CDP GUIDE: 4. EDIT / MIX PROCESSES

CDP does not come with its own visual sound-editor (although both main GUIs include one), but its Edit and Mix functions include all the fundamental operations found in those editors, such as cutting, pasting and extracting segments, as well as those associated with the mixing desk: spatial distribution and channels, level and balance, as well as mixing sounds. A growing number of functions extend these principles into less familiar areas, such as multi-channel spatial distribution. Finally, this section catalogues some soundfile utilities.

EDIT / MIX CATEGORIES:

4.1 EDIT SOUNDFILES (CDP groups: SFEDIT, HOUSEKEEP, ENVEL, EXTEND, MCTOOLS)

Cut and paste sounds, including extracting segments to separate files. Other processes mask portions with silence and switch between soundfiles.

CUT FUNCTIONS

- **Cut** extracts a segment as a separate soundfile, using given START and END times. **Cutend** measures the cut from the end of the file.
- **Zcut** (*Soundshaper: Zerocut*): as **Cut**, but makes the cut at the nearest zero crossover points.
- **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.
- **Excise** cuts out and discards the selected segment.
- **Excises** (*Soundshaper: Discards*) discards listed portions of a sound.
- **Topntail** (***housekeep extract 2***) trims the sound, removing low level signal from start and/or end. It uses a **GATE** level, above which the signal is accepted.
  TopanTail2 (*Soundshaper: Trim*) is a newer and more versatile trimming function, with separate start and end gate levels and a backtracking option.

CUT WITH MULTIPLE OUTPUTS

- **Cutmany** (*Soundshaper: Multicut*) cuts listed portions of sound and extracts these to separate files.
- **Zcuts** (*Soundshaper: Zerocuts*) makes cuts at nearest zero crossover points and extracts these to separate files. LIke many CDP functions, this requires a text *times file* of segment start and end times, though these can be approximate as the function will find the nearest zero crossings.
- **Syllables** is a variant of Cutmany, aimed at extracting syllables, which tend to run into each other. It allows a degree of dovetailing before and after the segment times specified. However, a good sound editor is needed to set the cut-times, and in the same editor you can just test and cut the syllable(s) you want. (Not recommended.)
- **Gate-extract** (***housekeep extract 1***) cuts the soundfile into separate files, using a **GATE** level to find cut points. Parameter values can be tested in advance by running Mode 6, which writes the times of events to a textfile. (Setting GATE=0.035 and MINLENGTh=0.05 seems to produce good results.)
- Housekeep Gate (*Soundshaper: ZeroExtract*) extracts segments to separate files, cutting at zero amplitude points (there is no GATE).
  WARNING: this function tends to extract hundreds of tiny files!
- **Timegrid** segments the soundfile using a constant time-interval, but each outfile is the original length, with the segment preceded and followed by silence. Like Partition and Isolate, this allows the segments to be treated individually and remixed.

- **Envcut (R8)** cuts the sound into enveloped segments which are adjacent or separated by a time-step. Like **Timegrid**, there is a regular segment length.
- **Distcut (R8)** cuts the sound into waveset groups which are adjacent or separated by a time-step. The regular segment length is measured in pseudo-wavecycles and each is given a falling envelope.
- **Randchunks** cuts the soundfile into a given number of separate files, with segments chosen randomly according to a given length.
- **Randcuts** is similar to **Randchunks**, but the number of cuts decided by average segment length (itself randomly variable). This function can generate a large number of output files.
- **Packet** isolates or generates one or more sound packets (short enveloped sound grains), suitable for use within the TEXTURE suite.
- **Partition** and **Isolate** both partition sound into separate output streams: Partition by wavesets, Isolate by cut-times or level. The original segment times are retained by inserting silences, so that the outputs may be treated separately and remixed. **Rejoin** re-mixes the segments created by Isolate.
- **Cantor** cuts holes (reduced level) in the manner of a Cantor Set (holes within holes). The output is a series of sounds with more and more holes cut.

PASTE

- **Insert** inserts a second sound at the given time, either overwriting or spreading the original. Allowing for the splice length can sometimes be an issue.
- **Replace** discards a portion of the source (specified by start and end times) and replaces it with another sound. The second sound can be longer or shorter than the segment it replaces, as it is a straight substitution: the duration of the original segment is not maintained.
  Unfortunately, CDP has no mechanism for processing just part of a soundfile: you have to extract the segment, process it and then insert the treated sound, overwriting the original. (Soundshaper does this as a composite procedure [SEGCUT plus SEGPASTE], but it is still messy compared with most editors where you simply highlight a section and alter it. The problem is caused largely by the need for output files, with no provision for simply holding the processed version in memory.)
- **Insil** inserts silence into the soundfile (except at the start or end).
- **Prefix Silence** (*Soundshaper: Presil*) adds silence to the start of the soundfile: a useful pre-mixing function, as it delays the sound relative to others in the mix.
- **Silend** (*Soundshaper: Endsil*) adds silence to the end of a soundfile. This is often done prior to adding reverberation, which needs room for its "tail".
- **Manysil** (*Soundshaper: Silences*) inserts multiple silences at a list of times.
- **Noisecut** aims to replace signal noise with silence. Unfortunately, it is hard to get it to recognise noise. (Not recommended.)
- **Masks** overwrites listed portions with silence (hence the sounding portions retain their timings). A number of different Masks mixed together can create an interesting rhythmic and/or spatial sequence.
- **Twixt** and **Sphinx** (*Soundshaper: Switch*) switch between different soundfiles running in parallel – at specified times, by permutation, or randomly.

SPLICE AND SEQUENCE

CDP has several functions which realise a sequence of sounds just like a MIDI sequence, using different sound-sources. **Sequence2** has the fullest set of parameters and plays multiple sounds. Three other sequence functions use less detailed data files.

- **Join** splices two or more soundfiles together end to end and is the normal workhorse for this basic task. Sounds are joined in the order given as inputs. The input sounds must have the same channel-count.
- **Njoin**: an alternative splicing program, designed particularly for multi-channel sounds. The files to mix are listed in a textfile and must match in sample rate and in number of channels.
- **Joinseq** joins soundfiles end-to-end following a number sequence, with sounds identified by their order of input. The whole sample is played each time: there is no duration field.
- **Joindyn**: as Joinseq, but the data file adds a field for level.

- **Sequence** (*Soundshaper: Sampleseq*): like a sampler, this plays a sequence of a single sound sample using a text file of:
  time, transposition *(in semitones),* ***relative level*** *(<1 or >1)*.
  The whole sample is played each time: there is no duration field.
- **Sequence2** (*Soundshaper: Sequencer*) is a multi-source sequencer using a text data file of:
  sound-number, time, pitch, level, duration.
  This is clearly the most versatile of the different sequencing functions, though it lacks any spatial provision. The sound lasts for the duration specified, whether it is finished or not.

4.2 LEVEL (CDP groups: MODIFY, HOUSEKEEP, ENVEL, MULTI-CHAN, DISTORT)

Various functions for adjusting gain, inverting phase and gating sounds. (See also **Envel Warp** functions which do the same sort of thing):

- **Gain** (***modify loudness 1/2***) sets or alters the level in a soundfile: values are 0-1 or in dB (<=0).
- **Normalise** (***modify loudness 3***)raises or lower the maximum level to the given level. (c.f. Envel Warp 1)
- **ForceLevel** (***modify loudness 4***) forces the level to the given level or the maximum possible.
- **Equalise** (***modify loudness 8***) forces the maximum level of all files to that of loudest sound.
- **Balance** (***modify loudness 5***) forces the maximum level to that of a second sound.
- **Multi-channel Gain** (***frame shift 7***) applies gain to one or more specified channels in a multi-channel file.
- **Phase** inverts the phase of the signal: positive becomes negative and vice-versa. An alternative function is **Modify Loudness 6**. An inverted signal mixed with the original produces silence. (This is a useful way of testing whether two processes produce exactly the same result.)
- **ChanPhase** inverts the phase of one channel. This solved a problem specific to SoundLoom, but might have other uses.
- **Gate** cuts out any signal which falls below a given threshold level (c.f. **Envel Warp 8**). The gated signal is either replaced with silence or removed altogether, shortening the file-length. (N.B. this function is not the same as HOUSEKEEP GATE, which extracts segments to separate files using a gate.)
- **Clip (R8)** distorts the signal by clipping it (cutting off the top of the waveform) when it reaches a certain level.
- **Overload** similarly clips the top of the signal, optionally replacing it with a waveform.
- Quirk (R8) distorts the signal by raising sample values to a power.

4.3 CHANNELS (CDP groups: HOUSEKEEP, MODIFY, MCTOOLS, SFEDIT, SUBMIX, MULTI-CHAN)

These processes handle channel operations: split a sound into its separate channels (or selected ones); interleave files to make a stereo or multi-channel file; convert between mono, stereo and multi-channel files; manipulate channels in multi-channel space by rotation, renumbering, etc.

- **Extract chans** (***housekeep chans 2***) splits (channelises) a stereo or multi-channel file into separate monos, from where they can be processed individually.
- **ChannelX** similarly extracts the selected channel(s) from a stereo or multi-channel file, to separate mono soundfiles.
  An alternative function extracting one channel (***housekeep chans 1***) is also available.
- **Pairex (R8)** extracts any two channels, as a stereo pair, from a multi-channel file.
- **Mirror** (***modify space 2***) swaps channels in a stereo file. (The function will accept more than two channels, but returns a stereo file.)
- **Subtract** subtracts a mono file from one channel of another (stereo or multi-chan) file: a solution looking for a problem?
- **Zero chan** (***housekeep chans 3***) zeros the signal in one channel of a stereo file.

- **Interleave** or **Interlx**: these functions merge up to 16 mono or 8 stereo files to make a (Wave-Ex) multi-channel file. **Interleave** requires mono inputs whereas **Interlx** can merge stereo or multi-channel files, with the same channel-count.
- **Repair (R8)** is a quirky function for interleaving mono files of the same length. Four sources 1,2,3,4 with *outchans*=2 produces two stereo outputs: 1+3 and 2+4; with *outchans*=4 it outputs a single 4-channel file (1+2+3+4). With 12 inputs and *outchans*=4, there are three outputs: 1+4+7+10, 2+5+8+11, 3+6+9+12.
  It is surely much easier just to use **Interlx** to interleave mono or stereo inputs of different lengths in any order you like.

- **MtoS** (***housekeep chans 5***) converts a mono file to stereo: the channels are identical.
- **StoM** (***housekeep chans 4***) convert a stereo or multichannel to mono: the channels are mixed. There is an option to invert channel 2 before mixing (ignored in a multi-chan file) – see also ChanPhase above.
- **MtoN** converts mono to multi-channel, with the same signal in all channels.
- **StoN** is a Soundshaper composite function using **Interlx**, duplicating a stereo file in chans. 1-2, 3-4, 5-6 etc. of a multi-channel file.
- **MchStereo** places several stereo files within a multi-channel space.
- **ChOrder** re-orders channels within a multi-channel soundfile.
- **Frame Shift** has several modes to move channels around in a multi-channel file: changing order, swapping, mirroring and rotating.

4.4 SPATIAL (CDP groups: MODIFY, MULTI-CHAN, MCTOOLS)

Spatial processes pan sound across stereo or multi-channel space, including ambisonics. The multi-channel functions either distribute a mono sound across multi-channel space, or re-order the spatial distribution within a multi-channel file (e.g. by changing channel order or rotation).

- **Pan** (***modify space 1***) spatialises a mono soundfile within stereo space, using a fixed value or breakpoint datafile of pan positions. Pan values can simulate space beyond full left or right, by attenuating the sound. (Soundshaper permits different parameter settings for each channel.)
- **ScaledPan** is a variant of Pan, in which a breakpoint pan file is automatically scaled in time to fit the length of the input sound.
- **Panmix** is a Soundshaper composite function: two mono inputs are each placed in stereo space using **Pan**, then mixed together. This simulates the mixing-desk, where each mono channel is panned before mixing.
- **Narrow** (***modify space 4***) narrows or inverts the stereo image of a stereo sound.
- **Phase Shift** (Phase Mode 2) enhances stereo separation by shifting the phase of the signal.
- **ToStereo (R8)** gradually diverges from a mono mix to full stereo. Input and output can be stereo or multi-channel.

Pan Data Functions

- **MirrorPan** (***modify space 3***) inverts the stereo image in a pan breakpoint file.
- **Findpan** (INFO function) finds the pan position, at any time-point, within a stereo file created using Pan.
- **DShift** (DATA function): from a pan breakpoint file, this calculates the necessary transposition to simulate Doppler shift.

MULTI-CHANNEL

- **MchanPan** (***mchanpan 1***) pans a sound in multi-channel space.
- **Rotate** (***mchanpan 9***) rotates a sound in multi-channel space. (See also ABFPAN below.)
- **Spin Stereo (R8)** rotates a stereo image across stereo space or 3 channels of multi-channel space, both with optional doppler shift. At very slow speeds, this can give a general stereo enhancement instead of a noticeable rotation. (See also Spin Quad below)
- **Spin Quad (R8)** rotates two stereo images across 5 channels of multi-channel space, with optional doppler shift.
- **Pan Set** (***mchanpan 7***) pans from one set of channnel positions to another, passing through the centre.

- **Spread** (***mchanpan 4***) spreads a sound across multi-channel space, outwards from a central channel.
- **Antiphon** (***mchanpan 6***) pans antiphonally between sets of channels. The set switches after a specified time.
- **Spread Events** (***mchanpan 3***): as MchanPan Spread, except that the spread is of silence-separated events.
- **Panorama** places (mono) files spatially in surround-sound. To place a single stereo sound, channelise it first; to place one mono sound, copy it first. Natively, **Panorama** produces a multi-channel mixfile (**.mmx**) for use with **NEWMIX** (see below). (Soundshaper also puts it in a batch file that includes NEWMIX and produces a soundfile.)
- **Switch Events** (***mchanpan 2/5/10***): the input switches between output channels at the ocurrence of each new silence-separated event, with antiphonal and random options.
- **TANGENT** is a subgroup of four functions that aim to produce the effect of a repeated sound or sounds passing across multi-channel space at a tangent to a circle of 8 loudspeakers. Natively, the **TANGENT** functions produce a multi-channel mixfile (**.mmx**) for use with **NEWMIX** (see below). (Soundshaper also puts some of them in a batch file that includes NEWMIX and produces a soundfile.)
- **TRANSIT** is a subgroup of six functions that aim to produce the effect of a repeated sound or sounds passing into and across 8-channel space, with possible filtering and/or doppler-shift. (A ring of speakers is presumed.) Natively, the **TRANSIT** functions produce a multi-channel mixfile (**.mmx**) for use with **NEWMIX** (see below). (Soundshaper also puts some of them in a batch file that includes NEWMIX and produces a soundfile.) It might be worth comparing both TANGENT and TRANSIT with the possibilities offered by MCHANPAN, which can also pan a repeated sound across multi-channel space.
- **Ambisonic Pan** (***abfpan/abfpan2***) generates a fixed or simple periodic rotating pan around the listener, using ambisonic encoding. It can also generate a rotating or fixed multi-channel pan without ambisonic output.
- **Ambisonic Decode** (***fmdcode***) decodes a 1st or 2nd order B-Format (ambisonic) soundfile to a choice of speaker layouts. A suitable input file may be created using **Ambisonic Pan**. There is a wide range of output options and speaker layouts.

4.5 MIX
(CDP groups: SUBMIX, MCTOOLS, MULTI-CHAN)

There are three categories of mix processes in CDP:

- Simple mixing functions avoiding the use of a mixfile
- **Mix** and **Newmix**, which employ a mixfile
- Mixfile functions that manipulate mixfiles.

CDP Mix functions are best for mixing small sounds to create larger units. Although larger mixes are perfectly possible, a visual editor is more intuitive and tends to have better controls.

SIMPLE MIXING FUNCTIONS

- **Mergemany** (*Soundshaper: Mixmany*) mixes several files with the same number of channels.
- **NMix** (*Soundshaper: MchMixtwo*) is a simple mix of two mono, stereo or multichannel files, with an optional time OFFSET. Files must have the same channel-count.
- **Merge** (*Soundshaper: MixTwo/MergeTwo*): a quick mix of two soundfiles (mono, stereo or multi-channel), which can have different channel counts (although output is always mono or stereo). There are a number of useful optional parameters.
- **PreMix** is a Soundshaper composite function setting Start-time, Level and Pan prior to mixing with e.g. Merge or MergeMany. These are same three parameters used in a mixfile; it is often more convenient to pre-set and test these values before mixing. It also allows Level and Pan to be altered throughout the mix, which the MIX program does not.
- **Balance** mixes two sounds, with a balance control and optional Start and End times. Mono and stereo soundfiles may be mixed freely, but multi-channel files must have the same channel-count. The balance control can vary the effect of any process mixed with its source, and do so over time.
- **Crossfade** performs a quick cross-fade between two sounds with the same channel count.

- **Faders** mixes a number of sources with a time-variable level for each, the equivalent of movable faders on a mixing desk. The level variations are entered in a textfile of:
  time, level#1, level#2,...etc.
  Levels are continuously varied between different values and are automatically scaled to avoid overflow.
- **InBetween/2** generates multiple outputs representing the changing balance between two inputs. **InBetween2** attempts to synchronize the zero-crossings in the two input soundfiles.

MIX USING MIXFILE

CDP has two dedicated mixfile formats for mixing: one for mono/stereo and the other for multi-channel output.

- **Mix** mixes together the soundfiles listed in a text mixfile (**.mix**). This has any number of lines of:
  soundfile-name, start-time, level_chan1, pan_chan1, [level_chan2, pan_chan2].
  Note that level and pan are fixed for the duration of the mix; they cannot be continuously altered.
  For further details see **File Formats**.

- **Newmix** (*Soundshaper: MchMix*) mixes files to multi-channel, using a multi-channel mixfile format (**.mmx)**:
  soundfile-name, start-time, no_chans, routing1, level1, [routing2, level2, ... to no_outchans]
  where no_chans is the number of input channels. Inputs may be mono, stereo or multi-channel. Routing maps each input channel to an output one, with a level; an input channel may be routed to more than one output channel. Note that level and routing are fixed for the duration of the mix; they cannot be continuously altered.
  For further details see **File Formats**.

Soundshaper has a dedicated MIX PAGE for both programs. It creates a basic mixfile on the Mix Page from the selected sounds. The start-time, level and pan can be set/altered for each sound, then the mix can be performed without seeing a mix list. Alternatively, a mix list can be created, saved to a mixfile, edited etc., – essential when one or more of the sounds is repeated in the mix or a mix is reproduced with different sounds.

FUNCTIONS MANIPULATING MIXFILES (.mix)

- **Attenuate** alters the overall level in a mixfile.
- **GetLevel** tests the maximum level of a mix, as defined in a mixfile.
- **Shuffle** alters the order of events in a mixfile.
- **Pan** sets or resets the starting pan positions in a mixfile, in accordance with an overall ***time pan*** textfile, but it does not enable that position to change as the mixed sound plays.
- **Spacewarp** alters pan positions in a mixfile. These are fixed positions, though: pan cannot be altered as the mixed sound plays.
- **Sync/SyncAttack** synchronises the attack, middle or end points of soundfiles in a mixfile, or a textfile containing a list of soundfiles only.
- **Timewarp** offers 16 ways of altering the timings of events in a mixfile or the gaps between start times.

Further ancillary functions help in creating mixfiles. (In some cases, simple text editing of the file may be quicker!)

- **Dummy** creates a basic mixfile (**.mix**) ready for further editing of levels and pan position.
- **Multimix** likewise creates a basic mixfile (**.mmx**) for multi-channel mixing with **Newmix**.
- **Atstep** and **OnGrid** create a basic mixfile with regular start times (Atstep) or following a list of times (Ongrid).
- **AddtoMix** adds further soundfiles to the mixfile.
- **Model** replaces soundfiles in a mixfile.
- **Test** tests a mixfile, checking both that all the soundfiles listed in it are still valid and that the syntax of the mixfile is correct.

4.6 SOUNDFILE UTILITIES
(CDP groups: MCTOOLS, HOUSEKEEP, MODIFY)

RECORDING and PLAYBACK

- **RecSF** records a (multi-channel) soundfile from a specified device in a number of formats (WAV,AIFF,AIFC or ambisonic AMB), using 24-bit resolution by default (16-bit and 32-bit floats also supported) and supporting all the usual sample rates.

- **PAPlay** plays a (multi-channel) soundfile, with a wide range of options, including support for WAVE_EX and ambisonic files (**.amb**) and output to a wide range of speaker layouts. ASIO and DirectSound are supported in Windows, and CoreAudio and Jack in OS X.

- **PVPlay** plays (multi-channel) soundfiles and frequency analysis files (**.ana, .pvx**). The available options are virtually identical to PAPLAY, including support for ambisonic files (**.amb**) and output to a wide range of speaker layouts.

  Full details of these functions are available at Richard Dobson's **MC-TOOLS PAGE** on his website.

- **ListAudevs** lists the audio devices on your system, with the same details as listed under PVPlay, PAPlay and RecSF.

OTHER SOUNDFILE UTILITIES

- **CopySFX** is CDP's "Swiss army knife" for copying/converting file-types. CDP supports only **.wav** and **.aif** file formats (not **.mp3**, **.flac** etc.), but has full support for file types within the two formats (**.aifc** and **.afc** are also supported). The default in all CDP processing is that the outfile has the format of the infile. **CopySFX** supports a wide range of input and output types, with special support for multi-channel **WAVE_EX** and Ambisonic (**.amb**) files.

- **Respec** changes the header of the soundfile:
  – **Resample** the sound at a new sample rate.
  – **Integer-Float**: converts between Integer ("shorts") and Floating-point soundfile formats ("floats").
  – **Rewrite Header**: the information in the soundfile header is changed but the actual sample data is left untouched. Directly changing the sample rate will change the speed, for example.
- **CHXFormat** changes the soundfile header of a multi-channel WAVE_EX file to switch the GUID type between plain WAVE_EX and Ambisonic (**.amb**), or change the speaker-position value to specify a particular speaker layout, as listed under PAPLAY/PVPLAY (0=generic WAVE_EX). Unusually for CDP, this is a ***destructive*** function, which should be used with care: copy the file first.
- **Degrade** (***modify radical 4***) reduces the sample rate, possibly lowering the pitch, and/or lowers the bit-resolution, which reduces the precision of the data, making the material sound coarser. For example, it can reduce speech to not much more than the rhythm and pitch contour, especially in conjunction with **Blur**.

- **DCRemove** (***housekeep extract 4***) shifts (offsets) the entire signal to eliminate unwanted DC component
- **Deglitch** attempts to remove glitches in a soundfile.
- **Endclicks** removes clicks at the start or end of a soundfile.

NEXT SECTION: SYNTH / INFO / DATA PROCESSES

Last Updated 30 Apr 2023 for CDP8