C3 Automation Tools
Functions Reference

Introduction

What is this
This is the reference doc for all CAT functions. Read only if you know a bit about programming (even simple Web programming) and want/need to know more about functions work...

What does *function(n,n)* in the docs mean?
This is a complete documentation of the scripting project: those bits are how the actual functions work in the scripts. They're there to keep track of how functions work and are meant for people comfortable with programming.

Functions

Project/Misc.

Beat track functions

create_beattrack(*halve*,*selected*)
0,0 - create from scratch; 1,0 - create halved from scratch; 1,1 - halve selected measures (derived from notes)

Data functions

filter_notes()
No parameters.

fix_events()
No parameters.

All Instruments (GBDK)

remove_notes(*what,level,instrument,how,same,bend,selected*):
what: w/h/q/e; whole, half, quarter, eighth grid
level: x/h/m/e; expert, hard, medium, easy
instrument: "PART DRUMS"/etc. (leave "" to apply to currently selected track
how: 0-30; ticks tolerance to keep notes that aren't perfectly quantized. 30 ticks is 1/64th
same: 0/1; not keep or keep same consecutive notes even if they are 1 grid level faster than 'what', provided the next on grid note is also the same. In example, instead of reducing GGGG to G-G- this option leaves GGGG. However, if we have GGBBRR, this option doesn't kick in and the function reduces it to G-B-R. *same* will not work on *sparse* notes.
bend 0/1; enable pitch bends detection. If a note to be kept is 1/16th or 1/32nd long and there's a different note after that a check is performed: if it's a sustain or a single note with no other note right after it, it's deemed a pitch bend. The note to be kept takes the pitch and the length of the other note+its own length.
sparse: 0/1; keep notes that are not on grid if there aren't other notes in the space of *what*. That means that if you notes on 1/8th grid spaced by ¼, with this option those notes will be kept because even though they are not on a ¼ grid there aren't neighbouring notes. By default is on. *same* will not work on *sparse* notes.
selected: 0/1; apply to all or to selected notes (whole measures only)

edit_by_mbt*(position,notes,level,instrument,selected)*
position: The "BT" of "MBT"; which position to remove notes from in each measure
notes: any number of [g/r/y/b/o]; the gem color(s) you want to remove
level: x/h/m/e; expert, hard, medium, easy
instrument: "PART DRUMS"/etc. (leave "" to apply to currently selected track)
selected: 0/1; apply to all or to selected notes (whole measures only)

create_triplets(*instrument*):
No parameters.

create_animation_markers(*instrument, expression, pause*):
expression: use ‘mellow’ or ‘intense’ to force all [play] markers to be either of those instead of play
pause: the space in ticks that triggers the [idle] event. Use 0 to use the default value, which is 3 measures calculated as ticks per quarter (480) * 4 beats * 3.

polish_notes(*instrument, grid, tolerance, selected*)
grid: it’s the grid to be used to snap notes. By default it’s 1/32, which should be perfect for any project. Go to 1/64 if you have a very dense chart.
tolerance: in ticks, if a note is off (minus or plus) by this amount or less is snapped to the nearest grid line. By default is 20, which is roughly a 1/128. Lower it only if you have an incredibly dense chart (notes nearer than 1/64 triplets)

Guitar/Bass/Keys/PK

fix_sustains(*instrument, level, fix, selected*)
fix: with 1, the function shortens sustains; with 0 it only removes stubby sustains.

Currently sustain space is hard set, here are the parameters:

40-70 BPM
Expert: 1/32nd
Hard: 1/16th
Medium: 1/8th
Easy: 1/4th

71-160 BPM
Expert: 1/16th
Hard: 1/16th
Medium: 1/8th
Easy: 1/4th

161-300 BPM
Expert: 1/8th
Hard: 1/8th
Medium: 3/16th
Easy: 1/4th

Currently, minimum sustains length (shorter sustains are removed) is hard set, here are the parameters:
40-70 BPM
1/8th

71-160 BPM

3/16th

160-300 BPM
5/16th

reduce_singlenotes(*instrument, level, selected*)
Instrument and level parameters only.

reduce_chords(*instrument, level, option, selected*)
option: this is based on difficulty:
Hard: when set to 1 uses GB and RO chords when reducing 3 note chords
Medium: when set to 1 allows for BO chords
Easy: allows for single blue and orange notes

copy_od_solo()
No parameters.

Drums

remove_kick(instrument,level, what, selected)
instrument:PART DRUMS or PART DRUMS 2x
what: use ‘a’ to remove kick from any object, use ‘s’ to remove kick from a snare object; use ‘t’ to remove kick from any tom object; use ‘p’ to remove kick from any snare or tom object
Warning: the tom/percussion flags work only if you correctly place one tom marker per drums note. If you draw a long tom note to cover a number of drums notes, the function won’t work.

single_snare(instrument,level, what, selected)
instrument:PART DRUMS or PART DRUMS 2x
what: use ‘a’ to leave a single snare when paired with any note; use ‘k’ to leave a single snare when paired with a kick; use ‘t’ toto leave a single snare when paired with any tom; use ‘c’ to leave a single snare when paired with any cymbal.
Warning: the tom flag works only if you correctly place one tom marker per drums note. If you draw a long tom note to cover a number of drums notes, the function won’t work.

simplify_roll(*instrument, level, selected*)
Instrument and level parameters only.

drums_animations(*instrument, crash, soft, flam, grid, cymbals, how*)
instrument:PART DRUMS or PART DRUMS 2x
crash: 0 to use CRASH2 as the default crash, 1 to use CRASH1 as default crash
soft: 0 for HARD default snare/crash, 1 for SOFT default snare/crash
flam: 0 is default, set to 1 if you want any snare snare couple with yellow tom to be transformed to a flam animations, both hands on the snare
grid: ‘e’ for eight is default, set to ‘s’ for sixteenth to indicate you the lowest subdivision grid is 1/32 instead of 1/16 (when you doubled the tempo and in example a disco beat is on a 1/32nd grid instead of 1/16th)

cymbals: the number of consecutive cymbals hits that trigger a two handed action. In example, setting it to 3 forces a 2 handed animation instead of a single handed animation whenever 3 or more consecutive hits on the same cymbal are detected

flip_discobeat(instrument, level, selected, mute)
mute: set to 0 to disable it, set to 1 to not touch sections that seem already flipped without prompting the user

unflip_discobeat(*instrument, level, how, selected*)
Instrument, tolerance and level parameters only.

single_pedal(level, how, selected)
Level and tolerance parameters only.

Vocals/Harmonies

Add slides lyrics (all|selected)
Add a “+” text marker to any note without lyrics. Useful for those who add lyrics directly to each note to avoid having to add + markers and for those who go over previously finished charts and find many slides to add

Autogenerate phrase markers (all|selected)
Autogenerate phrase markers. Use the @ character to precede a word that is capitalised to force a phrase start. If H1 and H2 have been created, overlapping phrase markers that can be compacted are fixed so that on both H1 and H2 start at the same time and end at the same tim.

Auto vocals OD
Automatically create overdrive phrases with a user defined frequency. Percussions phrases are ignored.

Change selected lyric events to non-pitched (all|selected)
For each lyric event, append # or ^, user defined, at the end.

Hide selected lyric events (selected)
For each selected lyric event, append $ at the end to hide the lyric (used to hide H2 or H3 when they differ; generally hiding the part matching H1 (if neither match, the ‘easier’ part, i.e. one that has been repeating for a while or an ‘ahh’/’ooh’ event, though that will have to be decided upon by the author).

Add space between tubes (all|selected)
Add space between note tubes that touch by shortening the first one by 1/64th

Remove illegal punctuation (all|selected)

Strip all commas and quotes from lyrics

Capitalize first lyric in phrases (all|selected)

Check/fix capitalization (all|selected)
Check for all mid-phrase capitalized lyrics that aren't "I", "God", etc., and uncapitalized first letters and ask the user if they're supposed to be capitalized and switch to lowercase/uppercase if not

Move phrase start and end to closest downbeat (all)
If a phrase starts or ends on anything but a quarter note or an eighth note (user defined), move the start/end of the phrase to the closest non-overlapping downbeat

Compact phrases(all)
Take H1 and H2 phrases and try to make them the same if there are no other overlapping phrases.

Fix lyrics set as text events(all|selected)
Anything that's not a Trackname event and it doesn't begin with "[" is converted to lyrics. Selected works on all selected events and not measure by measure.

Fix uneven harmony parts(all)
Check for notes beginning at the same time on harm1/2/3 and with the same lyrics attached (compensating for lower/upper case) ending with a difference of less than 1/64th (or a different value passed to function). In that case, harm2/3 are set to harm1 length.

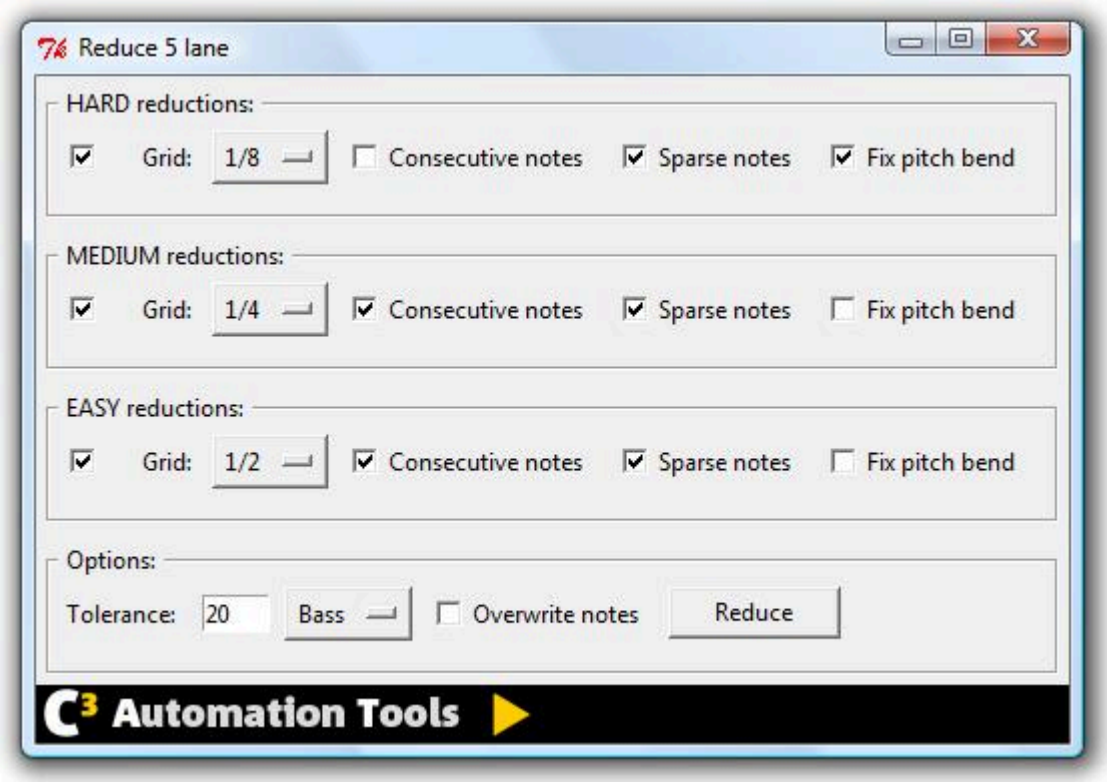
Cleanup empty phrases
If a phrase is empty on harm1 and harm3 or harm2 and harm3, remove the phrase

Supersets

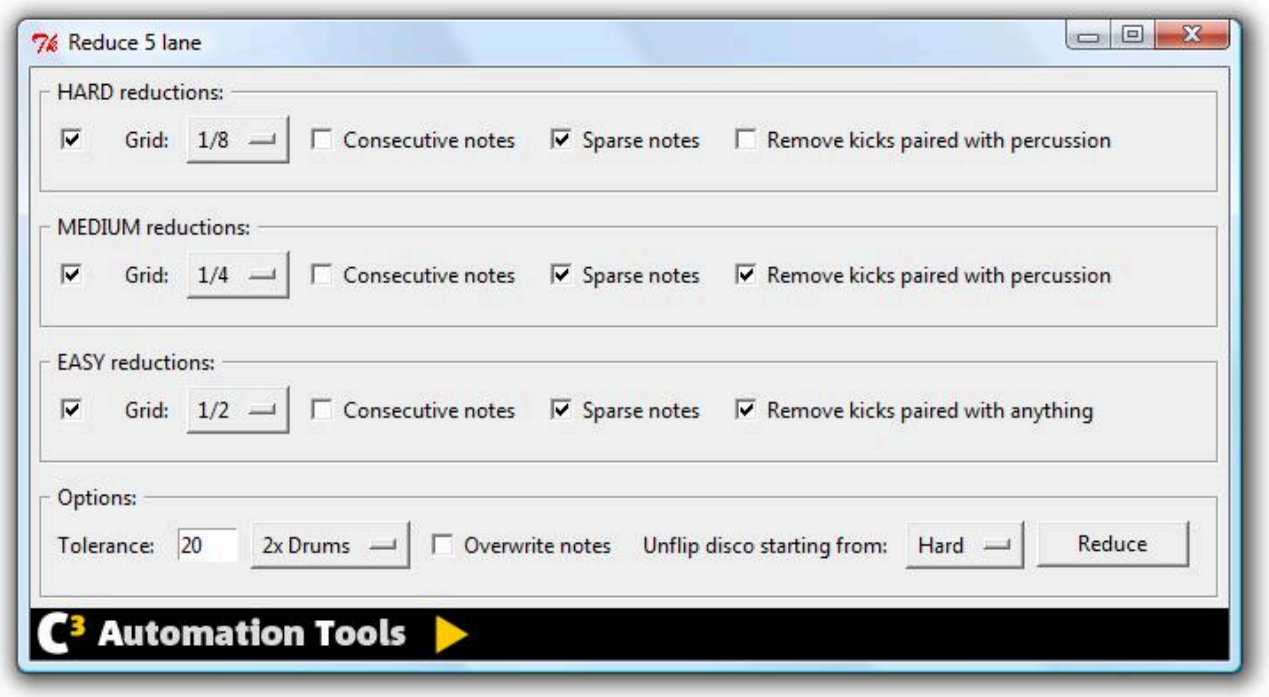
Supersets are essentially group of commands executed in a series that take advantage of functions to execute complex tasks. Supersets are controlled via GUI, however not all parameters of the single commands will be shown in the supersets window. In example, the “fix” option of fix_sustains won’t be shown in the Automatic reductions superset because sustain length on each level is dictated by RBN rules and therefore will be applied automatically.

Automatic reductions

The automatic reductions superset works on all level. You can disable each leve you don’t need/want reductions for. For each level you can elect to keep consecutive notes, sparse notes and to fix pitch bend. For each level you can select the grid setting. You can set a global tolerance level and you can choose not to be prompted when notes are found before overwriting them. You can also choose whether to simplify chords and notes and which options to apply.



For drums, you can choose to remove kicks when paired with percussion. You can also choose at which level unflipping disco beats, if present.



In addition to the selectable options, the superset automatically **simplifies rolls/washes/trills/tremolos starting from Hard** and **fixes sustains length** based on difficulty level, instrument and BPM.

Tip: Automatic reductions work best on charts with not a lot of offbeat notes and such. For that reason, for optimal results, it is recommended that for charts with weird rhythms you author Hard and let the automatic reductions superset take care of Medium and Easy.

Charts clean up

Vocals fixup

Animations creation
This superset creates the drums animation note, populates the BEAT track and adds animations markers to all instruments.

Tips

Anything that affects a whole track, try to apply before reviewing, so you can catch anything that didn’t work as expected.

Possible issues

Weird track names
If you import MIDI files with special characters in the track names, the commands may not work