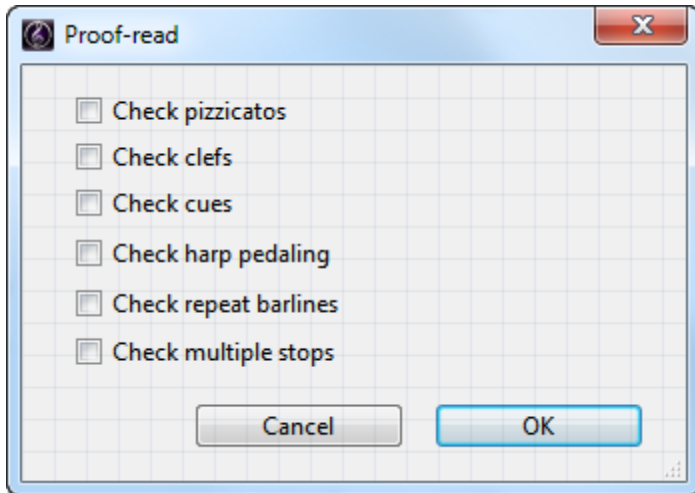


Plug-ins That Run Other Plug-ins

Bob Zawalich March 3, 2016

Since Sibelius version 1, plug-ins have been able to run other plug-ins. For example, the **Proof-read** plug-in is just a container that lets you call the main **Run()** method of the plug-ins Check Pizzicatos, Check Clefs,... so you can do all your proofreading at once. These called plug-ins were intended to be called by other plug-ins, and typically do not display a dialog when they run.



Some plug-ins are designed to be libraries of code to be called exclusively by other plug-ins; often they do not appear on the plug-in menus as all. An example of this is the plug-in **utils**, which provides a number of useful and debugged bits of code for other plug-ins to use, for things like absolute value or rounding a number to a specific number of decimals. Such a plug-in could have a number of methods, each of which can be called by a plug-in.

This document will discuss plug-ins that call the top-level Run() method of other plug-ins, which is equivalent to running the plugin from the plug-in menu.

There are 2 venerable plug-ins that let a user select a plug-in to run. Hans-Christoph Wirth wrote **Fixed Plug-in Shortcuts** at a time when keyboard shortcuts to plug-ins would be reassigned by Sibelius every time you added or removed a plug-in. Hans-Christoph's plug-in would let you set up shortcuts to a number of plug-ins and they would not change when plug-ins were added. If you assigned a shortcut to **Fixed Plug-in Shortcuts**, it might change, but you could reassign it and only need to change the one.

I had used **Fixed Plug-in Shortcuts** quite a bit, but it has a bit of setup every time you wanted to add a new shortcut to a plug-in, so I wrote **Run Plug-ins By Name**, which provides an alphabetical list of plug-ins you can run immediately. It also puts the names of the last several plug-ins you had run at the top of the list so it would be easier to find them next time. If you assign a shortcut to **Run Plug-ins By Name**, you will have easy access to all the plug-ins you have run recently, without needing to assign separate shortcuts for them.

You can read about these plug-ins in the Scoring Notes Blog post [Using Run Plugins By Name or Fixed Plug-in Shortcuts](#).

Dealing with Duplicate Plug-ins

If you have multiple copies of a plugin in different locations (typically a downloadable plugin you have forgotten about installing, and so you installed it again to a different location), you may not be running the copy of the plugin you expect.

Duplicate plugins in Sibelius are always bad. Even if you run a plugin yourself from a plugin menu, it can run the wrong one. Sibelius ignores the path name of the plugin file and always runs the first example of this file name it finds. If you have multiple versions of a plugin in different folders you are likely to run the wrong one. So clean up those duplicates! For more details please see the Of Note post [Eliminating Duplicate Plugins in Sibelius](#).

Multi-running any plug-in

There are plug-ins, especially in the Batch Processing category, that can run against all the scores in a folder, or against all the parts in a score. These plug-ins had to be specially designed to do this, and making a plug-in run this way required a lot of programmer work.

I decided to write a plug-in that could run an existing plug-in on all the scores in a folder without having to change the called plug-in, and so the folder could be processed without custom programming. I ended up with **Run Plugin on Folder of Scores**. That seemed to work pretty well, so I also wrote **Run Plugin on Score and Parts**, which will run the target plugin on the current score and all of its dynamic parts. Both plug-ins are in the same zip file on the plug-in the download page under the name of **Run Plugin on Folder of Scores**, in the **Batch Processing** category. If you use the plug-in installer, it will install both plug-ins.

Issues unique to running a plug-in more than once

Most of the plug-ins that call other plug-ins will call a plug-in once. If that plug-in puts up a dialog or a message box, it is no big deal. When plug-ins call a plug-in multiple times, having dialogs and messages pop up every time the plug-in runs can be annoying, especially if there are a large number of scores or parts.

The plug-ins that will work best when run this way either do not put up a dialog, or have an option to show a dialog the first time the plug-in is run in a session, but hide it on further runs, such as **Find and Replace Text**. If they also do not display results in a message box at the end, there is a better chance that such a plug-in can be run on a folder without interruptions.

Run Plug-in On Folder of Scores and Run Plug-in on Score and Parts

These plug-ins allow you to select a plug-in from an alphabetical list of plug-in names, and run it either on all the scores in a folder (so the called plug-in will be called once for each score in the folder) or will run it on the score and each of its parts.

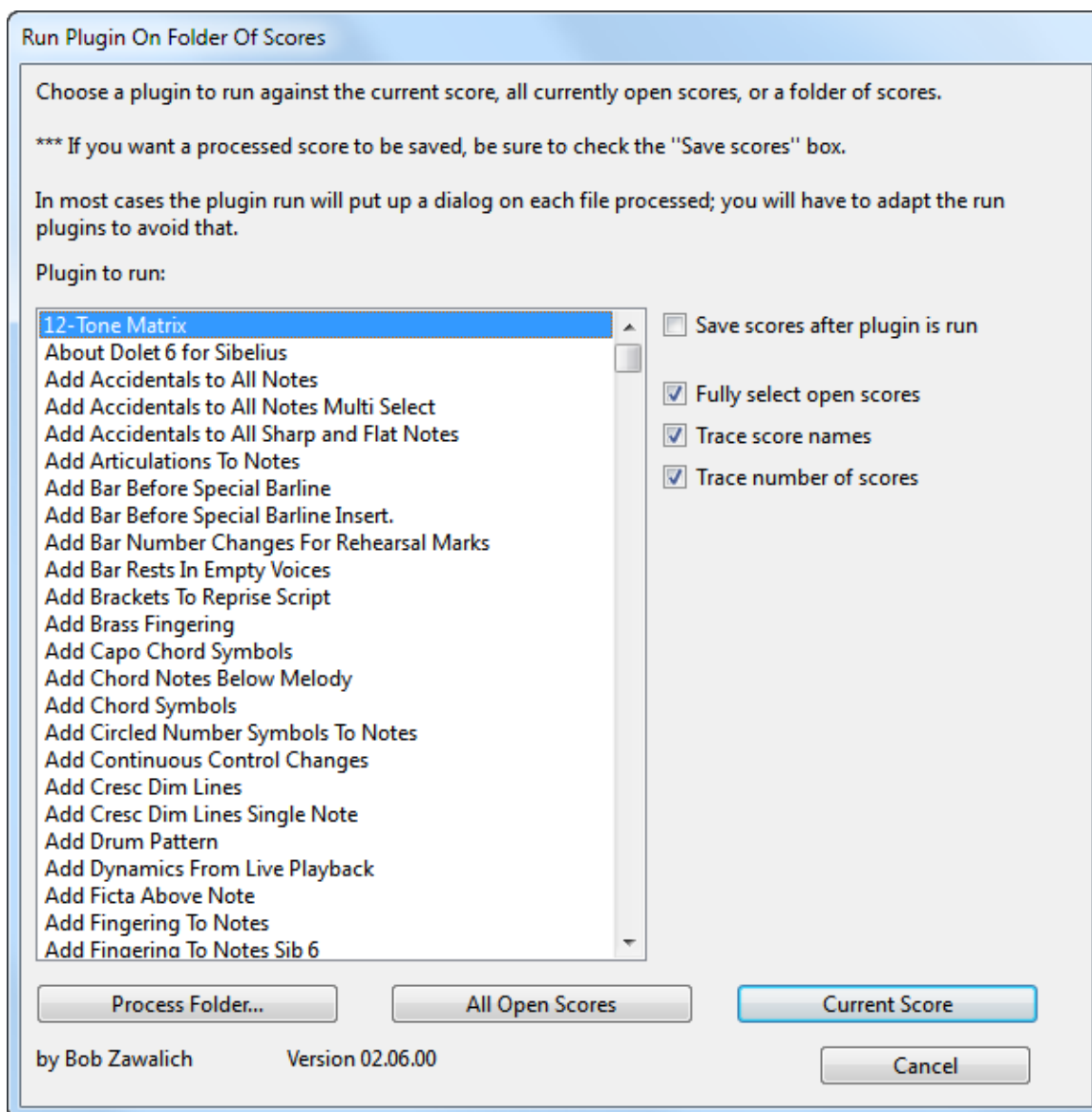
Both plug-ins do some special setup before calling their target plug-in. Because many called plug-ins work on a selection, these plug-ins select the entire score or part before calling the target. They also make any score or part they open the active score, because most plug-ins operate on the active score.

These plug-ins have no idea what the called plug-in will do, and so they try to be flexible enough to deal with most situations.

Run Plugin on Folder of Scores

This plug-in has a number of options; the main one being an option to save processed scores. By default it will not save the scores that are processed. Typically, a plug-in that modifies a score will save the score itself, so we would not want to save it twice. Also you might only be inspecting a score, and so would not want to save it. On the other hand, if the called plug-in does modify a score but does not save it (since it expected to be working on a file that was open and assumed you would decide to save it or not when you close the file), then there is an option to have **Run Plugin on Folder of Scores** do the saving.

It can also write the score name to the trace window, change the selection to a full selection for files that were already opened (it always fully selects scores it opens itself), and tell you how many scores were processed.



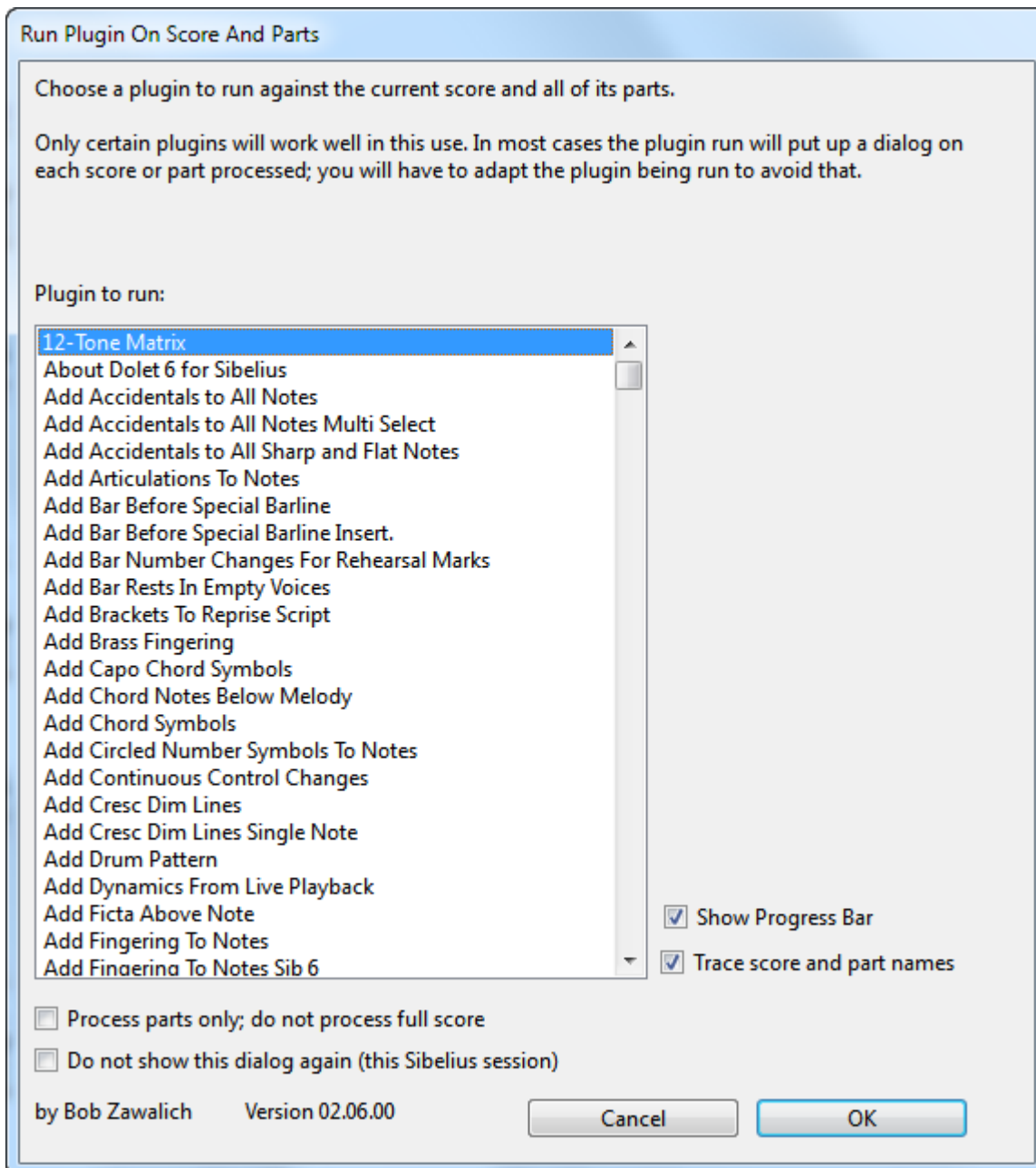
Run Plug-in on Score and Parts

This plug-in will be run once for the full score (if you chose to process it) and once for each dynamic part. It will select the entire score or part, and will make the score or part the active score before calling the target plug-in.

It has some options that reflect the fact that it will call other plug-ins, and that it can be called by another plug-in (such as **Run Plug-ins on Folder of Scores**).

By default it will display a progress bar, which allows you to cancel the plug-in while it is running. Progress bars cannot nest, so if the target plug-in has its own progress bar, this one will be cut off, so it is probably best to not show it, or turn off the progress bar in the target plug-in if possible. Similarly, if a plug-in that has a progress bar calls **Run Plug-in on Score and Parts**, that progress bar could get cut off. It is usually best to keep the progress bar for the calling plug-in, rather than the called plug-in, if possible.

This plug-in can also trace the score and part names. If the called plug-in or the calling plug-in also traces names, you might want to turn this option off.



Adapting plug-ins to work better when called by other plug-ins (mostly for Manuscript developers)

Run Plug-in on Score and Parts has an option that makes it well suited to being called by other plug-ins. It will always show its dialog the first time the plug-in is run in a Sibelius session, but you can turn it off for any later runs in the same session.

Without this option, if we were to run this plug-in on a folder of scores, the dialog would appear for each score. What you usually want is to set the plug-in options once, and then run it on each score without seeing the dialog again. The option **Do not show this dialog** will let you do that, and it can be very nice to have when you are processing several hundred scores.

Be aware that if you hide the dialog you will not be able to get it back without closing and restarting Sibelius.

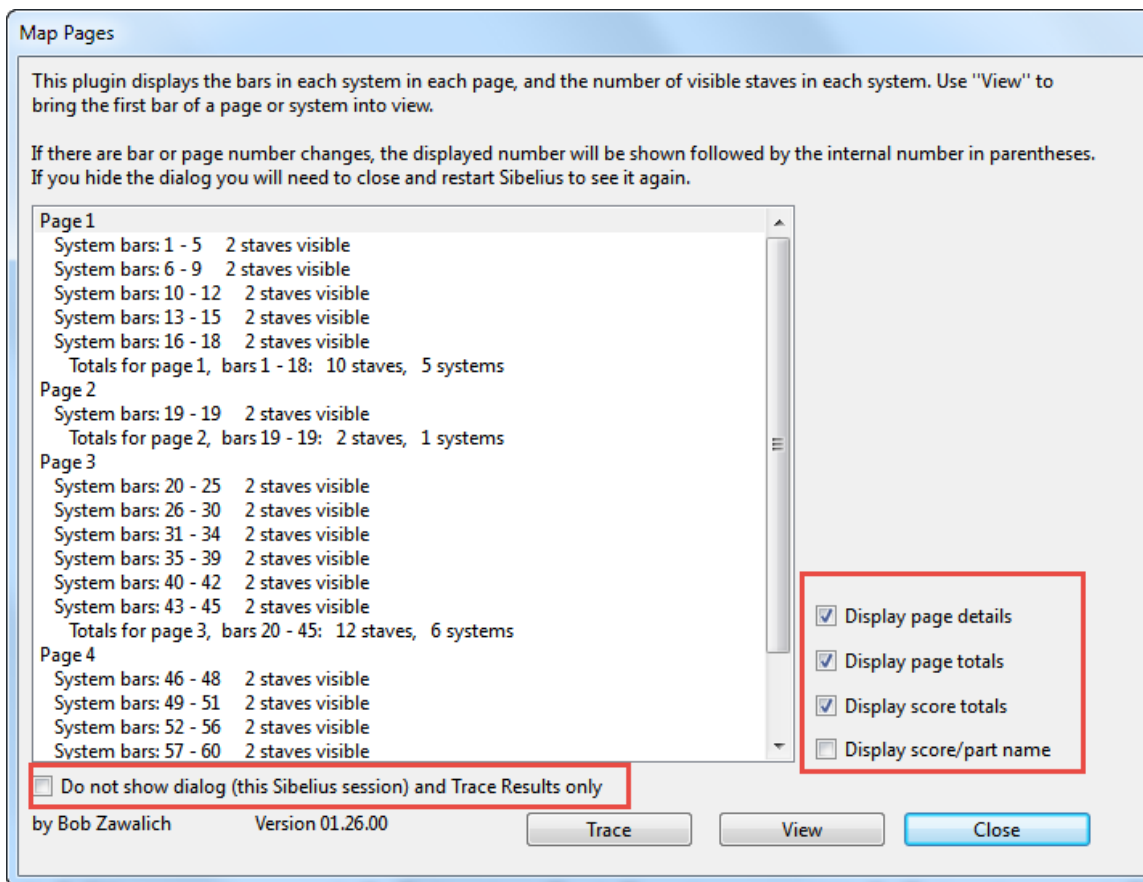
This is probably the simplest modification that can be made to a plug-in so it can be easily run from another plug-in. If there is output from the plug-in, it is best to have it write the output to the plug-in Trace window, rather than putting up a message box at the end of each run.

Run Plug-in on Folder of Scores and **Run Plug-in on Score and Parts** were written in part so that plug-in developers could write simpler plug-ins, without having to deal with getting all the scores in a folder or all the parts in a score. New plug-ins with no dialogs or with first-time-only dialogs can be written, or existing plug-ins can be adapted to be run on multiple scores.

I wanted to have the plug-in **Map Pages** be run by **Run Plug-in on Score and Parts**, but Map Pages worked by putting up a dialog and letting you navigate in the score while the dialog was up. I gave Map Pages an option to not show the dialog **and** to write all the output to the trace window.

To run this plug-in from **Run Plug-in on Score and Parts**, it might be best to run **Map Pages** first, and choose the desired options including **Do not show dialog** and **Trace Results**. Then when it gets called, it will immediately trace its results. If the plug-in called Map Pages when the dialog would show, you would need to explicitly use the **Trace** button for the full score, then tell it not to show the dialog, so all the parts would be traced.

We can now choose **Map Pages** to be run by **Run Plug-in on Score and Parts**.



Sometimes it will take some preplanning to make running all the plug-ins work smoothly.

An elaborate example: Run Plugin on Folder of Scores calls Run Plugin on Score and Parts, which calls Map Pages

Most times when you are using one of these plug-ins to call another it will not be this complex, but this shows what is possible.

In this case we will want to run **Map Pages** first so the dialog will be hidden the first time it is called.

Since all 3 plug-ins can display score names, we want to decide the best place to do it. In this case I suggest doing is as far down the chain as possible, though you would get the same results showing the score and part names from **Map Pages** or **Run Plug-in on Score and Parts**. So let's check that box in **Map Pages**, and then hide the dialog.

We can now run the top level plug-in **Run Plug-in on Folder of Scores** (see the image of the dialog above).

Map Pages does not change the score, so we can uncheck the option to save the score. **Map Pages** will trace the score and part names, so we can uncheck that option as well. You need to know a bit about how the called plug-in handles a selection if you are processing an already open score. In this

situation, it is likely that you should keep the option to select open scores checked. You can trace the number of scores processed or not, as you prefer.

Now choose **Process Folder...** to select a folder, and then use OK to run **Run Plug-in on Score and Parts**. Assuming it is the first time you are doing this in a session, the dialog will come up so you can choose options.

Run Plug-in on Folder of Scores will display a progress bar, so we should turn off the one in **Run Plug-in on Score and Parts**, otherwise we will not be able to cancel **Run Plug-in on Folder of Scores**, and things might run for a long time. Similarly, **Map Pages** will trace the score and part names, so we should turn it off here.

You can change the plug-in to process parts only, but most likely you will want the full score as well. Now you can choose the option **Do not show dialog**, and hit OK to run it.

Map Pages should now run a large number of times uninterrupted, dumping great quantities of data into the plug-in **Trace Window**. This will show the page structures for each part in a folder of scores.

If this seems like a lot of work to set up, imagine running **Map Pages** on each score and part in each file in the folder by hand. It probably will not seem so bad!

Appendix: published plug-ins that have a Do Not Show Dialog option as of March 2016

These are possible candidates for being run multiple times. I have excluded plug-ins that would really make no sense to run multiple times, or which only make sense running on a selection. Some of the plug-ins below may still not be appropriate to run multiple times.

Sometimes you will need to run the plugin once to setup the options and hide the dialog before running in on a score and parts. Experiment a little!

Downloadable plug-ins

Accidentals\DisplayAccidentalsInSelectedNotes.plg

Chord Symbols\BracketChordSymbols.plg

Chord Symbols\RespellChordSymbols.plg

Color\ReplaceColor.plg

Composing Tools\ExplodeStaffToParts.plg

Composing Tools\ReduceStavesToFullScore.plg

Developers' Tools\AlignNotes.plg

Developers' Tools\MultipleSelectionToPassage.plg

Engravers' Tools\AddLVSymbolsToNotes.plg

Filter and Find\FindAndReplaceMulti-Filter.plg

Layout\MapPages.plg

Lines\BracketNotes.plg

Notes and rests\AddInterval.plg

Notes and Rests\FillSelectionWithSlashNotes.plg
Notes and rests\HideArticulations.plg
Notes and rests\MergeToVoice1.plg
Notes and Rests\SimplifyTied-ToNotes.plg
Notes and rests\SlashNotesAtMidLine.plg
Notes and rests\Unarpeggiate.plg
Other\InsertOneSymbol.plg
Playback\MuteUnmute.plg

Shipping plug-ins

Text\FindAndReplaceText.plg
Transformations\AugmentIntervals.plg
Transformations\DiminishIntervals.plg
Transformations\TransformScale.plg