

Tutorial on using *RunCommandIfCondition* commands in cmdutils and Execute Commands

Bob Zawalich July 17, 2024 updated September 12, 2024

Table of contents:

Tutorial on using <i>RunCommandIfCondition</i> commands in cmdutils and Execute Commands	1
Overview	1
Case study: If the selection doesn't contain tuples, perform this action. If it contains a tuple, quit	1
Case study: Run a command, then exit, but warn the user before exiting	4

Overview

This document describes one case in which the cmdutils ***RunCommandIfCondition*** commands can be used to change the flow of control in an **Execute Commands** macro based on the evaluation of a “condition” defined in the plugin **Evaluate Plugin Condition**.

Case study: If the selection doesn't contain tuples, perform this action. If it contains a tuple, quit

There are 2 things you need to do, **find a condition**, and **find an appropriate cmdutils command**.

To find the condition run the plugin **Evaluate Plugin Condition** and see if there is an appropriate condition in the list. In this case it will be **tuples_selected**. If you press **Trace List**, it will trace all the commands, and you can **find it in the trace window and copy it**. You will want this copied condition name later.

This plugin is intended to be called by the cmdutils "IfCondition" routines. You can test whether a condition is working as expected by running this plugin directly; it will put up a message box that shows how the condition was evaluated.

Most conditions will act on Sibelius.ActiveScore.Selection.

```

bottom_staff_selected
top_staff_selected
first_bar_in_score_selected
last_bar_in_score_selected
one_staff_only_selected
-----
passage_selection_bars_fully_selected
-----
selection_contains_only_all_staves_of_grand_staff
selection_contains_bottom_staff_of_grand_staff
selection_contains_only_staves_in_grand_staff
selection_contains_top_staff_of_grand_staff
-----
selection_is_empty
selection_is_empty_system_ok
selection_is_passage
selection_is_system_passage
-----
notes_selected
notes_or_rests_selected
bar_rests_selected
rests_selected
rests_or_bar_rests_selected
tuplets_selected
tuplets_or_child_notes_selected
tuplets_or_child_notes_or_rests_selected
tuplets_or_child_rests_selected
-----
voice_1_objects_selected
voice_2_objects_selected
voice_3_objects_selected
voice_4_objects_selected
    
```

Evaluate Condition

Trace Selected Condition

Trace List

Close

<-- The "-----" entries in the list box are non-functioning visual separators and should not be selected.

But wait, you say. I want to run the command if the selection does ***not*** contain tuplets.

There is only a single form of each condition. It will evaluate to True or False. **“tuplets_selected”** will return True if the selection contains tuplets, and False if not.

Each **RunCommandIf... command has both a True and a False form**. Here you need to decide which form to use. Since we want to run a command and exit based on the condition, the cmdutils command will be one of these:

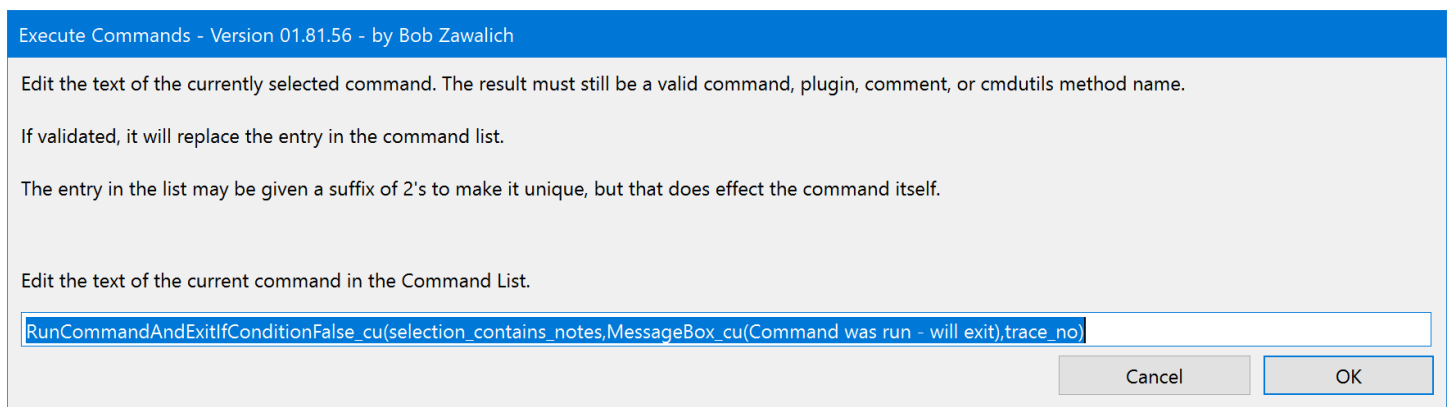
RunCommandAndExitIfCondition**True**_cu(notes_selected,MessageBox_cu(Command was run - will exit),trace_no)

RunCommandAndExitIfConditionFalse_cu(notes_selected,MessageBox_cu(Command was run - will exit),trace_no)

The condition “**tuplets_selected**” will return True if the selection contains tuplets, and False if not. **We want to run a command when the condition returns False, so we want the False form of the command.**

RunCommandAndExitIfConditionFalse_cu(notes_selected,MessageBox_cu(Command was run - will exit),trace_no).

Go to **Execute Commands**, and find **RunCommandAndExitIfConditionFalse_cu** in the **Cmdutils Exit Plugin** category and add it to the Command List. Then press **Edit Command**, and you will see



RunCommandAndExitIfConditionFalse_cu(notes_selected,MessageBox_cu(Command was run - will exit),trace_no)

These commands have the form **<IfConditionCommand>(<condition>, <command>, <trace_YesNo>)**. **Execute Commands** puts **placeholder** values in the commands that you will usually need to change.

- **notes_selected** is the placeholder condition. You want to replace it with **tuplets_selected**, which you copied earlier. Remember?
- **MessageBox_cu(Command was run - will exit)** is the placeholder command to be run. Replace this with the command you want to run. You only get a single command, but it could be a plugin, a Sibelius command, or a cmdutils command. Not a macro though. Let's say that you want to clear the selection before exiting. You could run the command “**select_none**”.
- <trace_YesNo> will always be “**trace_no**” unless you are debugging , where “**trace_yes**” will trace some information about the state of things. This may go away eventually but it is useful now, for me at least.

In the end you will have the command

RunCommandAndExitIfConditionFalse_cu(**tuplets_selected**,**select_none**,trace_no)

Choose OK on the **Edit Command** box, and the updated command will appear in the **Command List**, and it should do what you asked it to do.

This seems like a lot of work, but there is a lot of power here.

These commands are very fussy. You must spell the condition and command exactly correctly, and I recommend not adding spaces after the commas, though in the case of these commands the extra spaces are skipped. But it is a good programmer habit to follow the syntax exactly, so I recommend no spaces here.

Case study: Run a command, then exit, but warn the user before exiting

In this example we want to **clear the selection and exit** if there are any notes selected, but we would like to **warn the user** what was happening before exiting.

The following command will run the command **select_none**, which clears the selection if the condition **notes_selected** evaluates to **True**.

```
RunCommandAndExitIfConditionTrue_cu(notes_selected,select_none,trace_no)
```

This accomplishes everything except the warning. One way to add a warning would be to first run a command that will show a message box but **not exit** if the same condition evaluates to true.

The following command will do that. It uses the non-exiting command **RunCommandIfConditionTrue_cu** with the condition **notes_selected** and runs the command **MessageBox_cu()** which will display a message, such as:

```
RunCommandIfConditionTrue_cu(notes_selected,MessageBox_cu(Stopping because notes are selected...This plugin will exit and the selection will be cleared.),trace_no)
```

To display a warning, then clear the selection, run the **warning** command first, then the **clear selection and exit** command immediately:

```
RunCommandIfConditionTrue_cu(notes_selected,MessageBox_cu(Stopping because notes are selected...This plugin will exit and the selection will be cleared.),trace_no)
RunCommandAndExitIfConditionTrue_cu(notes_selected,select_none,trace_no)
```

This basic principle can be used in many situations. Any time you want to run **RunCommandAndExitIfConditionTrue/False_cu** and want to include a warning before exiting, **run the non-exit form of the command first, and have the command be MessageBox_cu.**