Updated plugins Color Enharmonic Pitches, Apply Named Color, and Color Picker

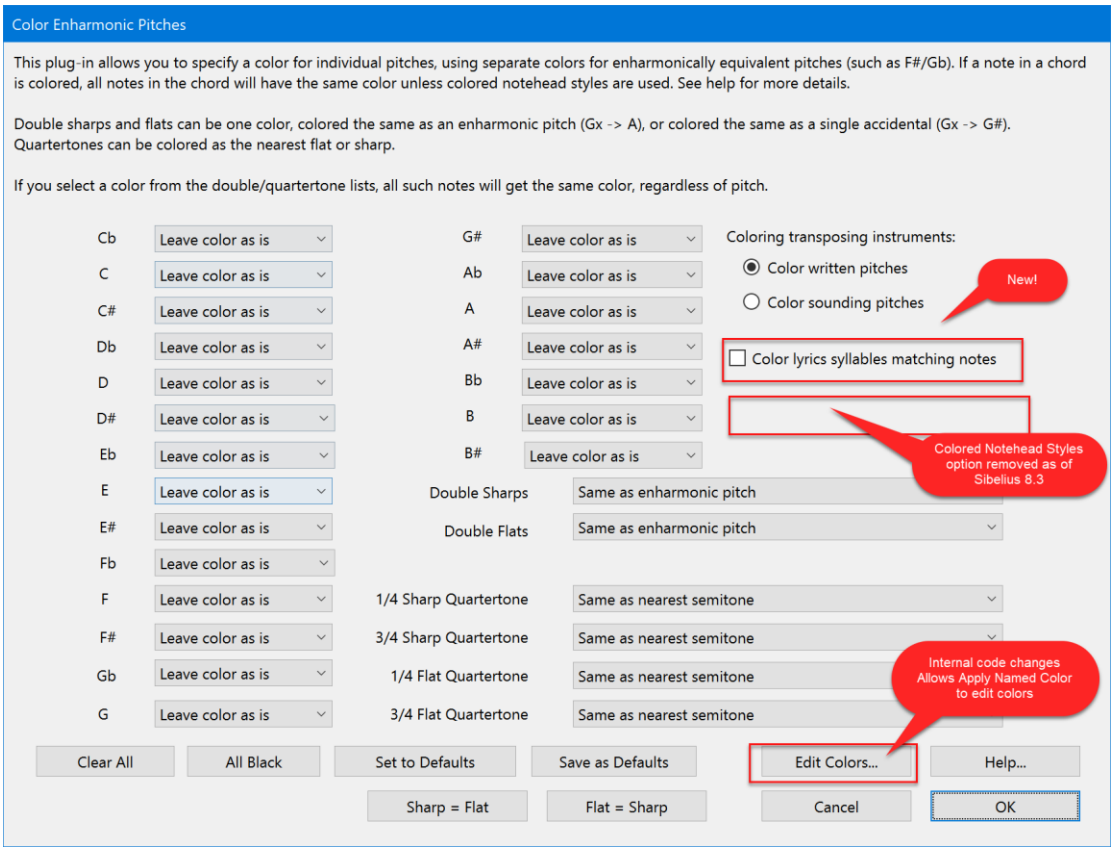
Bob Zawalich December 30, 2023

Table of Contents

These are plugins that can modify colors that are explicitly assigned to **Bar Objects**, such as notes, text, symbols, or lines, using **Home>Edit>Color** or a plugin. They will not affect **Voice** or **Out-of-range** colors, or colors assigned in **File>Preferences>Accessibility**.

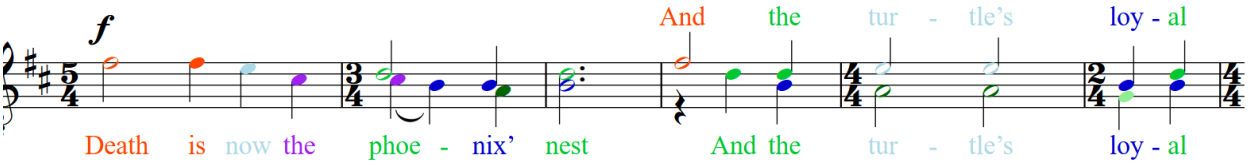
Color Enharmonic Pitches lets you assign specific colors, by name, to notes of a given pitch.

You can assign different colors to notes with enharmonic pitches, which are pitches that sound the same but are written differently, such as F# and Gb. Its dialog looks like this:



Use Help… for additional details.

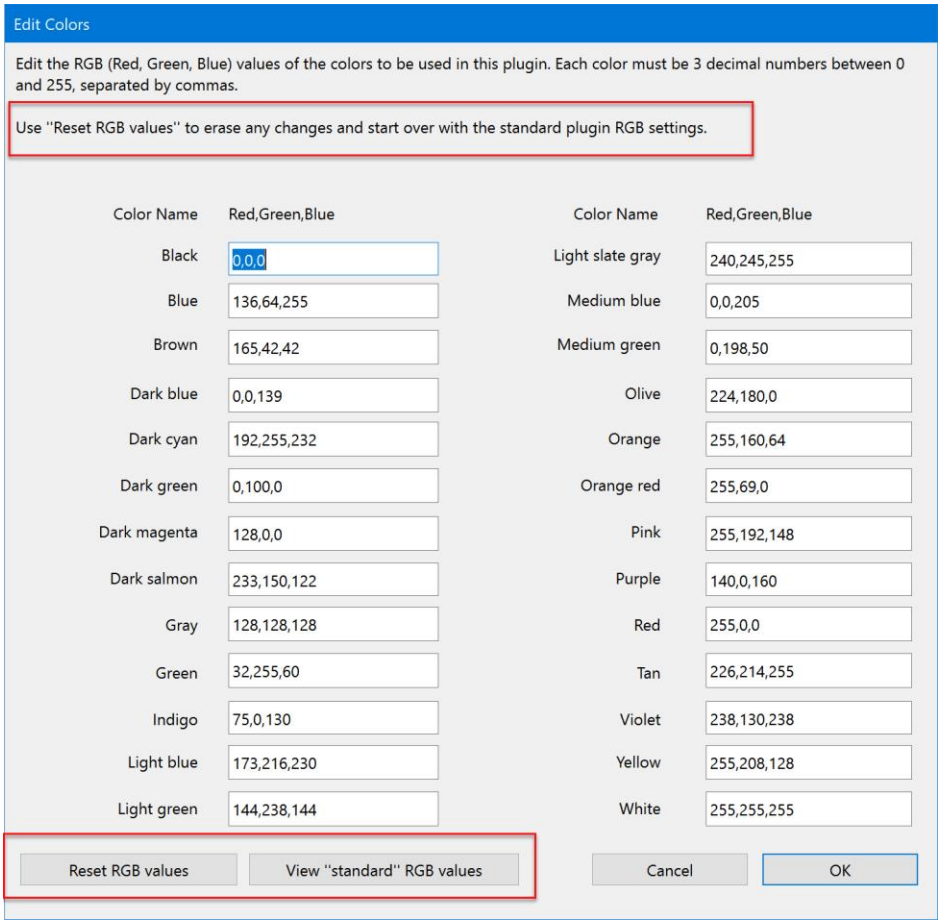
I was asked to allow **Color Enharmonic Pitches** to color **lyrics** associated with notes to be the same color as their "associated" note. Lyrics are not actually assigned to a note, so coloring is done if there is a note at the same bar and position within the bar as a lyric syllable. If there is a matching multi-note chord, the highest note in the chord is used to color the lyric syllable. It can now produce something like this:



I was planning to only use this in a custom version of the plugin, since I did not think it would be useful in general, but I was making some other changes, and thought it would be OK to leave it in. The **Color lyrics** checkbox is off by default.

I also removed an option to use **Colored Notehead Styles** as of Sibelius 8.3, since that appeared to be causing confusion. **Colored Notehead Styles** was a scheme I used before it was possible to have individually colored notes in a chord. Using the normal coloring of notes is simpler and more general, so I removed the option.

Color Enharmonic Pitches has the special ability of allowing you to specify the **RGB** (Red, Green, Blue) values of each of its named colors, so you can set up a custom **RGB to color-name map**, which associates a name with an RGB value. The mapping is done in the **Edit Colors** dialog:



The values shown above are the "standard" RGB values my color plugins have been using. The button **Reset RGB values** is a new feature to restore the standard RGB values if you end up trying to create a new **RGB to color-name map** and things go wrong. This is comparable to **Clear All** in the main dialog.

I also made some internal changes to the code in **Color Enharmonic Pitches** that allow other plugins to call it and set up their own **RGB to color-name maps**.

The current color settings in the main dialog can now be saved across Sibelius sessions, along with the default color settings and the **RGB to color-name map.** Settings are saved *only if you choose **OK** in the main dialog*. *Changes are not saved if you choose **Cancel**.*

If you are changing settings and want them to be saved, always use OK in the main dialog, even if you don't want to change the score. Use **Undo** when the plugin ends if you did not want to change the score.

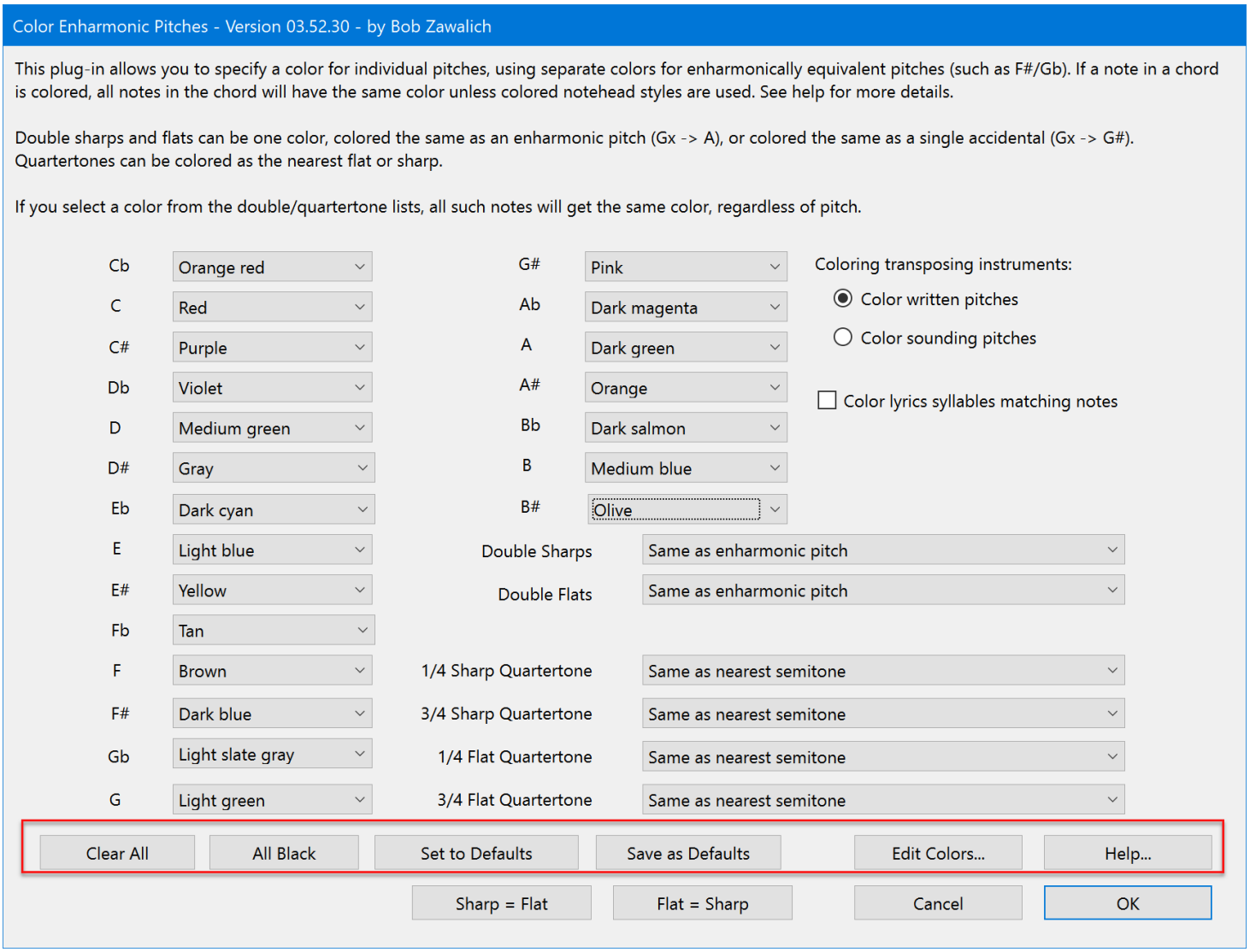
Pitch-Color mapping in Color Enharmonic Pitches (not new but not documented)

You can assign a color to any pitches (in all octaves) you choose, including using different colors for enharmonic notes which sound the same but are spelled differently, such as F# and Gb. This is the **Pitch to color-name map** (as opposed to the **RGB to color-name map** set up in **Edit Colors**).

Previously, after you set up a **Pitch to color-name map** in the main dialog its settings would be preserved only for the current Sibelius session. I have now changed the plugins so the colors chosen in the main dialog will be saved to the **Plugins Preferences** file so these colors will be restored the next time you run the plugin, even after closing and restarting Sibelius.

If you like your map, but would like to tinker with it some, you can save an extra copy of the current map by pressing **Save as Defaults**. I recommend doing this after you finish setting up your map, so you have an easy way to get back something you liked. To restore the map you saved with **Save as Defaults**, press **Set to Defaults**.

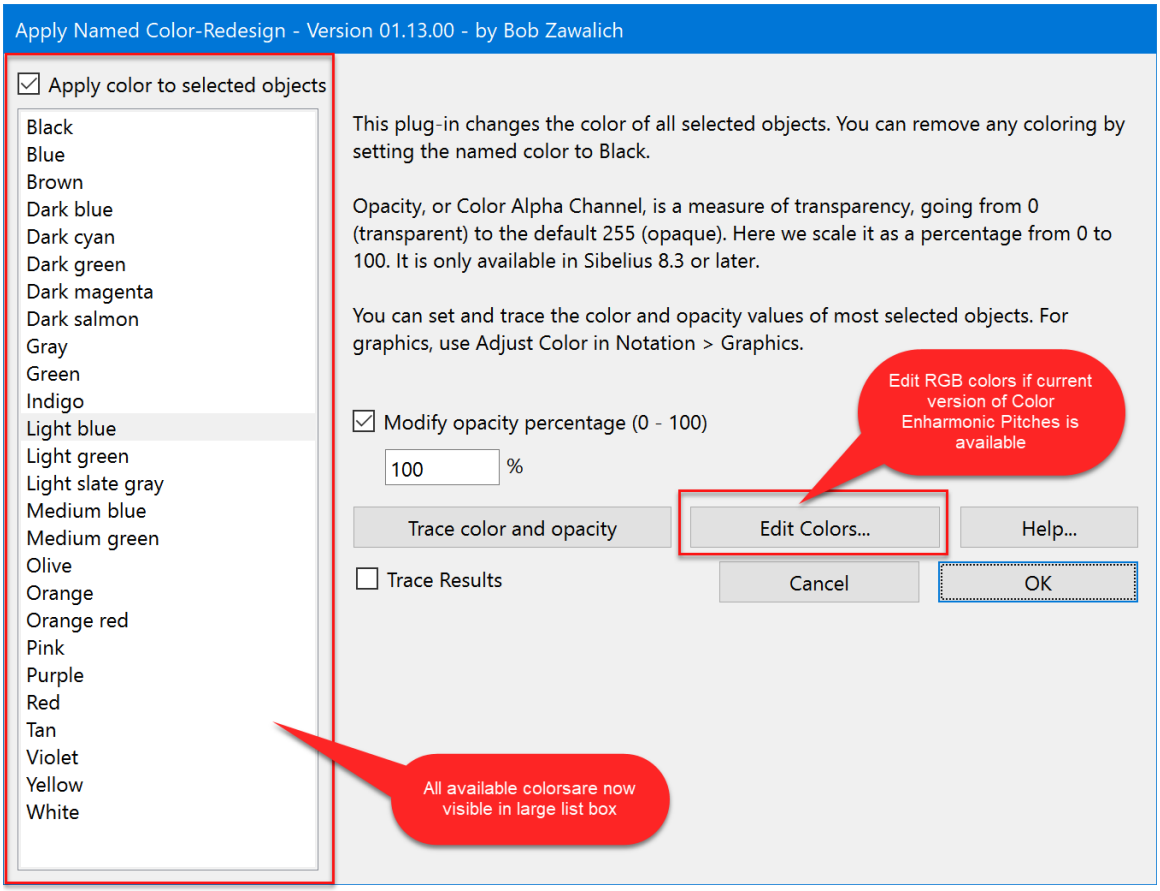
You can also set the map to make all pitches Black (**All Black**) or reset all pitches to **Leave color as is** (**Clear All**).



Apply Named Color

Apply Named Color lets you assign a color, using a name rather than an RGB value, to any selected Bar Objects. It also is the only way that a user can change the Opacity/ translucency of colored objects.

I redesigned the **Apply Named Color** dialog so that all the colors available are visible at once in a tall list box. The colors names were previously in a drop-down list box, which was more compact but not as convenient.



The other change to **Apply Named Color** is that you can now change the **RGB** color values used for each named color. This calls the plugin **Color Enharmonic Pitches** to do the RGB mapping, so the button is disabled unless the current version of **Color Enharmonic Pitches** is installed. Changing the RGB values is tedious and finicky, but some have found it to be useful.

The standard plugin color mapping is described in the **Apply Named Color Help** dialog, and in the **Edit Colors** dialog:

Apply Named Color Help

These are the *default* RGB (Red, Green, Blue) values and combined decimal color values for the named colors used in this plugin (values found at http://www.tayloredmktg.com/rgb/#PI and http://www.webdesigns2000.com/WebPage1/colortest140.htm) (corresponding to the values in Sibelius's Edit > Color command).

These colors correspond to the colors used in the pitch spectrum, plus black and gray and some additional colors.

RGB values may be edited using Edit Colors, which calls code in the plugin Color Enharmonic Pitches. The edited colors in each plugin are independent. Changing colors in one plugin does not affect the other.

Use Edit Colors to see the current RGB values.

RGB Value	Decimal	Name
R000 G000 B000	00000000	Black
R000 G000 B255	00000255	Blue
R165 G042 B042	10824234	Brown
R000 G000 B139	00000139	Dark blue
R000 G139 B139	00035723	Dark cyan
R000 G100 B000	00025600	Dark green
R139 G000 B139	09109643	Dark magenta
R233 G150 B122	15308410	Dark salmon
R128 G128 B128	08421504	Gray
R000 G128 B000	00032768	Green
R075 G000 B130	04915330	Indigo
R173 G216 B230	11393254	Light blue
R144 G238 B144	09498256	Light green
R119 G136 B153	07833753	Light slate gray
R000 G000 B205	00000205	Medium blue
R000 G198 B050	00050738	Medium green
R128 G128 B000	08421376	Olive
R255 G165 B000	16753920	Orange
R255 G069 B000	16729344	Orange red
R255 G192 B203	16761035	Pink
R160 G032 B240	10494192	Purple
R255 G000 B000	16711680	Red
R210 G180 B140	13808780	Tan
R238 G130 B238	15631086	Violet
R255 G255 B000	16776960	Yellow

OK

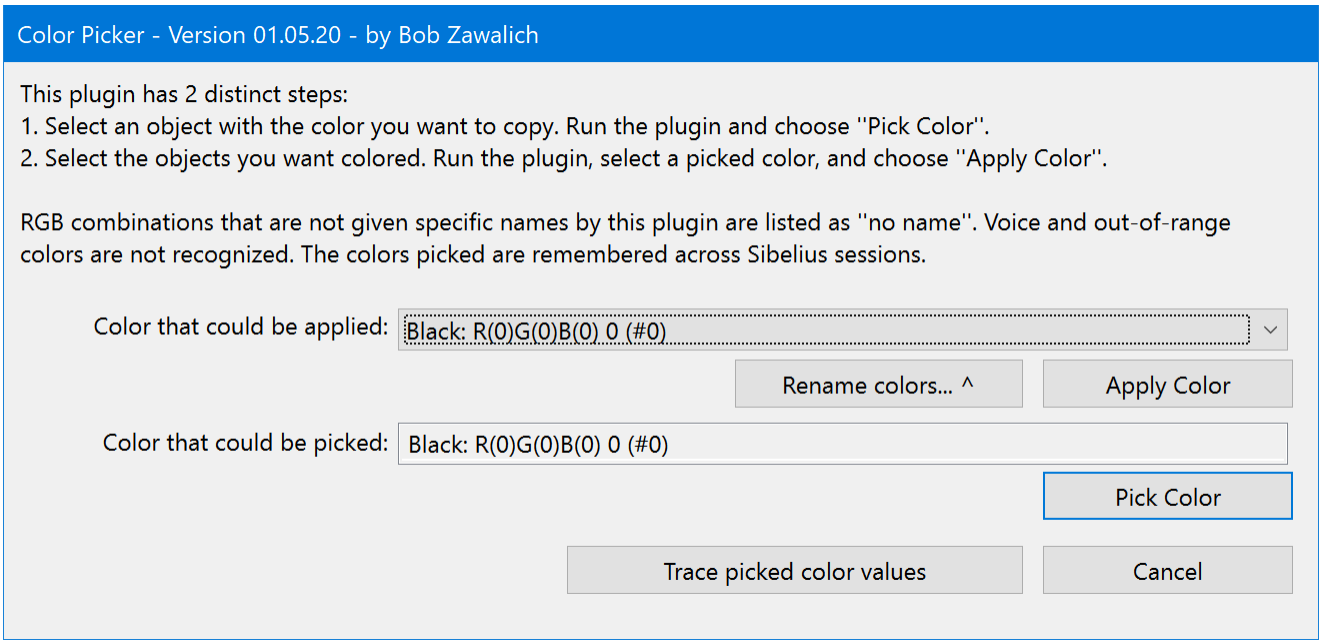
Though **Edit Colors** in **Apply Named Color** calls code in the plugin **Color Enharmonic Pitches**, the edited colors in each plugin are independent. Changing colors in one plugin does not affect the other.

Color Picker

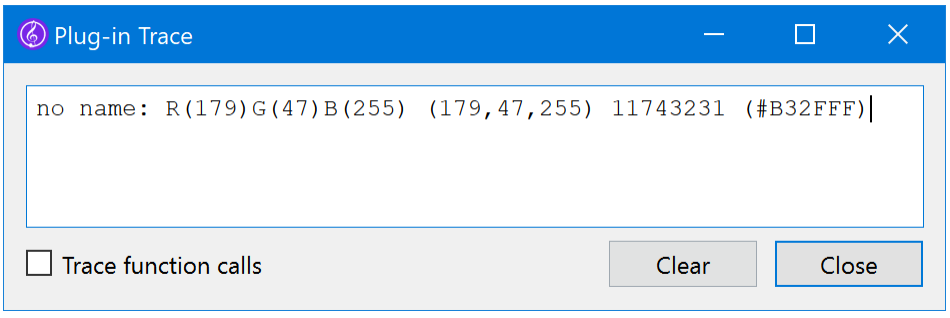
Color picker was originally designed to let you select a colored object and save the RGB values of that object so you could apply that color to other objects. As the dialog explains, this a a 2-step process:

- Select a colored object and run the plugin, choosing **Pick Color** to save the RGB values.
- Select objects you want to color, run the plugin, choose a color to apply, and choose **Apply Color**.

This can be useful if you had used **Home>Edit Color** to choose a custom color and did not remember which RGB values you used.



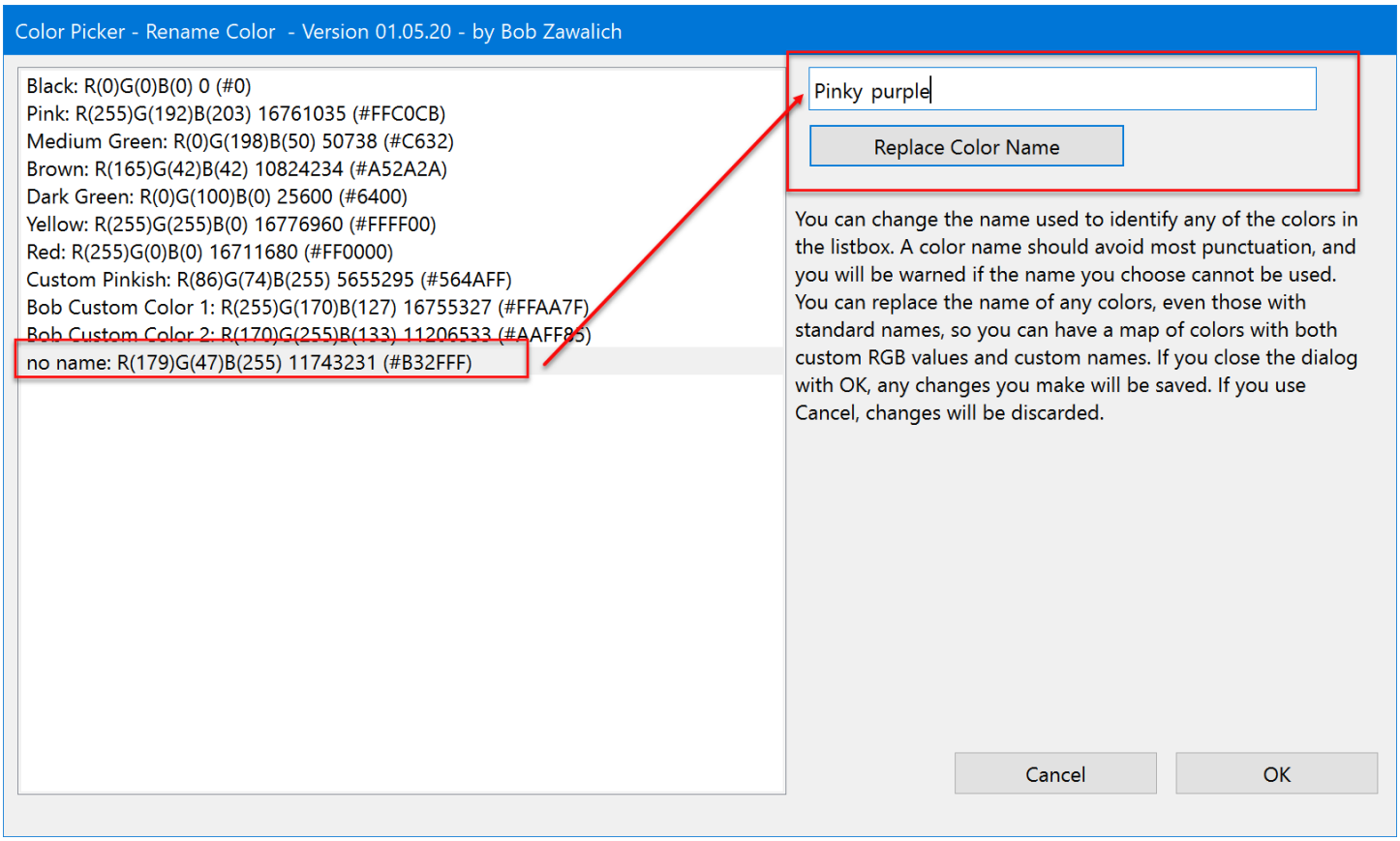
I decided that it could be useful to know the RGB values of specific colors when editing colors in the 2 plugins already discussed, so I added a **Trace picked color values** button here to display the color name and RGB values for colors for which you had pressed **Pick Color**.



It then occurred to me that it could be useful to have multiple picked colors rather than the single color it had, so I changed the **Color that could be applied** field from a text object to a drop-down listbox, which could store multiple colors. I then changed the plugin so the list of picked colors would be saved across Sibelius sessions. This would let you build up a library of custom colors you could use in multiple scores.

I have assigned color names in other plugins to a relatively small number of RGB values, and unnamed colors are usually identified only by their RGB color. This was not so bad if the color only existed when the plugin had just been run, but I thought it was not good to have a bunch of unnamed colors in the Color Picker list.

I added a **Rename Picked Colors** button, which lets you add or change the names used to identify any of the colors in the **Color that could be applied** list. You can only rename a color after you have chosen it with **Pick Color**.



I could imagine this list being used similarly to the way the color lists in **Apply Named Color** are used, to provide a consistent color naming and RGB color-name mapping for colors in multiple scores.

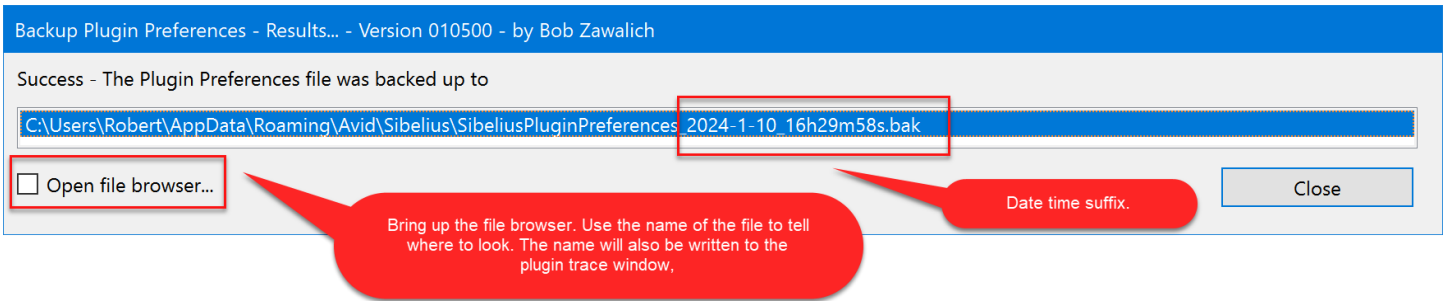
The color **names** used in **Color Enharmonic Notes** and **Apply Named Color** are the same, though the RGB values associated with the names can be different. The color map and color names used in **Color Picker** is completely independent of the other 2 plugins.

All these plugins can be installed in Sibelius 7 or later using **File>Plug-ins>Install Plug-ins.**

Appendix: Which dialog settings are saved and where?

Each of these plugins save dialog settings to the **Plugin Preferences** data file **SibeliusPluginPreferences.dat**. The Preferences data file is usually intended to save settings as a convenience for the next time the plugin is run, but for these plugins, the saved settings take a lot of work to set up and are hard to reproduce. The Plugin Preferences file can be fragile, so it should at least be backed up when settings are changed.

I recommend taking screen shots of dialogs when they are set up the way you want them, and you can also make backup copies of **SibeliusPluginPreferences.dat** by installing and running the plugin **Backup Plugin Preferences**. It will make a copy of the current file in the same subfolder as the original file and renames the copy so the current data and time are included, and the extension is .bak. Here is the dialog that appears when you run the plugin:



Open File Browser might be useful if you want to look at the file in context. The mechanism plugins use to put up a file browser is pretty good at starting in the desired folder. The file name will also be written out to the plugin Trace window when you run the plugin,

Success - The Plugin Preferences file was backed up to:
(C:\Users\Robert\AppData\Roaming\Avid\Sibelius\SibeliusPluginPreferences_2024-1-10_16h29m58s.bak)

You can go there later and copy the full path name it you want to use it in a browser. Ignore the parentheses at the start and end of the name in the traced output.

The **Preferences** plugin

You can run the shipping plugin **Preferences** to see what is being saved to **SibeliusPluginPreferences.dat**, and even edit the data if you are both daring and careful. I recommend backing up the Plugin Preferences file with **Backup Plugin Preferences** before doing any editing.