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

Bob Zawalich July 17, 2024 updated August 10, 2025

Table of contents:

Overview

Macros and plugins generated by the plugin **Execute Commands** are typically a sequence of Sibelius commands, plugins, or cmdutils commands, that run without interruption. While there are no **if-then-else** instructions, there is a set of cmdutils commands in the category **CmdutilsExitPlugin** that can be used to alter flow of control.

As of August 10, 2025, these are the available commands, which fall into the sub-categories:

- ContinueIf
- ExitIf (fixed condition)
- ExitOrSelectAll<fixed condition>
- ExitPlugin
- IfCondition
    - ExitIfCondition[True/False]<condition>
    - RunCommand1IfConditionFalseElseCommand2
    - RunCommand1IfConditionFalseElseCommand2
    - RunCommand1IfConditionTrueElseCommand2
    - RunCommandAndExitIfConditionFalse
    - RunCommandAndExitIfConditionTrue
    - RunCommandIfConditionFalse
    - RunCommandIfConditionTrue

Here is the full set of CmdutilsExitPlugin commands

```
ContinueIfSelection_Empty_cu(The selection is not empty. This plugin will now exit.)
ContinueIfSelection_Empty_YesNo_cu(The selection is not empty.)
ContinueIfSelection_NotEmpty_cu(Nothing is selected. This plugin will now exit.)
ContinueIfSelection_NotEmpty_YesNo_cu(Nothing is selected.)


ExitIfPlugin_Unavailable_cu(Resize Bar)
ExitIfSelection_Avoid_BottomStaff_cu(The selection may not include the bottom staff in the score. This plugin will now exit.)
ExitIfSelection_Avoid_FirstBar_cu(The selection may not include the first bar in the score. This plugin will now exit.)
ExitIfSelection_Avoid_GrandStaff_Bottom_cu(The bottom staff of the selection may not be the bottom staff of a multi-staff instrument.
This plugin will now exit.)
```

```
ExitIfSelection_Avoid_GrandStaff_Top_cu(The top staff of the selection may not be the top staff of a multi-staff instrument. This plugin will now exit.)
ExitIfSelection_Avoid_LastBar_cu(The selection may not include the last bar in the score. This plugin will now exit.)
ExitIfSelection_Avoid_TopStaff_cu(The selection may not include the top staff in the score. This plugin will now exit.)
ExitIfSelection_Empty_cu(Nothing is selected. This plugin will now exit.)
ExitIfSelection_Empty_SystemOK_cu(Nothing is selected. This plugin will now exit.)
ExitIfSelection_Needs_FullSelect_cu(The selection must have all bars fully selected. This plugin will now exit.)
ExitIfSelection_Needs_GrandStaff_All_cu(The selection must include all the staves of a multi-staff instrument including ossias. This plugin will now exit.)
ExitIfSelection_Needs_GrandStaff_Any_cu(The selection must include only staves of a single multi-staff instrument including ossias. This plugin will now exit.)
ExitIfSelection_Needs_OneStaff_cu(The selection must include only a single staff. This plugin will now exit.)
ExitIfSelection_NotEmpty_cu(The selection is not empty. This plugin will now exit.)
ExitIfSelection_NotPassage_cu(A passage selection is required. This plugin will now exit.)
ExitIfSibeliusVersion_LessThan_cu(2021.9.0)

ExitOrAll_Selection_Empty_cu(Nothing is selected. Choose Yes to select all and continue or No to exit.)
ExitOrAll_Selection_Empty_SystemOK_cu(Nothing is selected. Choose Yes to select all and continue or No to exit.)
ExitOrAll_Selection_NotPassage_cu(A passage selection is required. Reply Yes to select all and continue or No to exit.)

ExitPlugin_cu()


ExitIfConditionFalse_cu(tuplet_objects_selected,The selection does not contain tuplets. This plugin will now exit.)
ExitIfConditionTrue_cu(tuplet_objects_selected,The selection contains tuplets. This plugin will now exit.)

RunCommand1IfConditionFalseElseCommand2_cu(notes_selected,MessageBox_cu(Command1 was run),MessageBox_cu(Command2 was run),trace_no)

RunCommand1IfConditionTrueElseCommand2_cu(notes_selected,MessageBox_cu(Command1 was run),MessageBox_cu(Command2 was run),trace_no)

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

RunCommandIfConditionFalse_cu(notes_selected,MessageBox_cu(Command was run),trace_no)
RunCommandIfConditionTrue_cu(notes_selected,MessageBox_cu(Command was run),trace_no)

MessageBoxYesNo_Exit_No_cu(Choose No to exit this plugin or macro)
MessageBoxYesNo_Exit_Yes_cu(Choose Yes to exit this plugin or macro)
```

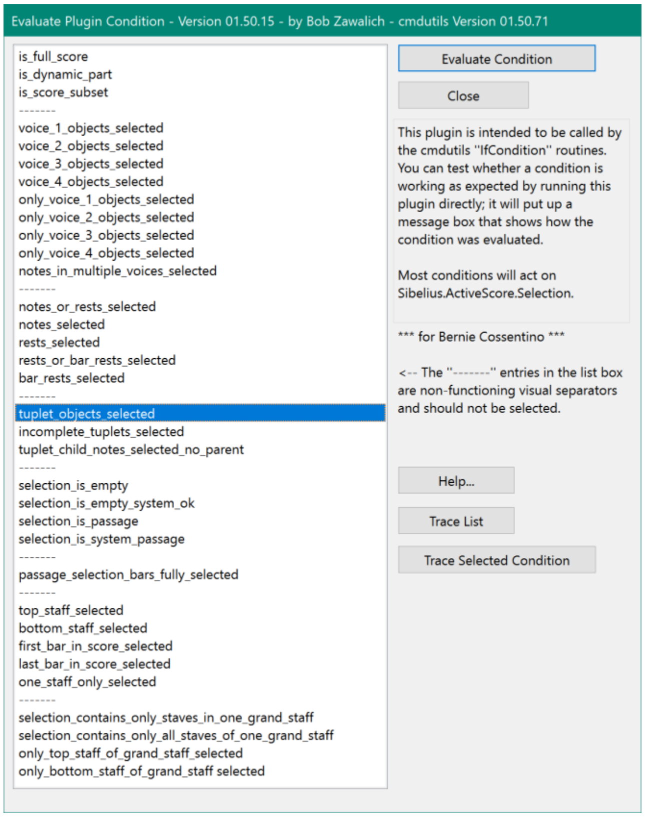
The **IfCondition** commands allow you to specify a condition, chosen from a limited set of conditions defined in the plugin **Evaluate Plugin Condition**, which must be installed to use these commands. They decide to exit or continue based on whether the condition evaluates to True or False, or to run additional commands depending on how the condition evaluates. See **If condition commands in cmdutils and Execute Commands** for more details. **Evaluate Plugin Condition** can be modified by a ManuScript programmer to add additional conditions.

This document describes one case in which the cmdutils ***RunCommandIfCondition*** command 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 selection contains no tuplets, perform this action. If it contains a tuplet, 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 these is an appropriate condition in the list. In this case it will be **tuplets_objects_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.



Evaluate Plugin Condition - Version 01.50.15 - by Bob Zawalich - cmdutils Version 01.50.71

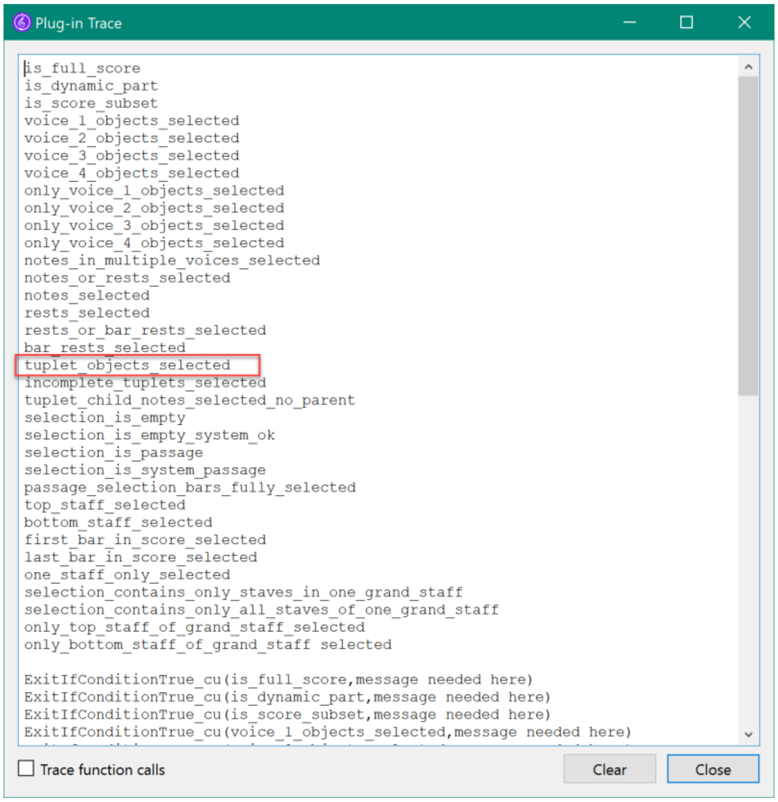
```
is_full_score
is_dynamic_part
is_score_subset
-------
voice_1_objects_selected
voice_2_objects_selected
voice_3_objects_selected
voice_4_objects_selected
only_voice_1_objects_selected
only_voice_2_objects_selected
only_voice_3_objects_selected
only_voice_4_objects_selected
notes_in_multiple_voices_selected
-------
notes_or_rests_selected
notes_selected
rests_selected
rests_or_bar_rests_selected
bar_rests_selected
-------
tuplet_objects_selected
incomplete_tuplets_selected
tuplet_child_notes_selected_no_parent
-------
selection_is_empty
selection_is_empty_system_ok
selection_is_passage
selection_is_system_passage
-------
passage_selection_bars_fully_selected
-------
top_staff_selected
bottom_staff_selected
first_bar_in_score_selected
last_bar_in_score_selected
one_staff_only_selected
-------
selection_contains_only_staves_in_one_grand_staff
selection_contains_only_all_staves_of_one_grand_staff
only_top_staff_of_grand_staff_selected
only_bottom_staff_of_grand_staff selected
```

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.

*** for Bernie Cossentino ***

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



Plug-in Trace

```
is_full_score
is_dynamic_part
is_score_subset
voice_1_objects_selected
voice_2_objects_selected
voice_3_objects_selected
voice_4_objects_selected
only_voice_1_objects_selected
only_voice_2_objects_selected
only_voice_3_objects_selected
only_voice_4_objects_selected
notes_in_multiple_voices_selected
notes_or_rests_selected
notes_selected
rests_selected
rests_or_bar_rests_selected
bar_rests_selected
tuplet_objects_selected
incomplete_tuplets_selected
tuplet_child_notes_selected_no_parent
selection_is_empty
selection_is_empty_system_ok
selection_is_passage
selection_is_system_passage
passage_selection_bars_fully_selected
top_staff_selected
bottom_staff_selected
first_bar_in_score_selected
last_bar_in_score_selected
one_staff_only_selected
selection_contains_only_staves_in_one_grand_staff
selection_contains_only_all_staves_of_one_grand_staff
only_top_staff_of_grand_staff_selected
only_bottom_staff_of_grand_staff selected

ExitIfConditionTrue_cu(is_full_score,message needed here)
ExitIfConditionTrue_cu(is_dynamic_part,message needed here)
ExitIfConditionTrue_cu(is_score_subset,message needed here)
ExitIfConditionTrue_cu(voice_1_objects_selected,message needed here)
```

☐ Trace function calls

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_objects_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:

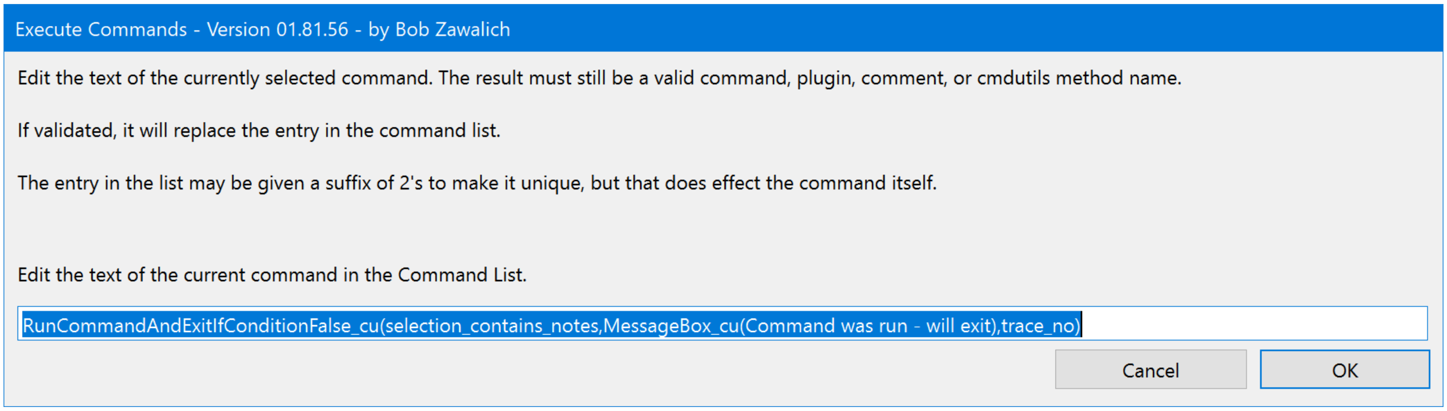
RunCommandAndExitIfCondition**True**_cu(notes_selected,MessageBox_cu(Command was run - will exit),trace_no)

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

Since the condition **tuplets_objects_selected** returns 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.**

RunCommandAndExitIfCondition**False**_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



Execute Commands - Version 01.81.56 - by Bob Zawalich

Edit the text of the currently selected command. The result must still be a valid command, plugin, comment, or cmdutils method name.

If validated, it will replace the entry in the command list.

The entry in the list may be given a suffix of 2's to make it unique, but that does effect the command itself.

Edit the text of the current command in the Command List.

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

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_objects_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_objects_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 is 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**.