



## Release 7 – The Software: New Additions

### Added to the COMBINE Set

SPECTWIN – **Combine the formant and/or total spectral envelopes of two spectra** (Modes: 4)

- 1 - Formant envelope of *inanalfile1* with the formant envelope of *inanalfile2*
- 2 - Formant envelope of *inanalfile1* with the total envelope of *inanalfile2*
- 3 - Total envelope of *inanalfile1* with the formant envelope of *inanalfile2*
- 4 - Total envelope of *inanalfile1* with the total envelope of *inanalfile2*

### Added to the ENVEL(OPE) Set

(Also note TOPANTAIL2 - Gated sound extraction with trim and backtracking – documented in Release 7)

TREMOLO – **Apply width-controlled tremolo to a soundfile** (Modes: 2)

- 1 - Interpolate linearly between frequencies in any frequency breaktable. This is the Default.
- 2 - Interpolate logarithmically (like pitch). Take care with any zero frequencies.

### Added to the EXTEND Set

CERACU – **Repeat source sound in several cycles that synchronise after specified counts** (Modes: 0)

ECHOES – **Repeat a sound with timing and level adjustments between repeats** (Modes: 0)

ITERLINE – **Iterate an input sound, following a transposition line** (Modes: 2)

- 1 - Interpolate between transpositions (glissandi)
- 2 - Step between transpositions (discrete pitch changes)

ITERLINEF – **Iterate an input sound set, following a transposition line** (Modes: 2)

- 1 - Interpolate between transpositions (glissandi)
- 2 - Step between transpositions (discrete pitch changes)

MADRID – **Spatially syncopate repetitions of source soundfile(s) in spatially-separated repetition streams** (Modes: 2)

- 1 - Random output file order
- 2 - Use *segfile* to determine the order of output files

SHIFTER – **Generate simultaneous repetition cycles, shifting focus from one to another** (Modes: 2)

- 1 - Use the same input sound for all cycles
- 2 - The number of input files must equal the number of cycles. The program assigns the input files, in order, to the cycles, in order.

SHRINK – **Repeat a sound, shortening it on each repetition** (Modes: 6)

- 1 - Shrink from the end
- 2 - Shrink around the midpoint
- 3 - Shrink from the start
- 4 - Shrink around a specified time
- 5 - Shrink around found peaks and output each segment as a separate soundfile, also creating a mixfile with which to assemble them
- 6 - Shrink around specified peaks and output each segment as a separate soundfile, also creating a mixfile with which to assemble them

### Added to the FILTER Set

FILTRAGE – **Generate randomised VARIBANK filterbank files** (Modes: 2)

- 1 - Generate a fixed filter
- 2 - Generate a time-varying filter

### Added to the FOCUS Set

**SUPERACCU – Sustain each spectral band until louder data appears in that band** (Modes: 4)

- 1 - Operates like FOCUS ACCU
- 2 - Forces the (start of) resonances to the tempered scale
- 3 - The frequencies are specified in a 'tuning' file (a 'harmonic set')
- 4 - The frequencies and their octaves are specified in a 'tuning' file (a harmonic field)

### Added to the GRAIN Set

**NEWTEX – Generate a texture of grains made from a source sound or sounds** (Modes: 3)

- 1 - The transpositions of *insndfile* are spread over  $N$  octaves and spatially, and fade in and out randomly
- 2 - *insndfile* is read at its original rate (i.e., no transpositions), spread spatially, and fades in and out randomly
- 3 - *insndfile* is read as 'drunken walks', spread spatially, and fades in and out randomly

### Added to the HOUSEKEEP Set

**CHANPHASE – Invert phase of one channel of an input sound** (Modes: 0)

### Added to the MODIFY Set

**NEWDELAY – Delay with pitch-defined output sound** (Modes: 0)

### Added to the Morph Set

**NEWMORPH NEWMORPH – Morph between dissimilar spectra** (Modes: 7)

- 1 - Interpolate linearly ( $exponent = 1$ ) between the average peak channels or over a curve of increasing ( $exponent > 1$ ) or decreasing ( $exp < 1$ ) slope, simultaneously moving spectral peaks, and interpolating all remaining channels.
- 2 - Interpolate cosinusoidally ( $exponent = 1$ ) between the average peak channels or over a warped cosinusoidal spline ( $exponent$  not equal to 1), simultaneously moving spectral peaks, and interpolating all remaining channels.
- 3 - As mode 1, using channel-by-channel calculation of peaks.
- 4 - As mode 2, using channel-by-channel calculation of peaks.
- 5-6 - Sound 1 is (gradually) tuned to the (averaged) harmonic field sound 2.
- 7 - Sound 1 is morphed towards sound2 in *outcnt* steps, each step a new output file.

**NEWMORPH NEWMORPH2 – Textfile of prominent peaks OR morph betw dissimilar spectra** (Modes: 3)

- 1 - Output a textfile of the most prominent peaks in order of prominence
- 2 - The sound is (gradually) tuned to the harmonic field specified in the textfile output of Mode 1. The textfile lists goal-peak frequencies, most prominent first.
- 3 - Ditto, but the interpolation is timewise-cosinusoidal.

### Added to the MULTI-CHANNEL Set

**FRACTURE – Disperse a mono signal into fragments spread over  $N$  channel space** (Modes: 2)

- 1 - The output is  $N$  channel dispersal in  $N$  channel space
- 2 - The output is stereo dispersal (possibly moving) in surround space

**PANORAMA – Distribute  $N$  source files in a panorama across a specified angle of a sound-surround loudspeaker array** (Modes: 2)

- 1 - Loudspeakers are assumed to be equally spaced
- 2 - Loudspeaker positions are defined in a textfile

**TANGENT GROUP** (All have 2 identical modes:)

- 1 - When the focus is at 1, the tangent path starts along a line at right angles to loudspeaker 2
- 2 - When the focus is at 1, the tangent path starts along a line formed by loudspeakers 2 and 3

---

**ONEFILE** – Play repeats of a mono soundfile along a tangent path  
**TWOFILES** – Play repeats of two synchronised mono soundfiles along a tangent path  
**TANGSEQUENCE** – Play a sequence of mono soundfiles along a tangent path  
**TANGLIST** – Play a sequence of mono soundfiles as listed in a textfile along a tangent path

**TEXMCHAN** – Create textures over a multi-channel frame (Modes: 5)

- 1 - On a given harmonic field
- 2 - On changing harmonic fields
- 3 - On a given harmonic set
- 4 - On changing harmonic sets
- 5 - None (Neutral)

**TRANSIT GROUP** (All have 5 identical modes:)

- 1 - Glancing
- 2 - Edgewise
- 3 - Crossing
- 4 - Close
- 5 - Central

**SIMPLE** – Place repetitions of a mono soundfile on a path *into* and *across* an 8-channel array

**FILTERED** – Place filtered repetitions of a mono soundfile on a path *into* and *across* an 8-channel array

**DOPPLER** – Place pitch-shifted repetitions of a mono soundfile on a path *into* and *across* an 8-channel array, suggesting a doppler shift

**DOPLFILT** – Doppler effect on a path *into* and *across* an 8-channel array with filtering, to suggest greater distance

**SEQUENCE** – Position a sequence of mono sounds (at least 3) on a path *into* and *across* an 8-channel array

**TRANLIST** – Position a sequence of mono sounds (at least 3), as listed in a textfile, on a path *into* and *across* an 8-channel array

#### Added to the PITCH Set

**TUNEVARY** – Replace spectral frequencies with the harmonics of specified pitch(es) (Modes: 0)

#### Added to the SFEDIT Set

**CANTOR** – Cut holes in a sound in the manner of a cantor set (holes within holes within holes) (Modes: 3)

- 1 - *holesize* is a percentage
- 2 - *holesize* is a (fixed) duration
- 3 - Use superimposed vibrato envelopes

**ISOLATE** – Disjunct portions of soundfile are specified by textfile or dB loudness and saved to separate files (Modes: 5)

- 1 - Create *several* output soundfiles each of which contains *one* segment of source (*cutsfile*)
- 2 - Create *several* output soundfiles each of which contains *several* segments of source (*cutsfile*)
- 3 - Create *one* output soundfile consisting of *several* disjunct segments (*dBon* & *dBoff*)
- 4 - Cut the *entire* soundfile into disjunct segments (*slicefile*)
- 5 - Cut as in Mode 4 but also *overlap* the segments slightly: separates speech syllables (*slicefile*)

**PACKET** – Isolate or generate a sound packet (Modes: 2)

- 1 - Found packet: looks for signal minima to determine the edges of the wave-packet
- 2 - Forced packet: creates a packet at a specified time

**PARTITION** – Partition a mono soundfile into disjunct files in blocks defined by groups of wavesets (Modes: 2)

- 1 - block durations are determined by number of wavesets
- 2 - block durations are specified by the user

SILEND – **Add silence to end of file** (Modes: 2)

- 1 - Specify duration of the silence
- 2 - Specify total output duration

SUBTRACT – **Subtract one file from another** (Modes: 0)

#### **Added to the SPECNU Set**

RAND – **Randomise the order of spectral windows** (Modes: 0)

SQUEEZE – **Squeeze the spectrum into a frequency range, around a specified frequency** (Modes: 0)

#### **Added to the SPECT Set**

GLISTEN – **Randomly partition the spectrum into bins and play back in order** (Modes: 0)

SELSIM – **Replaces spectral windows with the most similar, louder window(s)** (Modes: 0)

SPECGRIDS – **Partition a spectrum into parts, over a grid** (Modes: 0)

SPECSPHINX – **Impose the channel amplitudes of *inanalfile2* onto the channel frequencies of *inanalfile1*** (Modes: 2)

- 1 - Impose channel amplitudes of *inanalfile2* onto the channel frequencies of *inanalfile1*
- 2 - Multiply the spectra

#### **Added to the STRETCH Set**

(STRETCHA has been in the system, but is being documented for the first time in Release 7)

STRETCHA – Utility: **Calculates *timestretch* factor relating to beats and tempo**

- 1 - FIND NUM. BEATS AT A TEMPO | o
- 2 - FIND *timestretch*, VARIOUS IN/OUT |
- 3 - FIND *timestretch*, TEMPO CHANGES

#### **Added to the SYNTH Set**

NEWSYNTH – **Generate complex spectra from fundamental and partial balance information in one or more textfiles** (Modes: 3)

- 1 - Generates tones with any number of (possibly varying) partials
- 2 - Generates wave-packet streams with any number of (possibly varying) partials
- 3 - Multi-channel mode in which partials spread over  $N$  octaves fade in and out randomly