On Hiding Dialogs in Sibelius Plugins

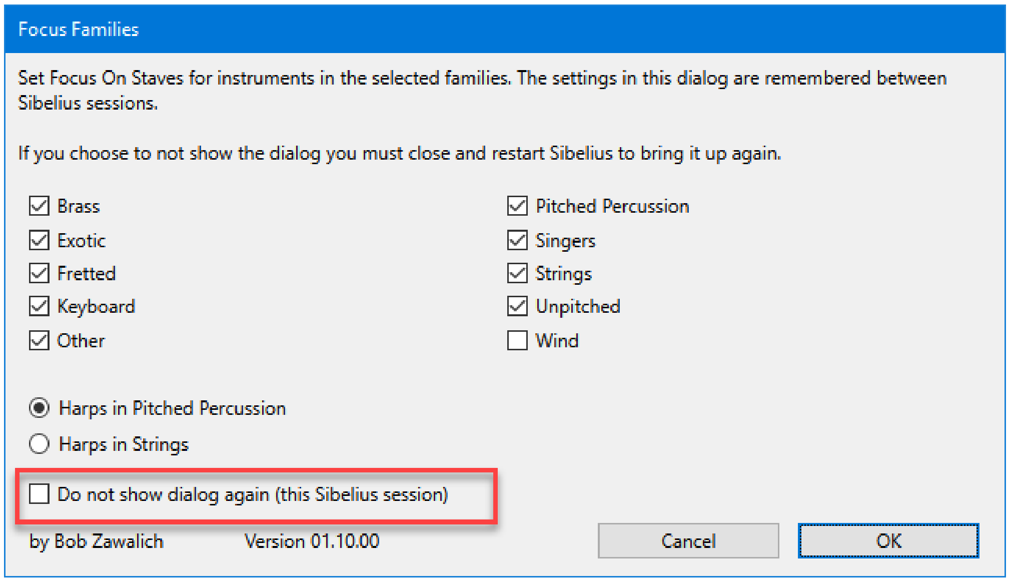
Bob Zawalich August 1, 2021

Many plugins present complicated a dialog with lots of options, and we might want to be able to run the plugin without having to deal with the dialog, especially when running it multiple times. Unlike Sibelius commands, which can store settings in Engraving Rules, a plugin needs to carry its settings around with it.

Here are some ways to avoid having to deal with dialogs every time you run a plugin.

Hiding a dialog

Most plugins with dialogs do not allow you to hide the dialog. Most that do allow it will have the dialog appear if there is no selection and will also reset the **Do not show dialog** setting off the first time the plugin is run in a Sibelius session. Here is an example of such a dialog.


Focus Families

Set Focus On Staves for instruments in the selected families. The settings in this dialog are remembered between Sibelius sessions.

If you choose to not show the dialog you must close and restart Sibelius to bring it up again.

- ☑ Brass
- ☑ Exotic
- ☑ Fretted
- ☑ Keyboard
- ☑ Other
- ☑ Pitched Percussion
- ☑ Singers
- ☑ Strings
- ☑ Unpitched
- ☐ Wind

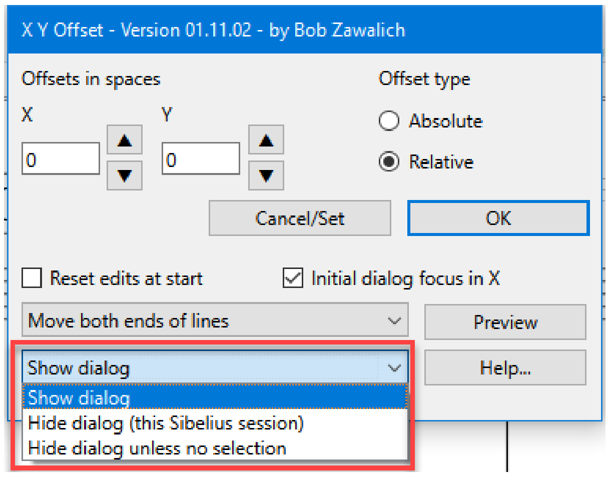
- ● Harps in Pitched Percussion
- ○ Harps in Strings

☐ Do not show dialog again (this Sibelius session)

by Bob Zawalich Version 01.10.00

You might still hide a dialog sometime in a Sibelius session and forget how to bring it back, but there is a good chance that you might have seen the warning “for this Sibelius session”, and even if you miss it or forget it you will be able to get the dialog back in the next Sibelius session.

Note Spacing and **X Y Offset**, though, have an extra option to always hide the selection. Here is the dialog for X Y Offset.


X Y Offset - Version 01.11.02 - by Bob Zawalich

Offsets in spaces: X 0, Y 0

Offset type: ○ Absolute ● Relative

☐ Reset edits at start ☑ Initial dialog focus in X

Move both ends of lines

Show dialog

Show dialog
Hide dialog (this Sibelius session)
Hide dialog unless no selection

I was reluctant to add this option, fearing that people would hide the dialog and not know how to get it back. **In both these cases, the dialog will appear if you run the plugin when there is no selection**. The plugin will warn you about this when you are choosing the option, but it is really easy to miss or forget the warning. And once the plugin is hidden, there is nothing to read.

There is not much room for a loud warning in the dialog itself, but I note that in the Help text for **X Y Offset** there is this warning:

"Hiding the dialog:

The list box lets you always Show the dialog or Hide it for the rest of the Sibelius session, and there is a new option to Always Hide the dialog unless you run the plugin with no selection.

This can be convenient if you use the plugin a lot, and it can be mystifying if you set it and then don't use it for a long time, and cannot remember how to get the dialog back.

The dialog will always appear if there is no selection, and you can change the dialog settings then. The settings are saved across Sibelius sessions even if you cancel the dialog**. Do not complain to me if you choose this option and cannot remember how to get the option back. This is a power user option**."

Still, I prefer to not have users be stuck when they can't figure out how to get the dialog back, and do not plan to use this mechanism again. It is too easy to forget how to get the dialog back, especially if you hid it a while ago and need to use it again.

The settings for **Do Not Show Dialog** for these plugins need to be stored in the **Plugin Preferences**, and so uninstalling and reinstalling the plugin will not help. Technically, you could edit the settings in the plugin **Preferences**, but that is really beyond even most power users to know to do that.

Ensure that the dialog settings are what you want

One thing I want to emphasize before continuing is that **if you hide a plugin's dialog you need to know what the settings will be when the plugin is run without a dialog**. The effect of hiding the dialog is the equivalent of bringing up the dialog and immediately choosing OK.

Pay attention when you choose a hiding option and check before you do a lot of work with a plugin that might do things you are not expecting.

Issues with hiding dialogs

It can be useful to be able to run a plugin without a dialog, especially when plugins run other plugins, as in Command Macros and Command Plugins in Sibelius 2021.2 and later, so I looked for a different way to do this.

The "Hide always" model of **Note Spacing** and **X Y Offset** is both easy for the user to forget and tricky to implement in a plugin. The Hide dialog code is often tied up with checking for an empty selection, and possibly offering to select the entire score or exit, and it can be remarkably difficult to retrofit a plugin to support the new option.

Also, when you bring up a dialog when there is no selection, you typically just want the dialog to come up so you can check or change the options, and you typically do not want to continue after the dialog goes down, since nothing has been selected. That often requires extra code to handle an graceful exit if the user wants to continue after the dialog goes down.

The Run Plugin Hide Dialog model

Hiding dialogs, especially with the standard model of always showing the dialog at least the first time in a session, gives the flexibility of being able to see the settings, and change them as needed.

There are situations, though, where you really never want to see the dialog, as when calling it from a macro. The **Run Plugin Hide Dialog** model, when called plugins are set up to be called from another plugin without showing their dialog but not changing any internal settings in the called plugin, works pretty well in that situation.

The problem with it is that the user can run that plugin and show the dialog, and change the settings to something that the macro does not expect; then when the macro runs and the dialog is hidden, there can be unexpected results. If you forget that your macro calls a plugin with the dialog hidden, or you give the macro or plugin to another user who has never known that it will call that plugin, then the settings in the dialog when the plugin is called and unknown, and can have a significant effect on the results.

This will probably not happen all that often, but when it does happen it will be both unexpected and mysterious, neither of which is good.

Sometimes you need to control all the dialog settings when you call another plugin.

Clone plugins

One approach I have used for plugins, especially those that save their preferences across Sibelius sessions, is to make clones of the plugins to be called (using the plugin **Copy Plugin**). Once the clones are created, they can be run with the dialog showing, and the dialog settings can be established. The original plugin could be used as it had always been used, and **Run Plugin Hide Dialog** or the **cmdutils** equivalent could always be run on the clone.

The **Focus On Staves** plugin suite makes use of both cloned plugins and parent/child plugins.

In addition to the work setting up the clones, you would also need to be sure to make new clones any time you update the original plugin, so it can be a bit tricky in the long term, but very effective in the short term.

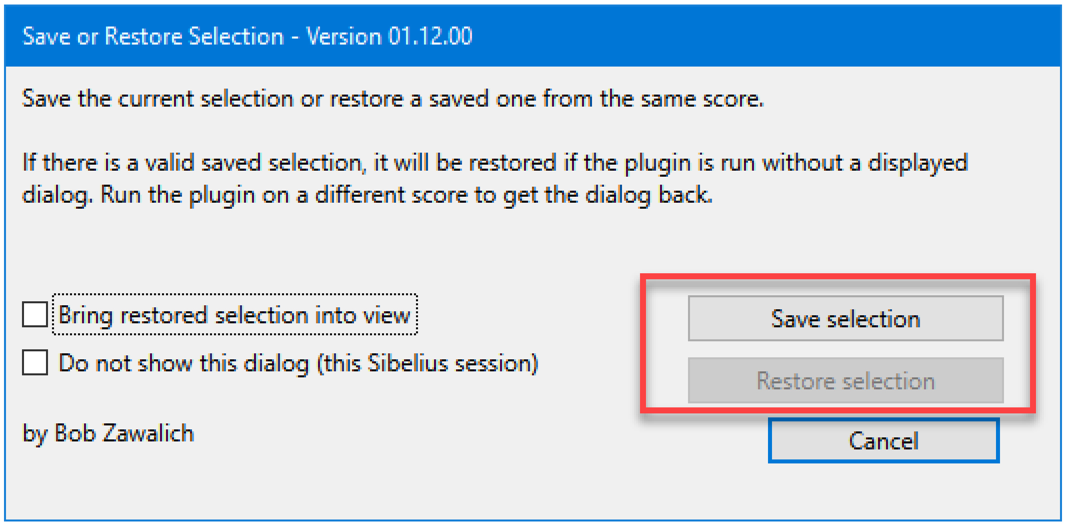
The advantage of the clone model is that it can be done pretty easily by users, especially if they use Copy Plugins to produce the clones. A use can produce copies of existing plugins, with different names, without having to change any code manually.

If the plugin does not save its settings in the Preferences data base, a most clones can be set to use different dialog settings by running the clone once, changing the dialog to use your desired settings, and choosing OK to be sure the dialog settings will be set. (You can try running it again, check that the dialog is set up as desired, and then cancel). Now go to File>Plug-ins>Edit Plug-ins, and use Find to find your clone plugin. Click on Edit, and then **without changing anything,** click on OK. The dialog settings will now be stored in the plugin file itself, and should be the defaults from now on.

This will not work for some plugins that hard-wire dialog settings, but most of the plugins I have written will allow you to change the setting this way. This process is discussed in more detail in this Scoring Notes post.

Parent/Child plugins

The Parent/Child plugin model is based on the work I have done in **cmdutls.plg**, which itself is based on some plugins like **Save and Restore Selection**, which is published with a *Parent* plugin, **Save Or Restore Selection**, and 2 *Child* plugins, **Save Selection** and **Restore Selection**.


Save or Restore Selection - Version 01.12.00

Save the current selection or restore a saved one from the same score.

If there is a valid saved selection, it will be restored if the plugin is run without a displayed dialog. Run the plugin on a different score to get the dialog back.

☐ Bring restored selection into view
☐ Do not show this dialog (this Sibelius session)
Save selection
Restore selection
Cancel

by Bob Zawalich

The children pretty much do what the **Save Selection** and **Restore Selection** buttons do. In this case they need to be aware of whether there is a selection to save, or there is one to restore (in the screenshot, there is no saved selection and the **Restore** button is disabled, so **Restore Selection** has to deal with that scenario since it will not be able to restore anything. In such a case, it would just put up a message box describing the situation, and then the plugin exits.

There are a lot of details, so please see the document **Parent-Child plugins** for the details, including using such plugins with the **cmdutils** command **RunPluginEntry_cu**.