

Using the Style Sheet plugin in Sibelius

Bob Zawalich November 5, 2015

Sometimes you want to know exactly how a Sibelius score has been put together. You might, for example, want to import a score into a program that does not support MusicXML, or you might just want to document the elements present in a score.

Most of the ingredients of the score, such as text, lines, symbols, and noteheads can be looked at one at a time, but it is hard to get a larger view of what is in a score. There are some plugins like **Calculate Statistics**, **What Is Where?** And **There It Is!** that break out the components, but I wanted a document, or Style Sheet, that would describe the contents of a score.

I wanted to show which text styles, line styles, symbols, and notehead styles were being used in a score. I knew I could also display the main fonts used, and the Document Layout information. There is a limited subset of the Engraving Rules which is available to a plugin, so I could display this as well.

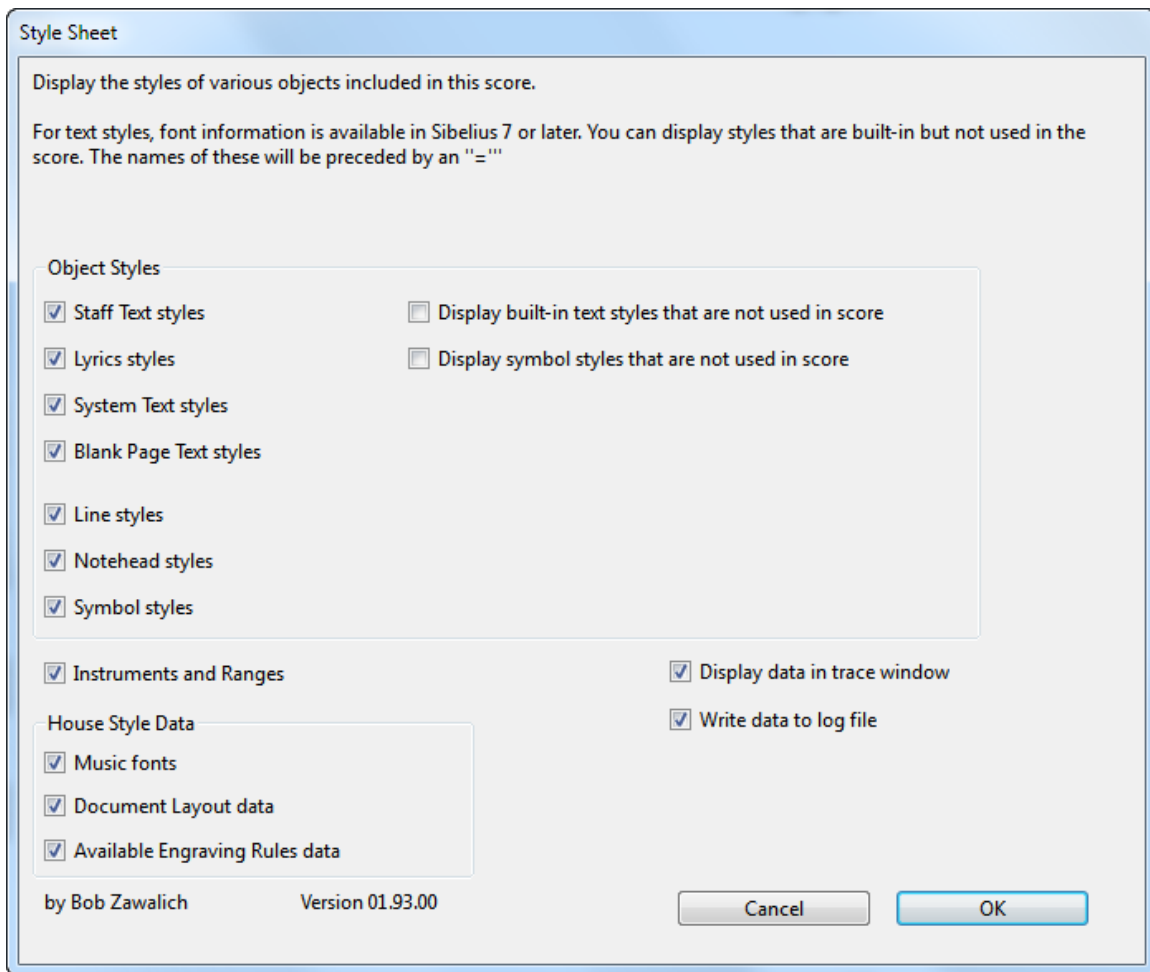
To display text styles rather than individual text objects, I looked at all the text objects in the score, and kept track of the text styles for each of these objects, so I could display them without duplication. I did the same thing for lines, symbols and noteheads.

Some of the most useful data about text styles, namely the font properties, is not available to a plugin. A plugin can see formatting such as a font change which has been applied directly, but cannot access formatting provided by text styles.

While investigating whether this information could be derived by reverse-engineering the Sibelius file (which is not allowed by the license) or by looking in other exported formats such as PDF, a colleague learned that the MusicXML files produced natively by Sibelius 7 and later contained font data for text items from both direct formatting and from the text styles. This is not available in MusicXML generated by the Dolet plugin, and so is not provided if the plugin is run in Sibelius 6.

I took advantage of this by creating a copy of the score and writing out text for each text style used in the score, exporting that to MusicXML, and then analyzing the MusicXML data to get font information for each text style.

This is the dialog displayed by the plugin. Most of the data fields can be omitted if desired:



Here is an example of what the plugin produces for the example score **Early music.sib**:

***Styles in score: C:\Users\Bob\Documents\Scores\Notation\Early music.sib
 ***Text data is of the form: <style name>;<style id>;<X offset>;<Y offset>;;;;
 ***Font data is available in Sibelius 7 and later only.
 *** is normal or italic; is normal or bold. Save text data as a text file and open in a spreadsheet.

***Staff text styles

Small text;text.staff.small;-0.59;4;Times New Roman;7.2;normal;normal
 Technique;text.staff.technique;-0.59;4;Times New Roman;8.5;normal;normal

***Lyrics styles

Lyrics;text.staff.space.hypen.lyrics.verse1;0;-6;Times New Roman;8.5;normal;normal

***System text styles

Composer;text.system.page_aligned.composer;0;0;Times New Roman;7.9;normal;normal
Copyright;text.system.page_aligned.copyright;0;0;Times New Roman;7.2;normal;normal
Header (after first page);text.system.page_aligned.header_notp1;0;0;Times New
Roman;10.0;normal;normal
Instrument name at top left;text.system.page_aligned.instrnametopleft;0;0;Times New
Roman;10.0;normal;normal
Metronome mark;text.system.metronome;-3;0;Times New Roman;8.5;normal;normal
Subtitle;text.system.page_aligned.subtitle;0;0;Times New Roman;10.0;normal;normal
Title;text.system.page_aligned.title;0;0;Times New Roman;15.7;normal;normal

***Blank Page text styles

***Line styles

Bracket above;line.staff.user.0000014
Slur above;line.staff.slur.up

***Notehead styles

Normal;0

***Symbol styles

***Symbol data is of the form: <symbol name>;<index in symbol table>;<relative index
in symbol table row>;<X offset>;<Y offset>

Flat;258;Accidentals 1+2;-0.44;0
Sharp;262;Accidentals 1+6;-0.5;0
Square double whole note;324;Noteheads 1+4;-1.13;0
Whole note;322;Noteheads 1+2;-0.94;0

***Instrument names and ranges

***Instrument data is of the form: <staff number(s)>;<instrument name>;<full
instrument name>;<short instrument name>;<styleId>;<professional high>;<professional
low>;<comfortable high>;<comfortable low>

1;Primus discantus;Primus
discantus;;instrument.singers.soprano.user.0000001;A#3/Bb3(58);C6(84);A#3/Bb3(58);C
6(84)
2;Secundus discantus;Secundus
discantus;;instrument.singers.soprano.user.0000001;A#3/Bb3(58);C6(84);A#3/Bb3(58);C
6(84)
3;Altus;Altus;;instrument.singers.alto.user.0000002;A#2/Bb2(46);A#4/Bb4(70);A#2/Bb2
(46);G4(67)

4;[Voice (5 lines staff) (C-1-G9, C-1-G9)];;;instrument.singers.voice.user.0000003;C-1(0);G9(127);C-1(0);G9(127)
5;Tenor;Tenor;;instrument.singers.tenor.user.0000005;A#2/Bb2(46);A#4/Bb4(70);A#2/Bb2(46);A#4/Bb4(70)
6;Vagans;Vagans;;instrument.singers.baritone.user.0000006;G2(43);F4(65);G2(43);F4(65)
7;Bassus;Bassus;;instrument.singers.bass.user.0000007;E2(40);E4(64);E2(40);E4(64)

***House Style Data

***Music Fonts

Music Text Font Name;Opus Text Std
Main Music Font Name;Opus Std
Main Text Font Name;Times New Roman

***Document Layout Data - values based on units field

Units;inches
Staff Size;0.1969
Above Top Stave Gap;0.3937
Above Top Stave Gap After First Page;0.3937
Below Bottom Stave Gap;0.1969
Below Bottom Stave Gap After First Page;0.1969
First Page Has Unique Vertical Stave Margins;0
Orientation;0
Page Size;4
Page Height;11.6929
Page Width;8.2677
Margin Type;0
Page Bottom Margin;0.7874
Page Left Margin;0.5906
Page Right Margin;0.5906
Page Top Margin;0.5906
Right Page Left Margin;0.5906
Right Page Right Margin;0.5906
Staff Left Margin Full Names;1.0535
Staff Left Margin No Names;0.1077
Staff Left Margin Short Names;0.1077

***Available Engraving Rules Data

Adjust Transposition If Key Sig Wraps;False
Barline Width;0.1563
Beam Thickness;0.5
Cautionary Naturals In Key Signatures;False

Cue Note Scale;75
 Dashed Barline Width;0.1563
 Double Barline Separation;12
 Double Barline Width;0.1563
 Double Tremolo Style;1
 Extra Space Between Groups Of Staves;0
 Extra Spaces Above For System Object Positions;0
 Extra Spaces Below Vocal Staves;0
 Final Barline Separation;12
 Final Barline Width;0.5
 Grace Note Scale;60
 Instrument Names First System;Full
 Instrument Names New Sections;Full
 Instrument Names Subsequent Systems;Short
 Justify Grand Stave Instruments;True
 Justify Multi Stave Instruments;True
 Leger Line Thickness;0.1563
 Respell Remote Keys In Transposing Score;False
 Show Name Of Prevailing Instrument Change At Start Of Systems;True
 Slur Middle Thickness;0.1563
 Slur Outline Width;0.0625
 Small Staff Size Scale;75
 Spaces Between Staves;7
 Spaces Between Systems;9.25
 Staff Justification Percentage;100
 Staff Line Width;0.0938
 Stem Thickness;0.0938
 Tie Middle Thickness;0.1563
 Tie Outline Width;0.0625

Descriptions of the data fields

The data produced by the plugin can be written to the plugin trace window or to a text file, or to both locations. The plugin also has entry points that may be called by other plugins, and these will return the data as an array of text strings.

Because it is difficult for a plugin to format text in columns, the output data is written as a series of text strings whose fields are delimited by semicolons, as in:

```

Composer;text.system.page_aligned.composer;o;o;Times New
Roman;7.9;normal;normal
  
```

To get a formatted report once can produce a text file of the data and read it into a spreadsheet as a semicolon-delimited file. (It would be more typical to produce a comma-delimited file, but some of the standard style names contain commas).

15							
16	***System text styles						
17							
18	Composer	text.system.page_aligned.composer	0	0 Times New Roman	7.9	normal	normal
19	Copyright	text.system.page_aligned.copyright	0	0 Times New Roman	7.2	normal	normal
20	Header (after first page)	text.system.page_aligned.header_notp1	0	0 Times New Roman	10	normal	normal
21	Instrument name at top left	text.system.page_aligned.instrnametopleft	0	0 Times New Roman	10	normal	normal
22	Metronome mark	text.system.metronome	-3	0 Times New Roman	8.5	normal	normal
23	Subtitle	text.system.page_aligned.subtitle	0	0 Times New Roman	10	normal	normal
24	Title	text.system.page_aligned.title	0	0 Times New Roman	15.7	normal	normal

If all the data fields are requested, they will appear in this order:

- Staff text styles
- Lyrics styles
- System text styles
- Blank Page text styles
- Line styles
- Notehead styles
- Symbol styles
- Instrument names and ranges
- Music Fonts
- Document Layout Data
- Available Engraving Rules Data

In the descriptions below, <X offset> and <Y offset> are the X and Y values that are displayed in Sibelius in the Inspector when an object is selected. <styleId> is the internal language-independent identifier Sibelius uses for text and line styles.

- Text style data is of the form: <style name>;[<style id>];<X offset>;<Y offset>;;;;. is normal or italic; is normal or bold.
- Line style data is of the form: <style name>;<style id>.
- Notehead style data is of the form: <style name>;<notehead index in the current score>.
- Symbol data is of the form: <symbol name>;<index in symbol table>;<relative index in symbol table row>;<X offset>;<Y offset>
- Instrument data is of the form: <staff number(s)>;<instrument name>;<full instrument name>;<short instrument name>;<styleId>;<professional high>;<professional low>;<comfortable high>;<comfortable low>

The instrument range data will show the **written** note name with octave, followed by the MIDI number for the pitch in parentheses, such as: E4(64).

Displaying text styles and symbols not used in the score

To get a more complete picture of the contents of a score, it can be useful to show the styles that are defined in the score, but not used anywhere in the score. The difficulty of getting this information is that Manuscript, the plugin language does not provide a way of enumerating the styles in a score.

I employ two different workarounds to get additional data for symbols and text styles.

Symbols are identified and created using an index into the symbol table. To get all the defined symbols, I write symbol number 1, 2, 3... into a bar in the temporary score, and extract data from the written symbols. When the last symbol has been written, trying to write another symbol will fail, so the plugin stops.

No such mechanism exists for text styles, so instead I hand-built a list of all the text styles that shipped with Sibelius 7.5, and I write out text using those text styles, and extract data from the text. This will not, however, find any user-defined text styles that are not used in the score, so it might miss some defined styles. We will display all built-in text styles plus and user-defined text styles that are actually used in the score.

Entry points for calling by other plugins

Plugins may call these entry points so that it can run without displaying a dialog and can return its data in an array of text strings. Other routines (GetSymbol Names and SetUpTempScore, for example) are utility routines. Plugin writers should look at the code of Stylesheet.plg to find the parameters for these routines.

(Routines to call the plugin bypassing the dialog and getting the results in an array)

API_GetDialogSettings
API_SetDialogSettings
API_ProcessScore

(Routines to call the plugin at a later point than ProcessScore, so the caller can set up which score data should be processed. Use at your own risk.)

API_SetUpTempScore (make a clone of the active score with a single staff and no music or text)
API_DoScore

(Library routines to get symbol information)

API_BuildMapSymbolHeaders
API_GetRelativeIndex (English only)

API_GetSymbolNames

What would be nice to provide in the future

Currently there is no way to access this useful data:

- Symbol font and font number information or graphics file name
- Symbol information for each symbol of a composite symbol
- Notehead symbol data
- Additional line data

If it ever becomes available, or I can figure out a way to access it, I will add it to the plugin.