

The Store Data for Wildcard Plugin

Bob Zawalich Updated January 29, 2023

Description

Store Data for Wildcard is an extension of the **Get Page Count** plugin, adding the ability to use the score's highest Version number.

It stores the number of pages in a score (at the time the plug-in is run) or the highest Version number of the score, or several other bits of data, into a field of **File > Score Info**.

This makes the number available for use in text fields (such as \"Page x of y\"). If the MOREINFO wildcard is used. (The full text would be \"Page \\\\$PAGENUM\\ of \\\\$MOREINFO\\\").

The plugin was updated 11 November 2017 to add additional fields including dates and file names that will be rendered as text, and could be used in cloud sharing scores using a wildcard from a Score Info field.

Usage

You can use wildcards in Text objects to get Score Info data into a score and have the data update without having to directly change the text in the score.

You can read about wildcards in these really great blog posts.

<https://www.scoringnotes.com/tips/unleash-power-text-wildcards/>

http://www.rpmseattle.com/of_note/creating-and-using-wildcards-in-sibelius/

There are times when the data you want is not available to a wildcard. The classic example of this was the number of pages in a score prior to the addition of the \\\\$NUMPAGES\\ wildcard in Sib 7.1.2. If you wanted a footer that said Page xx of yy you could use \\\\$PageNum\\ for xx but you would have to hardcode yy, and if you changed the number of pages you would have to update the text in the footer.

As a workaround, people started reusing Score Info fields, typically the **Other Information** field, which lets you store the value you needed (e.g., the number of pages) in Other Information and then use a \\\\$MoreInfo\\ wildcard in the footer. Your footer would then look like

Page \\\\$PageNum\\ of \\\\$MoreInfo\\

The number of pages still needs to be updated if it changes, but you can change it in Score Info without having to find and edit text objects that use the value, and you can use the wildcard freely in text.

I wrote the plugin **Get Page Count** to automate the process. It would find the number of pages in the score (optionally excluding title pages and blank pages) and store that value in the Score Info field of your choice. It would put the wildcard text you would need in an edit box so you could copy it into your score.

Get Page Count

The number of pages will put into the chosen Score Info field for the score and each of its parts.

Create text using a \ScoreInfo\ wildcard to display the number of pages. To have a text item say "Page x of y" use the text "Page \ScoreInfo\ of \ScoreInfo\ ", which can be copied from the edit box.

Rerun the plug-in to update the wildcard value if the number of pages changes.

Score Info field to hold page count: Other information

Text for wildcard: Page \ScoreInfo\ of \ScoreInfo\

☐ Exclude Special Page Break title page and following blank pages from count (Sib 5 and later)

by Bob Zawalich Version 01.07.00 Cancel OK

If the number of pages changed you would need to run the plugin again, and be sure to use the same Score Info field, but all in all it simplified the process.

Sometime later, I needed to use the highest version number of a score in some text and it was not available in a wildcard. I made a variant of Get Page Count, called **Store Data for Wildcard**, that could get either the page count (though without the options) or the highest version number, and store the value of the chosen data field in a Score Info field. It would also provide a copyable wildcard text.

Store Data For Wildcard Version 01.10.02

The selected data to be stored will be put into the chosen Score Info field for the current score or part.

The data itself will be "hard-coded" and will not be updated if its source changes.

Create text using a \ScoreInfo\ wildcard to display the desired data, or copy the data from the edit boxes into text objects in the score.

Rerun the plug-in to update the Score Info value if the data that was stored changes.

Data to be stored: Page Count

Score Info field to hold data: Other information

Text for wildcard: Page \ScoreInfo\ of \ScoreInfo\

Wildcard replacement text: 2

by Bob Zawalich Cancel OK

When I was first using the **Cloud Sharing** feature, I wanted to include a footer that showed the file path and last saved date and time to help identify the score. The footer used the FILEPATH and FILEDATE wildcards, but on screen these did not render as the name and date, but just showed the wildcard text.

It turns out that while most wildcards work with Cloud Sharing, FILEPATH, FILENAME, and FILEDATE do not render, USER renders as "System", and the date and time wildcards (DATESHORT, DATELONG, TIME) render as Greenwich/GMT/UTC time, rather than local time, and could be confusing.

I considered hard-coding the file data, and that made me think of **Save Data for Wildcard**. While, unlike NumPages and the highest version number, the file and date data were available in a wildcard, the data rendered onscreen was not what I wanted, so I added more fields to the list of storeable data in the plugin.

The selected data to be stored will be put into the chosen Score Info field for the current score or part.

The data itself will be "hard-coded" and will not be updated if its source changes.

Create text using a \\${MoreInfo}\ wildcard to display the desired data, or copy the data from the edit boxes into text objects in the score.

Rerun the plug-in to update the Score Info value if the data that was stored changes.

Data to be stored: Highest Version Number

Score Info field to hold data: Other information

Text for wildcard: Version \\${MoreInfo}

Current wildcard replacement text: 1

☐ Do not show this dialog (this Sibelius session)

☒ Trace results

New macro/plugin...

Help...

Cancel

OK

These are the currently available fields: (updated 29 January 2023)

Page Count
 Duration at End of Score
 Highest Version Number
 File Name
 File Name with Path
 File Path
 Current Date Short
 Current Date Long
 Current Time Short
 Current Time Long
 Modified File Date
 Modified File Time
 Title - Part
 File - Part
 File - Part - Current Date - Current Time
 File - Part - Modified Date - Modified Time
 File - Current Date - Current Time
 File - Modified Date - Modified Time
 Sibelius Version Number
 Transposing Score Status

This covers the File and Date/Time fields (I did not bother with USER) and gives some combined File/Date strings that could be used in footers as a single piece.

As with the Page Count, you will need to run the plugin again if its data changes, but otherwise you can set up a wildcard with the plugin and use it in text, and it will appear in a **Cloud Sharing** score the same way it appears on your machine. These wildcard fields will also work fine in ordinary scores.

Wildcards for Score Info fields

Arranger - \ \$Arranger\
Artist - \ \$Artist\
Composer - \ \$Composer\
Composer's dates - \ \$ComposerDates\
Copyist - \ \$Artist\
Copyright - \ \$Copyist\
Dedication - \ \$Dedication\
Lyricist - \ \$Lyricist\
Opus number - \ \$OpusNumber\
Other information - \ \$MoreInfo\
Publisher - \ \$Publisher\
Subtitle - \ \$Subtitle\
Title - \ \$Title\
Year of composition - \ \$YearOfComposition\

Recent Updates January 2023

I recently added some different time and date formats, and the **Sibelius Version number** that created the score, and the **Transposing Score Status**.

I also added a **New macro/plugin...** button, which will generate a command string that can be used in the Execute Commands plugin to produce a macro or plugin that will run **Store Data for Wildcards** with the options shown in the dialog when the **New macro** button is pressed, without bringing up the dialog. For example, if using the options from the previous screenshot, this command could be generated:

RunPluginEntry_cu(StoreDataForWildcard.plg, API_ProcessSelection, str_Data, **Highest Version Number**, str_ScoreInfo, **Other information**, str_TraceResults, **yes**)

When added to a macro or plugin, this will run **Store Data for Wildcards** and will add the current replacement text (1) to the **Other Information Score Info** field. You can then create a piece of text using the \ \$MoreInfo\ wildcard, and 1 will appear in the score. If you version changes, you will need to run the plugin/macro again, and it will update the **Score Info** field, and thus anything that uses the \ \$MoreInfo\ wildcard.

New macro/plugin

If this button is enabled (in Sibelius 7.1.3 or later) the command string for the dialog settings will be written to the plugin **Trace Window**. It can be copied from there and pasted into **Execute Commands** using its **Add Command** button. At that point **Execute Commands** will treat the command line like any other command. If the plugin **Custom Search Shortcut** is installed and you are running a recent version of Sibelius Ultimate, you will also see the following dialog.

New Plugin / Append to Command List

You can generate a custom plugin that runs the command line that has just been traced.

You can also add the command line to the Command List used by the plugin Execute Commands. The added command line will be at the bottom of the Command List the next time Execute Commands is run.

If you generate a plugin you will be given a chance to review and edit the new plugin location and name. You can just press OK/Enter in that dialog to accept the default name and location.

If you generate a plugin you will need to close and restart Sibelius before you can use it.

Current command line:

RunPluginEntry_cu(StoreDataForWildcard.plg, API_ProcessSelection, str_Data, Highest Version Number, str_ScoreInfo, Other information, str_TraceResults, yes)

☒ Generate a new plugin from the command line

☒ Append the command line to the Command List in Execute Commands

Cancel OK

It will let you automatically generate a new plugin that will run the command line, and/or it will add the command string to the bottom of **Execute Command's Command List**. The next time you run **Execute Commands** in the same Sibelius session, you will find the command string in the command list, ready to be combined with other commands, without needing to copy and paste it there yourself.

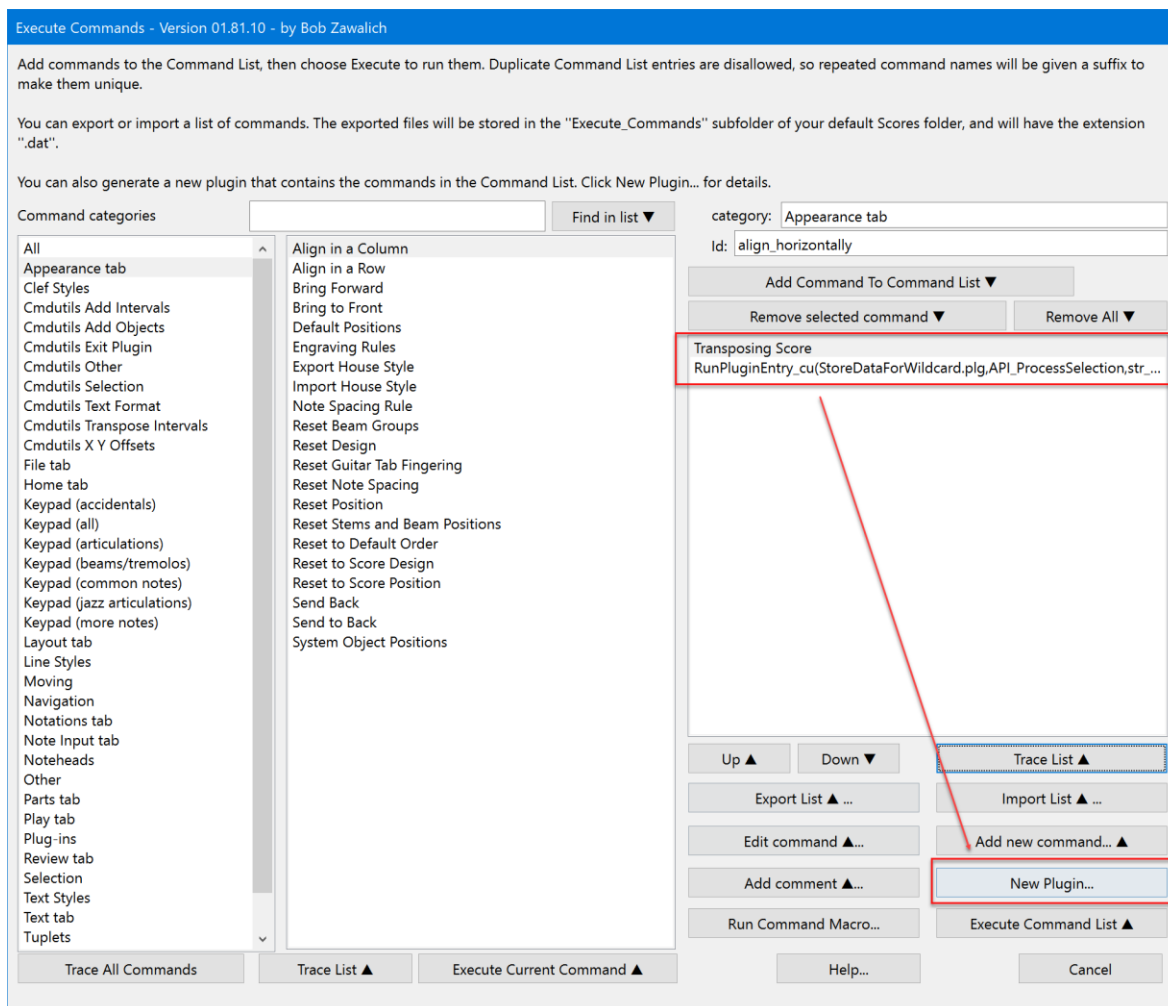
Combining a command string with other commands in Execute Command

It can be hard to remember to run **Store Data for Wildcards** before you save, print, or export a score. One approach to consider is to create a macro or plugin that will run it along with another command you will need to use anyway.

In the case of **Transposing Status**, you might create a plugin that you would use instead of clicking on the **Transposing Score** Ribbon button. If you had created a command line for **Transposing Score Status** using the **New macro/plugin button** in **Store Data for Wildcards**, you could set up commands in **Execute Commands** to run the **Transposing Score** command followed by the generated command line. It might look like this in the **Execute Commands** Command List:

Transposing Score
RunPluginEntry_cu(StoreDataForWildcard.plg,API_ProcessSelection,str_Data,Transposing Score Status,str_ScoreInfo,Other information,str_TraceResults,no)

Use **New Plugin...** to convert these commands into a plugin. After closing and restarting Sibelius you will be able to assign a shortcut to the new plugin and run it. Now you just need to remember to do that instead of clicking on **Transposing Score**!



You can also make a plugin like this that will run these commands and then save, print or export the score. These are easy to generate using **Execute Commands**.

WARNING!

Be very aware that in all these cases, there is a difference between **Score Info** wildcards, as I am using here, and something like **\\$PageNum**. **\\$PageNum** will be updated whenever the page number changes, so its value is always up-to-date. **\\$MoreInfo** or any of the other Score Info wildcards will only update then the **Score Info** field is changed, either manually or via a plugin. If you want to be sure that the **Transposing Status**, for example, is correct when you save or print the score, you need to run **Store Data for Wildcards** or its command string before you save, print, or export.

The advantage of reusing a Score Info wildcard in this way is that if you have the wildcard in some random Text objects in your score, you don't need to find them all and change the text in the score. You just need to update the **Score Info** field, as you would do if updating the **\\$Title** wildcard. You can just run this plugin again to accomplish that. These pseudo-wildcards are not as good as auto-updating wildcards for changing data, but, I think, they are better than doing all of this by hand.