

Updating plugins to properly transfer properties for Notes and Chords

Bob Zawalich December 3, 2020

Many plugins need to write notes to scores, and they will often do this when they need to replace notes or chords. This happens when you want to change the duration of a note or chord (a NoteRest object), or just to copy a NoteRest to another location or score when copy and paste does not work. Early plugins written before Transpose was available will do this to change the pitch of a note as well.

When a note is written, it has only a few properties, namely: position, sounding pitch, duration, tied state, voice, diatonic pitch, and string number (for stringed instruments on TAB staves). All other properties must be restored once the Note has been successfully created.

When creating notes, the plugins need to make a list of all of the original Note or NoteRest property fields so they can be restored. If new fields are added to these objects at a later time, like the recent LVTies and TieInto fields, all plugins that use NotePropertyStrings should be updated. They have currently, however, not been updated, and any non-updated fields will be lost when such plugins are used.

Some time ago I came up with a scheme of storing the properties of NoteRests and Note objects as strings, where fields are separated by semicolons (These are referred to as NotePropertyStrings). These strings are then stored in arrays (or arrays of arrays), and the arrays are passed to routines that actually write the NoteRests and Notes.

Dozens of plugins use this mechanism, and they often use the fields in slightly different ways.

Centralizing Note Property code in NotePropertiesLib.plg and utils.plg

Rather than updating each plugin that creates notes every time a change in properties is made, my plan is to create a new library plugin (NotePropertiesLib.plg) that contains standard routines to store and apply property strings. I will edit plugins to replace their internal code with calls to NotePropertiesLib.plg, and "normalize" their use of this resource. This way, when changes are made to Note or NoteRest properties, NotePropertiesLib.plg can be updated, and for the most part, plugins that call these routines will just need to get an updated NotePropertiesLib.plg file.

This means that the users will need to install NotePropertiesLib.plg in order to run any plugin that calls into it. This will only need to happen once for any given user, though it should be updated when it changes, which I hope will not be very often.

For the most part, my intention is to update plugins to use these standard property routines, and at the same time update them to properly handle quartertones in Sibelius Ultimate 2020.6 or later, which was the first version of Sibelius that allowed plugins to create quartertones.

Well over a hundred plugins will likely need to be updated to bring them into compliance.

I have made similar routines available to shipping plugins, and have updated some of them to support these functions. I hope they will be included in a future release of Sibelius Ultimate.

Note and NoteRest properties that are now supported.

Here are lists of the current properties that will be transferred. NoteRest properties apply to the "chord", which contains Notes. Note Properties are though that are unique to the individual notes. There is a bit of overlap (voice is in both sets). I show "newer" properties that may be missed in **bold**.

Color has been a NoteRest property for a long time, but has only been independent in Notes since Sibelius 8.3, and no published plugin I am aware of has successfully transferred color into Notes until the work I am doing

now. Some properties will only be transferred into Notes if the plugin is run in a version of Sibelius that supports the properties.

```
g_arrNoteRestPropertyNames
{
"NoteCount"
"StaffNum"
"BarNumber"
"Position"
"Duration"
"VoiceNumber"
"Beam"
"SingleTremolos"
"DoubleTremolos"
"CueSize"
"Hidden"
"Articulations"
"FeatheredBeamType"
"HasStemlet"
"StemletType"
"RestPosition"
"ArpeggioType"
"ArpeggioDx"
"ArpeggioTopDy"
"ArpeggioBottomDy"
"ArpeggioHidden"
"FallType"
"FallDx"
"ScoopType"
"ScoopDx"
"HasCustomDrawOrder"
"DrawOrder"
"UsesMagneticLayoutSettingOverridden"
"UsesMagneticLayout"
"StemFlipped"
"PlayOnPass"
"Color"
"ColorAlpha"
}
```

```
g_arrNotePropertyNames
{
"VoiceNumber"
"Name"
"Pitch"
"DiatonicPitch"
"Tied"
"NoteStyle"
"UseOriginalVelocityForPlayback"
"OriginalVelocity"
"UseOriginalDeltaSrForPlayback"
"OriginalDeltaSr"
"UseOriginalDurationForPlayback"
"OriginalDuration"
"AccidentalStyle"
"Bracketed"
```

```

"Slide"
"StringNum"
"Color"
"ColorAlpha"
"SlideStyleId"
"TiedInto"
"LvTie"
}

```

The list of unsupported properties varies by plugin, with older plugins that were not updated missing the most. There were a fair number of plugins written in Sibelius 3 and 4 which are still in use, and they had no knowledge of many of these properties.

Here is an example of properties now transferred by an updated Copy Double Notes Values plugin.

The image displays three staves of musical notation illustrating the transfer of properties from an original score to an updated plugin. The first staff, labeled 'Original', shows a sequence of notes with various colors (green, blue, pink, yellow) and stems. A callout bubble indicates that 'Note and NoteRest Properties transferred after update to Copy Double Note Values'. The second staff, 'Copy Double NV - original', shows the same notes but with some properties missing, indicated by red boxes around the notes. The third staff, 'Copy Double NV - updated', shows the notes with the missing properties restored, also indicated by red boxes. A callout bubble points to a 'Play on pass' symbol, stating 'Play on pass now transferred'. Below the first staff, a list of properties is provided: 'Note props: color, slide id, lv tie, tie into, accidental style - also nr - arpeggio offsets varied durations stem flipped, Artic, Play on pass'.

Note and NoteRest Properties transferred after update to Copy Double Note Values

Note props: color, slide id, lv tie, tie into, accidental style - also nr - arpeggio offsets varied durations stem flipped, Artic, Play on pass

Copy Double NV - original

Copy Double NV - updated

Play on pass now transferred