

this Renoise tool observes track names and automatically assigns colors to the tracks by predefined filter rules. This is best explained by example:

E.g. filter is: snare -> color blue

Now, if any track name contains the string "snare". E.g. snare1, softsnare, hardsnare,... it's color will automatically be change to blue. Beside this very simple filter, it's also possible to define multiple filters using more complex filters that may include wildcards and regular expressions.

#### == INSTALLATION

Just install this tool as every other Renoise tool. E.g. via drag and drop: drag the .rxnx package and drop it into Renoise.

#### == ADDING FILTERS BY COMMAND

First, you may wonder how to add filters at all. Actually this is very easy, but also a bit "unusual": to add a filter you have to click at a tracks name and enter e.g. the string "add:snare". Thus, a new filter "snare" and the tracks color setting are added to AutoColor's filter list. "add:" is interpreted as an AutoColors-internal command. There are more commands described below.

#### == HOW TO SELECT A COLOR

So far there exists no dialog or something for selecting a color for a specific filter. Instead you have to set the color RGB/HSV value and the color blend value in any track. Afterwards, you have to enter e.g. an "add:<filter>" or "upd:<filter>" command to assign the tracks color properties to the filter.

#### == VIEW AUTOCOLORS FILTER LIST

To check which filters exist, you can open the AutoColors Filter List Dialog:

Renoise View Menu -> AutoColors Filters

This dialog displays all defined filters together with their color values. The filters are grouped by color/color blend values.

#### == COMMANDS

AutoColors is completely driven by commands, which are:

add:<regex>[,<regex>,...]	add new filter(s)
rem:<regex>[,<regex>,...]	remove filter(s)
upd:<regex>[,<regex>,...]	update color of single filter(s)
upg:<regex>	update color of a filter's group
lst:	show/hide this dialog
reset:	reset = remove all filters
save:<name>	save all filters into xml file (name is no path !)
load:<name>	load all filters from xml file (name is no path !)

HINT: whenever the filter list is modified, AutoColors saves the changes automatically into a "config.xml" file. If you accidentally destroy this file, you can restore it by copying "config.xml.back" into "config.xml".

This has to be done manually.

## == LUA REGEX OVERVIEW (NOT COMPLETE)

regex stands for LUA regular expression and can be a simple text string like "snare" or a more complex matching pattern like `^drum[123]$` If you're not familiar with regular expressions, have a look at <http://lua-users.org/wiki/PatternsTutorial>, or google for "regular expressions".

IMPORTANT: all string comparisons are done case-insensitive. Means: Snare and SNARE or SnAre are the same.

### SPECIAL CHARACTERS

All characters: `^$()%.[\]*+~?)` are regex "magic" characters, with different meanings, e.g.:

- `^` = "starts with", e.g. filter `^snare`, e.g. matches `snare1`, `snare2`, but not `softsnare` etc.
- `$` = "ends with", e.g. filter `snare$`, e.g. matches `softsnare`, but not `snare1`
- `.` = "all character", e.g. filter `snare.`, e.g. matches `snare1`, `snaredrum`, but not `softsnare`
- `?` = "0..1 occurrence of the prepending character", e.g. filter `^1?snare$`, e.g. matches `"snare"`, `"1snare"` but not `2snare` or `popsnare`
- `+` = "1..n repetitions of the prepending element", e.g. filter `snare.+`, e.g. matches `"snare1"`, `"snaredrum"` but not `"softsnare"`  
Hint: `"snare+"` has the same effect as `"snare"`
- `*` = "0..n repetitions of the prepending element", e.g. filter `my*snare`, E.g. matches `"mysnare"`, `"mysoftsnare"` but not `"softsnare"`  
Hint: `"snare*"` or `"*snare*"` has the same effect as `"snare"`

### CHARACTER CLASSES:

- `%a` = "all letters." E.g. filter `snare%a`, e.g. matches `"snareA"`, not `"snare1"`
- `%d` = "all digits." E.g. filter `snare%d`, e.g. matches `"snare1"`, not `"snareA"`  
E.g. filter `snare%d%d%d`, e.g. matches `"snare001"`, but not `"snare1"`
- `%p` = "all punctuation characters."
- `%s` = "all space characters." E.g. filter `snare%s`, e.g. matches `"snare drum"`, but not `"snaredrum"`
- `%w` = "all alphanumeric characters." E.g. filter `snare%w`, e.g. matches `"snareA"`, `"snare1"`, `"snare1B"` but not `"snare#"`
- `%x` = "all hexadecimal digits." E.g. filter `snare%x`, e.g. matches `"snareF"` but not `"snareG"`

### SETS:

[<elements>] = union of all elements/character in set. E.g. filter `snare[abc123]`, e.g. matches `"snareA"`, `"snare3"` but not `"snareD"`, nor `"snare4"`  
E.g. filter is `snare[_%w]+$`, e.g. matches `"snare_123"`, `"snare_XY_123"` but not `"snare#123"`

[0-9] = all digits from 0..9. E.g. filter `snare[1-8]`, e.g. matches `"snare8"`

but not "snare9"  
[a-z] = all characters from a..z. E.g. filter "snare[a-z]", e.g. matches  
"snareA" but not "snare1"

[^<elements>] = complement set. E.g. filter "snare[^abc]", e.g. matches  
snareD, but not snareA

== LICENSE

Copyright 2012 Matthias Ehrmann,

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
you may obtain a copy of the License at  
<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed  
under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR  
CONDITIONS OF ANY KIND, either express or implied. See the License for the  
specific language governing permissions and limitations under the License.

CHANGELOG:

1/12 v1.01 initial release