

Tutorial: Execute Commands and cmdutils

Changing Text Formatting

Bob Zawalich June 6, 2021

You can use several commands in **cmdutils.plg** to change the format of selected text.

SetTextFormat_BoldItalic_cu()
SetTextFormat_Bold_cu()
SetTextFormat_Italic_cu()
SetTextFormat_Normal_cu()
SetTextFormat_Underlined_cu()

SetTextFormat_Position_Subscript_cu()
SetTextFormat_Position_Superscript_cu()

- Sets Text formatting properties on any selected Text, SystemTextItem, or LyricItem objects.
- No way to remove individual properties, but can reset formatting to normal and then adding back the desired properties

BracketText_cu()

- Puts parentheses () around all selected text objects.
- Requires the **Bracket Text** plugin (version 01.94.00 or later).

Here is a quick example of how the **SetTextFormat** commands work.

Let's say we have a text object with the text "Here is a quick example of how this works.";

If we select it and run `SetTextFormat_Bold_cu()`, the formatted text looks like this onscreen

Here is a quick example of how this works.

But its formatted text looks like this. There is a `\B\` wildcard at the start, and sort of like HTML code, that turns Bold formatting on and does not turn it off.

`\B\`Here is a quick example of how this works.

If I had the same text selected and added italics and subscript to it, it would appear as

The image shows a screenshot of a music score editor. At the top, the text "Here is a quick example of how this works." is displayed in a blue, bold, italicized font, enclosed in a dashed selection box. Above this text is a large, bold, black letter 'C' inside a square box. To the left of the 'C' is a large number '3', and below it is a smaller number '12'. Below the text and 'C' is a musical staff with a treble clef, a key signature of one flat (labeled 'Picc.'), and a time signature of 12/8. A dashed arrow points from the text to the staff. The staff contains a single note on the second line.

And the formatted text would internally look like

`\subscript\\B\`Here is a quick example of how this works.

The commands just add a new wildcard to the front of the string.

If you did not want subscript anymore (and were not just fixing it in the Inspector), you only option would be to clear the formatting, then add back bold and italic, as in

```
SetTextFormat_Normal_cu()  
SetTextFormat_Bold_cu()  
SetTextFormat_Italic_cu()
```

Which leaves the formatted text as

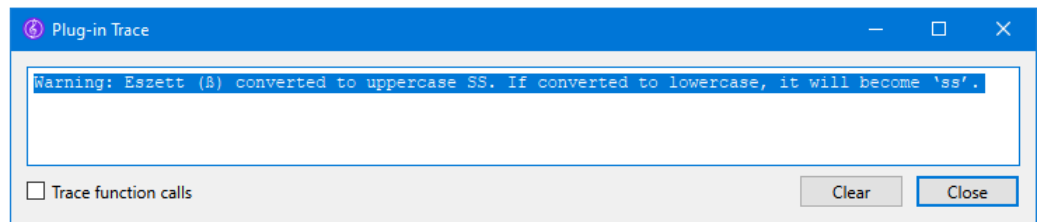
`\B`Here is a quick example of how this works.

You could take any of these commands, add them to the Command List by themselves, and use New Plugins to create and install a plugin that can run the command, and then apply a keyboard shortcut to the plugin.

I know of no other way to change the formatting of a selected piece of text that is not being edited without using a mouse, or the somewhat arcane Ribbon shortcut for the text styles in the Text pane (Alt+T+FA+B+Enter on my Windows 10 machine.).

If you install the **Change Case** plugin it will also install the plugin **makechangepcase**, written by Andrew Moschou. If that plugin is installed, these commands will use it to convert things like accented characters correctly. Here are some crude examples. If you convert the character β to uppercase, it turned to SS and you will get a warning from the **makechangepcase** plugin.

<p>Original</p> <p>á ä â ç è é ê ë ì í î ï ò ó ô õ ù ú û ü</p> <p>À Á Ç È É Ê Ë Ì Í Î Ï Ò Ó Ô Õ Ù Ú Û Ü β</p> 	<p>To Upper Case</p> <p>Á Ä Â Ç È É Ê Ë Ì Í Î Ï Ò Ó Ô Õ Ù Ú Û Ü</p> <p>À Á Ç È É Ê Ë Ì Í Î Ï Ò Ó Ô Õ Ù Ú Û Ü SS</p> 
<p>To Lower Case</p> <p>á ä â ç è é ê ë ì í î ï ò ó ô õ ù ú û ü</p> <p>à á ç è é í î ï ò ó ù ú ü β</p> 	<p>Toggle Case</p> <p>À Á Â Ç È É Ê Ë Ì Í Î Ï Ò Ó Ô Õ Ù Ú Û Ü</p> <p>à á ç è é í î ï ò ó ù ú ü SS</p> 



Otherwise, it should pretty much work as expected.