



AutoHotkey Kit for MuseScore

Colored Markers

June 2022

MuseScore: <https://musescore.org/en/download> Current version 3.6.2

AutoHotkey: <https://www.autohotkey.com/> Current version 1.1.33.10 (August 2021)

"The ultimate automation scripting language for Windows."

Download the Unicode version <https://www.autohotkey.com/docs/Tutorial.htm#s11>

This macro group contains the files:

Colored_Markers.pdf	this document
Colored_Markers.zip *	2 files: an mscz resp an mpal
Colored_Numbers_Notes.txt	macros for MuseScore
PixelMousing_Col_Markers.txt	tool

* For a pdf of .mscz see also the last page. The palette is Col_Numbers.mpal.

Change the .txt extension in .ahk At the top of the two .ahk files:
set the path to your location of MuseScore3.exe

When the extensions are not visible then (for Windows 10): Open Windows File Explorer, switch the ribbon to the View tab and tick the File name extensions box. Now you'll be able to change the actual extension.

Create a folder called 'AHK' in C:\.....\Documents\MuseScore3. This is your 'working directory'. It will include the two .ahk files and a few images.
The macros have been developed and tested on a 3.7 Ghz AMD machine - 2 cores - 4 logical processors. Screen resolution 1920 x 1080 at 96 DPI.

Colored Notes and Colored Numbers

The AutoHotkey command *PixelSearch* scans the Canvas looking for a specific color. If the color is found the mouse moves and selects the colored element. If not found the macro continues the search on the next or previous page dependent on the search direction.

Performance

On the test system in Page or Continuous View, at a zoomfactor of 100% it takes 9 seconds to search a score of 100 pages from start to end.
The test score has 18 staves, all visible at a staff size of 100% with sp in Page Settings set to 0.069 inch (1.750 mm). The macros use the MuseScore command Control + PageDown (resp. PageUp). So colored elements on lower staves will not be visible and therefore not found at a zoomfactor of 100%.

Differences between Colored Notes and Colored Numbers

In the test score colored numbers, bold 12 pt are found at a zoomfactor of 100%. But colored notes of head type quarter are already found at 45%. Or alternatively, in the Template 'Symphony Orchestra' with sp= 0.037 inch (0.950 mm) the notes are easily found but not the numbers.

In the test score you need a colored number to be bold 18 pt before it will be found at a zoomfactor of 60%.

Colored numbers have an advantage of their own: you can manage them all at once e.g. setting them 'invisible' by rightclicking a colored number -> select More -> same subtype. The subtype is the user text style 'Colored Numbers'. For how to create the numbers and set this user style see the .mscz file.

A combination of colored notes and colored numbers in the same score can be interesting as well. It doubles the number of search categories and enables searches also at small zoomfactors.

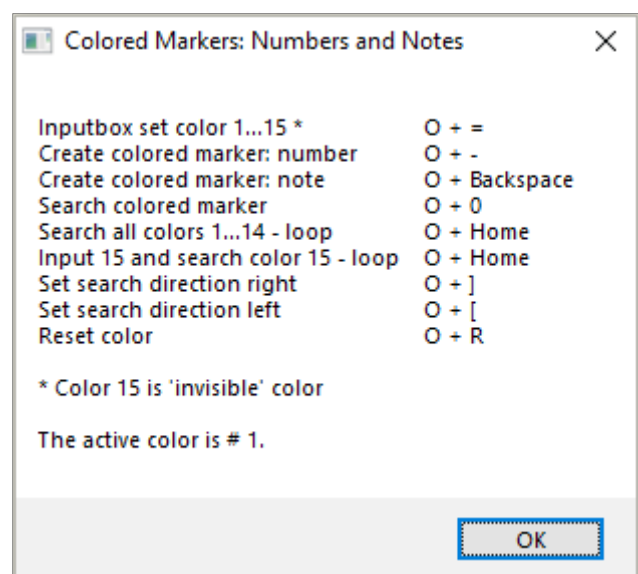
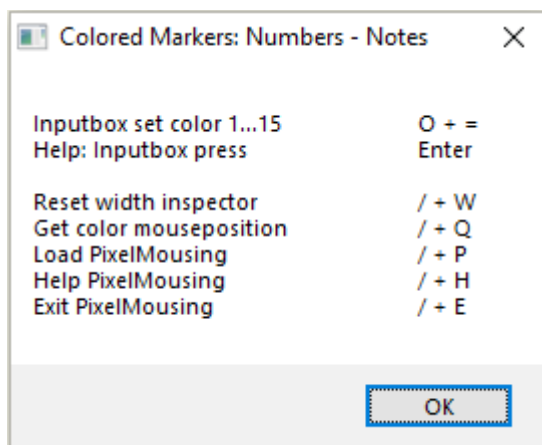
DIY items to get the macros working

Each system is different, so DIY items are in a sense part of the Kit. You have to create a few *images* and you have to determine the coordinates of some *search areas*, some *hotspots* within the Inspector and the size and position of the *Inputbox*. There is also one *color number* to be found.

You enter these numbers in the auto-execute section almost at the top of the .ahk file.

Tools

The first tool is *Window Spy* which comes with the installation of AutoHotkey. Next there is *PixelMousing* which is part of the kit. Also included are hotkeys for finding the (numbers of) colors and for restoring an important aspect of the *Defined State*.



The Defined State

The macros will only work if they are executed in the same screen layout as in which the coordinates of hotspots and search areas were originally determined. E.g. to ensure that a click on an Inspector hotspot always hits the right target.

The Defined State reflects the preferences of your workflow. It is your ideal fixed layout of the screen. Do you prefer *full screen* or *maximized* screen? And which *toolbars* must be present? All this influences the **height** of the *docked* Inspector. But also the position and thus the surface coordinates of the *Statusbar*.

In choosing the **width** of the Inspector the main consideration is finding the optimal balance between maximum Canvas real estate and accessibility of important Inspector controls.

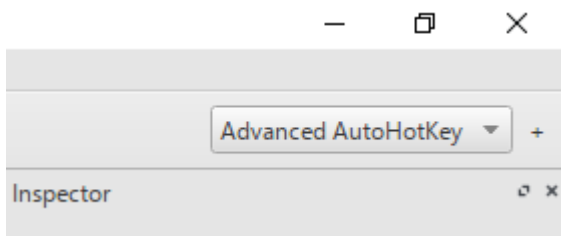
And when the width of the Inspector inadvertently changes it is important to be able simply to restore its width.

A simple Defined State check

Whichever defined state you may have chosen in each defined state this **Inspector** image *will always occupy the same surface*.

Create this image and put it in your AHK working directory.

Inspector **IM_01_Inspector_Inspector.png**



The image is part of the Inspector side panel. Shown is a Defined State consisting of a maximized screen and the toolbar *Note Input* combined with *Workspaces*. NB: The *Advanced AutoHotKey* workspace lets you enter all palette items using an Inputbox.

Almost all macros execute this line at the beginning:

```
ImageSearch, , , IM_01_X1, IM_01_Y1, IM_01_X2, IM_01_Y2, *40 IM_01_Inspector_Inspector.png
```

"Search for the image **IM_01_Inspector_Inspector** (**Inspector**) within the rectangular surface determined by the upper-left corner with the coordinates **IM_01_X1** and **IM_01_Y1** and the lower-right corner with **IM_01_X2** and **IM_01_Y2** and allow 40 shades of color variation in the search. (* 40)."

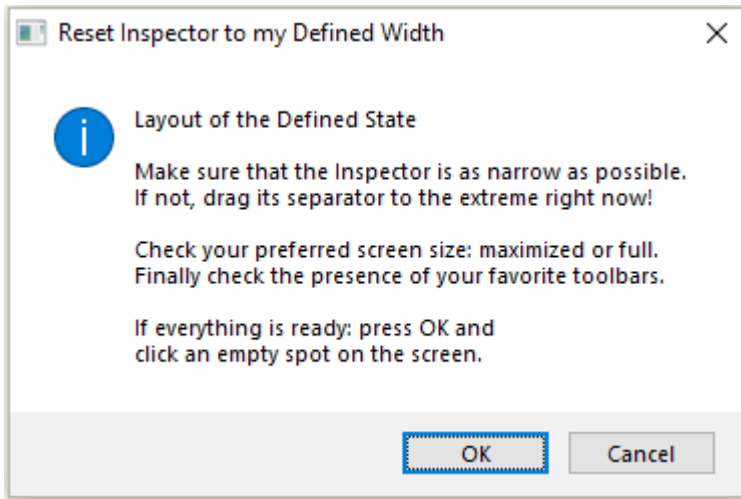
We inform AutoHotkey about the coordinates thus: the values are valid for the test screen

```
IM_01_X1 := 1639
IM_01_Y1 := 23
IM_01_X2 := 1697
IM_01_Y2 := 43
```

These coordinates - values - belong in *the auto-execute section at the top of the .ahk file*. The data will be read automatically after launching it. Make the Search Area a tiny bit bigger than the image. Use PixelMousing.

Resetting the width of the Inspector

After pressing **/ + W** this message is displayed:



Determine the minimal width of the Inspector. This is a fixed number. Using PixelMousing you move the mouse from the extreme right to the point where the Inspector separator becomes active. Make a note of the X-coordinate. Enable drag at the pixel where the mouse cursor changes shape. Move the separator to the desired Inspector width. Disable drag. Make a note of the X-coordinate.

```
Min_Insp_Width := 1852          ; my found minimal Inspector width
Def_Insp_Width  := 1640          ; my defined inspector width
```

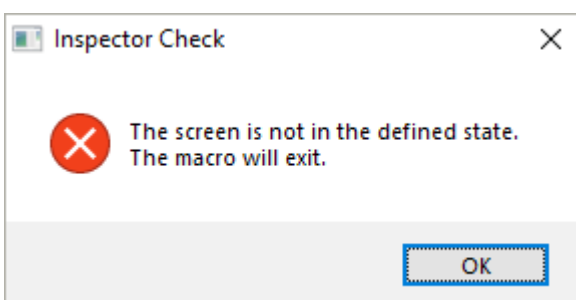
Visual verification

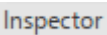
There is a calming appeal in having an on-screen indication that the Inspector has its defined width.



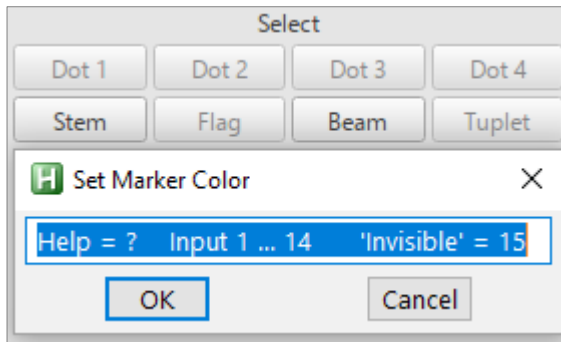
This indication is a tooltip. Its left border is aligned with the Inspector. Its position on the screen is described by the coordinates of its upper-left corner. So its X-coordinate is **Def_Insp_Width**. The picture shows a Defined State including fullscreen. Here its Y-coordinate is 0, as high as possible. Of course you can change this in the auto-execute section.

Warning message



You get this message if the image  is not found.

O + = Set color - Inputbox



The Inputbox within the Inspector.
Window Spy shows its position and size.

Active Window Position:					
	x: 1635	y: 806	w: 289	h: 100	
Client:	x: 0	y: 0	w: 273	h: 61	

In the auto-execute section we enter the numbers. For the test screen:

IB_X := 1635	Position	W_IB := 289	Size
IB_Y := 806		H_IB := 100	

The text style Colored Numbers - Defining the colors

For the style settings see Colored_Markers.mscz.
You can use it as a template for the creation of the colored numbers.
Or try loading Col_Numbers.mpal.

Hotspots

IN_012_X := 1862 IN_012_Y := 86 hotspot color picker rectangle



Notice the position of the red arrow point. If you select a text the black rectangle is displayed more to the right compared with a note selected. The chosen point is in the overlap area. The difference is caused by the S button of *Set as style*.

The blue arrow point: reset color of elements *without* a Style button.

IN_013_X := 1905 IN_013_Y := 86

Images and their search area

Inspector	IM_01_Inspector_Inspector.png	search area: described above status bar wide - see below status bar narrow - see below in lower-right corner of screen
Voice: 1;	Statusbar_Voice1.png	
Text:	Statusbar_Text.png	
Text edit mode	Screen_Text_edit_mode.png	

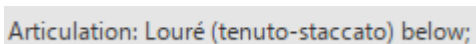
After a click on a colored number it happens frequently that we enter text edit mode. We need an image to automatically exit this mode. The search area must be wide enough to allow for long measure numbers

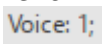
TxtEdit_X1 := 1769	TxtEdit_X2 := 1879
TxtEdit_Y1 := 1060	TxtEdit_Y2 := 1079

Statusbar Search Areas

In order to optimize speed of macro execution whenever possible we use two different search areas. They share the upper-left corner but have a different lower-right corner. *Upper-left corner*: set the height so that the surface can accomodate **2** stacked statusbars. The reason is that in some situations the statusbar can get an extra line.

For the *Lower-right corner* there are two variants, *Narrow* and *Wide*.

Narrow.  Create this articulation and take **SB_X2** a few pixels bigger than the width of this image.

Wide. The image  is sometimes part of a longer - wider - message. To get the image recognized in all cases create a note as seen in this image:



and determine **SB_Wide_X2** as lying a bit to the right of 

The important thing here is to take the 'duration percentages' of the **Piano Roll Editor** *OnTime* and *Len* into account.

To get to the **PRE** select a measure -> right click -> Piano Roll Editor.

In the auto-exec part of the file we assign the values to the variables.

SB_X1 := 0

SB_Y1 := 1038

SB_X2 := 237

SB_Wide_X2 := 710

SB_Y2 := 1079

An Imagesearch of the Statusbar takes 30 ms on the testscreen.

The eye catcher

This image appears for a short moment near the found element after a search. In AutoHotkey its appearance is controlled by the **SplashImage** command. It comes in handy especially in densely populated scores.



MuseScore_logo_round.jpg

NB: the image must be in jpg format.

The Canvas Search Area

CSA_X1 := 120 ; canvas upper-left corner

CSA_Y1 := 55 ; of real estate

CSA_X2 := 1633 ; canvas lower-right corner

CSA_Y2 := 1052 ; of real estate

Checking the view mode

In a search the next (or previous) page is selected with Control + PageDown (resp. Control + PageUp). MuseScore reacts differently depending on its view mode. After Control + PageDown in Continuous View the first displayed measure is the (pen)ultimate measure shown on the previous screen. In Page View however it is the next page. These canvas shifts are independent on the zoomfactor.

When your screen is big enough to display two pages at a zoomfactor of 100% you can double the search speed in Page View. [See (*change*) in ~o & 0::]

AutoHotkey can determine the view mode by searching for the background color which is only present in Page View. (*)

In MuseScore -> Preferences -> Canvas -> Background we have the options *Color* and *Wallpaper*.

If we prefer *Color* and we click on the colored ribbon the color picker opens showing the color id. number. For the test screen it is #ddd6d8.

In the auto execute section we enter

ColorBackground := 0xdddd6d8

(In AutoHotkey we write 0x instead of #.)

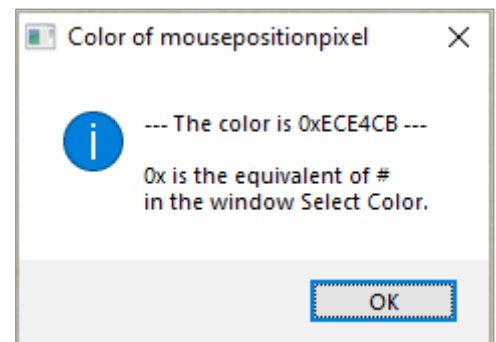
If we prefer *Wallpaper* we find the color

by positioning the mouse and pressing / + Q

For e.g wallpaper/paper3.png we get:

ColorBackground := 0xece4cb

(*) At a high zoomfactor the background color may not always be present. But this is not relevant for search purposes.



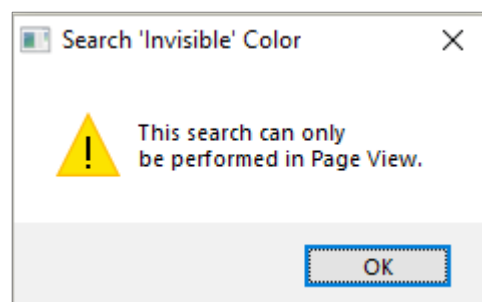
The 'invisible' color #15

This is the color of elements set *invisible*. You find its color id. in Preferences -> Advanced -> ui/score/layoutBreakColor



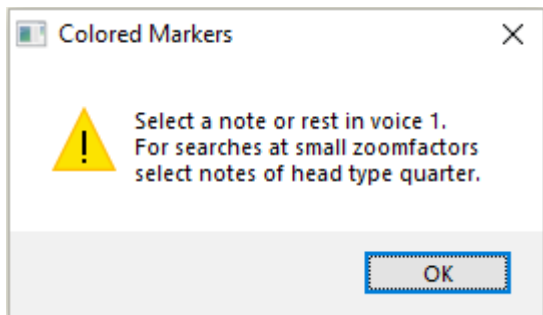
A search for this color only makes sense in Page View. The color is abundantly present in Continuous View.

Of course for visible markers colored with #15 the color is just another search color. But consider the number attached to colored notes. This stafftext will often be the beginning of e.g. an annotation or a remark which is temporarily set invisible.



Creation of colored markers

Colored Notes: search in the ahk file for `~o & bs::` to see how it works. The macro first checks the Defined State for the presence of `Inspector` and next the Statusbar for the presence of `Voice: 1;`. If `Statusbar_Voice1.png` is not found it displays this message:



If the image is found it creates a stafftext containing the number of the colored note as set by `O + =`. It moves back from the text to the note. Now a click on the color picker `IN_012` opens the window Set Color. The color id is send. The macro finishes by selecting the colored note using this id.

Colored Numbers: search in the ahk file for `~o & -::` to see how it works. In the *palette Colored Numbers* we have given a unique name to each number. The macro sends Control + F9 as MuseScore shortcut for *Palette Search*. The macro sends the text of the unique name to the search field. The macro sends Control + Alt + P as MuseScore shortcut for *Apply current palette element*. If your shortcut definitions are different change the lines marked with `(*change*?)`.

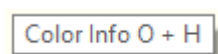
In AutoHotkey lingo Control is written as `^`, Alt as `!` and Shift as `+`

FYI: using the same technique we can enter *any* palette element by typing shortcuts in an Inputbox. This is an incredible timesaver. I have done this for all palette elements of the Advanced workspace. See <https://musescore.org/en/node/316166> especially Post 6. An update will be published on Github.

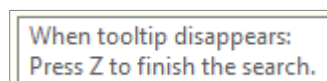
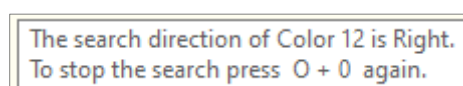
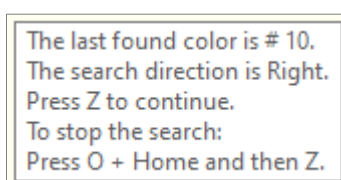
'Liberating' the prefix keys

This macro group uses `O` and `/` as prefix keys, the first key of the hotkey combinations. When these keys are occupied by MuseScore shortcuts we have to 'liberate' them by redefining the shortcut. For instance `/` as the default shortcut for *Add acciaccatura* could be redefined in **Shift + /**.

Positioning of Tooltips



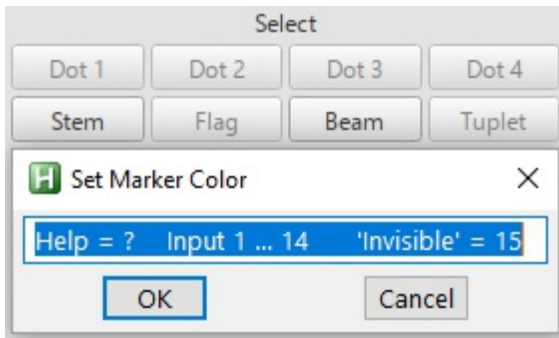
A reminder that the macro group is active. In auto-execute section.



These 3 tooltips appear near selected elements. Search for `(*change*?)` if you need to adjust their position.

Colored Markers

and how to create the Custom Palette 'Colored Numbers'



Tooltip reminder

Color Info O + H

(*) e.g name of
Colored Number 9

Palette Cell Properties

Name: CN-9-

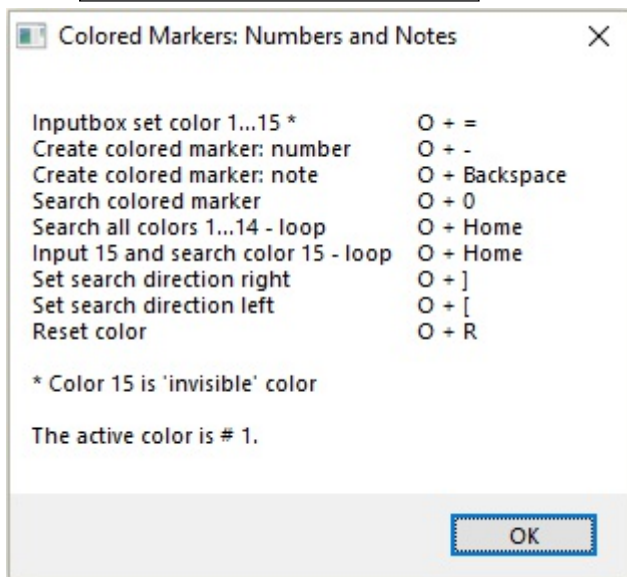


Colored Notes - copy # note color to the
corresponding number in the staff below.
So the number 1 must get the color of note 1.

The numbers of Colored Notes
are created by **O + Backspace**



Help Info Inputbox after **Enter**



Staff Text - create user style
'Colored Numbers' Edwin 12 pt **bold**,
independent on staff size.
Input the numbers. Set their style.
Color the numbers. Add them to a
(custom) palette. Edit their names. (*)
Save palette as Col_Numbers.mpal

General Info after **O + H**

