

# New Optional Plugin Features as of Sibelius Ultimate 2020.9

## SupportHalfSemitonePitchValues and TreatSingleCharacterAsString

*Bob Zawalich September 16, 2020*

Sibelius Ultimate 2020.9 added the ability to create notes that are quartertones, and the ability to treat single-character literals as strings. I will discuss these features in separate documents, but this document discusses the fact that these features are optional, and to use them, a plugin must opt in.

If a plugin does not opt in, it behaves exactly as it did in earlier versions of Sibelius. You will not be able to create quartertone notes, and single-character literals will have their traditional peculiar and usually unwanted behaviors. If you call a plugin from another plugin, the called plugin will behave based on whether the options are set in itself, regardless of whether they are set in the calling plugin.

These features are optional because they can cause unwanted behavior in existing plugins, and it was decided that it would be better to require plugins to explicitly opt-in to take advantage of these features, and to allow the programmer to ensure that there are no unwanted side effects, so that existing plugins could continue to function as they did before.

To enable these features in Sibelius 2020.9 or later, you must include special "opt-in" code in **Initialize()**. Plugins that are created using **New** from the **Plugin Editor** will now have opt-in code added automatically. That code looks like this:

```
// The following enables the latest behavior for the Manuscript interpreter.  
// If you intend your plugin to run on previous versions of Sibelius where that functionality  
// didn't exist, you will likely have to revisit the following:
```

```
if (Sibelius.ProgramVersion > 20200600) {  
    SetInterpreterOption(TreatSingleCharacterAsString);  
    SetInterpreterOption(SupportHalfSemitonePitchValues);  
}
```

**SupportHalfSemitonePitchValues** is the quartertone feature. **TreatSingleCharacterAsString** is another new feature that fixes some major buggy behavior involving single-digit literals. 20200600 is the new-format **ProgramVersion** value for version 2020.6.

These features can be activated independently.

I usually avoid putting any significant code in the **Initialize()** call, because Sibelius calls **Initialize()** for every plugin it installs at startup, even if it will not be called, so every plugin that includes this code will slow down startup. I think it is best to include this in **Initialize()**, rather than for example, in **Run()**, because we know **Initialize()** will have been called when another plugin calls a plugin, and the options will have been set if they were intended to be turned on.

You can determine if one of these features is active in a called plugin by calling **GetInterpreterOption** for that plugin, as in these examples:

```
val1 = Retrograde.GetInterpreterOption(TreatSingleCharacterAsString);  
val2 = utils.GetInterpreterOption(SupportHalfSemitonePitchValues);
```

**SetInterpreterOption** can take an optional 2nd parameter, a boolean, that can enable or disable the option. The default is True. To turn the feature off, you might say

```
Retrograde.SetInterpreterOption(TreatSingleCharacterAsString, False);
```

Make sure a called plugin is installed before calling with its name, or Sibelius with crash. The examples above use shipping plugins, which should be safe, but I usually call a routine called **IsPluginAvailable()**, passing in the plugin file name with no path and no extension before calling one. That code is pretty simple:

```
pList = Sibelius.Plugins;  
ok = pList.Contains(pluginFileNameNoExt);
```

The features **SupportHalfSemitonePitchValues** and **TreatSingleCharacterAsString** will be discussed in separate documents.

If plugin is run in a version of Sibelius prior to 2020.9, these features cannot and will not be enabled. If you tested for the version as in the example code in **Initialize()**, then Sibelius will not crash, but you need to be sure that the code will behave correctly with the features disabled, which can be a bit of a challenge to figure out.