitch files (**.frq**), extracted from spectral files (**.ana**).

- **Sndinfo Props** and **Sfprops** can both display the properties of a binary pitch-data file (**.frq**), or other CDP binary file (**.ana, .evl, .for, .trn**). The information displayed is much like that for an analysis file. (**.ana**.

- **Pitchinfo Info** displays (as Hz and MIDI) the minimum, maximum and mean pitches found in the file, if any.
- **Zeros** displays whether the file has any unpitched windows.
- **Pitchinfo Hear** converts a pitch-data file (**.frq**) to an analysis file of the extracted pitch (**.ana**) so that it can be heard via **PvPlay**. An identical analysis file is also produced (as a second output) by **GetPitch**.
- **Pitchinfo See** converts a pitch-data file (**.frq**) or binary transposition file (**.trn**) into a pseudo soundfile to view. (DO NOT PLAY THIS!)

UNIT CONVERSIONS (CDP groups: SNDINFO, SPECINFO)

Various types of units can be converted, with the result reported to the console. These include:

- **MIDI to Hz, Hz to MIDI**
- **Freq. ratio to semitones** and vice versa (for **Speed** or **Transpose**)
- **Freq. ratio to time ratio** and vice versa (for **Speed**)
- **Semitones to time ratio** and vice versa (for **Speed**)
- **dB to gain, gain to dB**
- **Samples to time** and **Time to samples**
- **Freq to Chan** and vice versa (see above)

5.4 SYSTEM UTILITIES
(CDP groups: SYSUTILS, HOUSEKEEP)

Recording and Playback utilities (RECSF, PAPLAY, PVPLAY) have been listed under **Soundfile Utilities**.

FILE and OTHER UTILITIES

- **DirSF** lists the soundfiles present in a folder (directory). Windows' File Explorer has no direct way listing files, though the **dir** command on the command-line (console) does this and the list can be redirected to a textfile (add > *filename*). DirSF does the same for files it recognises as being soundfiles or other CDP binary files (**.ana, .evl, .for, .frq, .trn**). Wildcards may be used by both **dir** and **dirsf**, e.g. *dirsf *.frq* lists all pitch-data files. **Dirsf** does not recognise text files (e.g. .txt, .brk, pch, .env), but **dir** can list them.

- **Bakup** backs up a number of soundfiles as a single file, with silences inbetween.
- **Bundle** produces a text list of the selected soundfiles. (Some facility with a text editor would do instead.)
- **Copy** makes a copy of a soundfile. (Standard file-copy procedures in Windows / OSX are equally good.)
- **Remove** deletes copies made with Housekeep Copy and Housekeep Extract. These are numbered *filename*_001, _002, etc. Unfortunately, by no means all of CDP's multiple outfiles use this numbering.
- **Sort** sorts files listed in a textfile, in various ways.
- **Disk** aims to show the free space left on a disk after any file – perhaps largely un-necessary in these days of multi-gigabyte disks. (This function is currently returning an error.)
- **Batchexpand** expands an existing batch file in various ways. Batch files (**.bat**) and Bash scripts (OS-X) are a powerful way of carrying out a sequence of CDP processes. (For instance, Soundshaper writes a custom batch file for each process it runs.) However, as they are easily edited in a text editor, the user must evaluate this utility for him/herself.
  Windows **cmd** and OSX/Linux **bash** commands are well documented at the **ss64 website**.
- **Getcol** is a textfile utility which extracts a column in a textfile (such as a list of times in a breakpoint file of *time value*) and saves it to another textfile.
- **Putcol** is a textfile utility which puts a column into a textfile, from another textfile.
- **Alias** provides a shortcut to a soundfile on PC systems (only). On the command-line, this saves typing the whole filename and path each time the file is used.
  (This function is documented, but has not been updated for a while and is possibly no longer included in the CDP package.)
- **Env View** (*housekeep extract 2*) displays the envelope as a pseudo-soundfile (not to be played). This can be viewed in CDP's **VIEWSF** or any soundfile viewer.
- **PrntSnd** prints all sample values (levels) to a (large) textfile, with optional start and end times (recommended).

TRANSFORM DATA

Columns

CDP's **COLUMNS** program has around 100 data transforming/generating functions. (Graphic versions are implemented within SoundLoom's **Table Editor** and Soundshaper's **Data-Editor**.) The **COLUMNS** functions transform columns of text data (such as MIDI pitch lists or *time value* breakpoint files) and are not concerned with the appropriateness of the resulting data for any particular purpose: the user must ensure that ranges and values are suitable for the intended parameter.

There are broadly five types of function:

- **Convert**: these convert one form of data into another – e.g. Midi to Hertz, Multiply values
- **Sort**: these re-order or group data in some way: for some functions the output is longer or shorter.
- **Generate**: these generate data – e.g. Generate harmonics, Generate accel. times – and typically require a start and end value as input.
- **Information**: e.g. Find greatest, sum times. Processing functions such as Midi-Hz or Sample-count to times can also be used as information functions on a column of data.
- **Table processing** e.g. Files to Columns, Concatenate files.

The various operations fall into these categories:

- **MUSICAL**
  1. Convert Pitches and Intervals
  These transform pitches, frequencies and intervals in familiar ways. For example:
  Midi to Freqs / Freqs to MIDI: convert MIDI values to Frequencies or vice-versa.

  2. Reorder or generate pitches
  These change the order of elements in the column or generate new groups based on the input list. For example:
  Motivically rotate list: cyclic rotation by N shifts.

  3. Process Times and Levels
  These mainly process onset times and durations. For example:
  Times from durations: produce a list of onset times from a list of durations.

- **MATHEMATICAL**   For example:
  Multiply: multiply all values in the list by a constant value
  or
  Approx. Multiples: quantize all values to multiples of given value, e.g. 221, 442, 663, 884 quantised to 110 produces 220, 440, 660, etc.

- **GENERATIVE**   For example:
  Log equal divisions: where 4 divisions between 10 and 20 produces 10.000000, 11.892071, 14.142136, 16.817928, 20.000000 or
  Repeat intervals: Starting with the last value in the list, the intervals are duplicated, e.g.(MIDI) 54,56,57 duplicated 4 times gives 54,56,57,59,60,62,63,65,66,68 - an octotonic scale.

- **RANDOM**
  These modify values by a random amount or generate random values. For example:
  Scatter: create variants near the original values, e.g. 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 with a scatter value of 0.5 produced 10.000000, 20.510651, 29.267479, 40.762581, 48.124409, 57.565615, 70.957900, 79.245048, 90.660039, 100.000000.

- **SORTING**
  These functions re-order the items in the list. For example:
  Skip every Nth item: e.g. KEEP 3 skips 1 in every 4.

- ○ **INFORMATION**
  These functions produce information about the items in a list. For example:
  Find greatest: finds the maximum value in the list
  or
  Text to Hz: produces a list of frequencies from a list of pitch names.

- ○ **TABLES: MANIPULATE COLUMNS & TEXT FILES**   For example:
  Separate Columns: split a multi-column file (e.g. breakpoint) into separate columns
  and store these as distinct files.

Vectors

The **Vectors** program provides a further range of thireen operations on columns of
numbers. Some of the these appear to duplicate those in **COLUMNS**; for details see here:
Vectors

FIRST SECTION: SOUNDFILE PROCESSES