WinActor Note
Text Processing Scenario
Creation Manual

NTT ADVANCED TECHNOLOGY CORPORATION

Contents

1. Introduction 1
   1.1. Trademarks 3
   1.2. Notes on this manual 3
2. Launching WinActor Note 4
3. Scenario creation tutorial 5
   3.1. Preparation 5
   3.2. Loading a file and moving the cursor 6
      3.2.1. Loading a text file 7
      3.2.2. Recommended character encoding 10
      3.2.3. Cursor operation 11
   3.3. Processing texts 16
      3.3.1. Determining which information to be extracted 17
      3.3.2. Deleting lines that do not contain the information to be extracted 18
      3.3.3. Editing a text in a line 23
      3.3.4. Selecting a line away from the current position 26
      3.3.5. Splitting into multiple lines 28
      3.3.6. Removing leading and trailing white spaces 30
      3.3.7. Review of moving the cursor and editing a text in a line 32
      3.3.8. Extracting information enclosed in parentheses 36
      3.3.9. Practices 37
   3.4. Recording and editing a macro 39
      3.4.1. Starting the macro recording 40
      3.4.2. Recording the text processing operations 41
      3.4.3. Saving the created macro 42
      3.4.4. Editing the created macro 44
      3.4.5. Adding a macro loop 46
   3.5. Creating a WinActor scenario 53
      3.5.1. Creating a new WinActor scenario 53
      3.5.2. Reading texts by the instruction from WinActor 54
      3.5.3. Processing texts by the instruction from WinActor 57
      3.5.4. Moving the cursor to TOP by the instruction from WinActor 60
      3.5.5. Reading information from WinActor Note 63
      3.5.6. Practice 66
      3.5.7. Checking the operations for reading information into variables 68

3.6. Operation mode 70
3.6.1. Hidden 70
3.6.2. View 70
3.6.3. Edit 70
3.6.4. Precautions when running a scenario 70
4. Library and property list 71
4.1. Note_ReadBlockLoop 71
4.2. Note_MoveCursorAndReadText 72
4.3. Note_MoveCursor 72
4.4. Note_InsertTextFromClipboard 74
4.4.1. Example 1 75
4.4.2. Example 2 76
4.5. Note_PasteTextFromClipboard 77
4.6. Note_CopyToClipboard 77
4.7. Note_Sort 77
4.8. Note_TabTool 78
4.9. Note_WriteTextFile 79
4.10. Note_ReadTextFile 79
4.11. Note_InsertTextFromFile 79
4.11.1. Example 1 81
4.11.2. Example 2 82
4.12. Note_BlockSplitTool 83
4.12.1. Example 1 83
4.12.2. Example 2 83
4.13. Note_BlockExtractionTool 84
4.14. Note_BlockSearchTool 84
4.15. Note_BlockNumberDesignationTool 84
4.16. Note_BlockMergeTool 85
4.17. Note_ReadAndRunMacro 86
4.18. Note_MarkTool 86
4.19. Note_ChangeMode 86
4.20. Note_ReplaceVariablePlaceholder 87
4.20.1. Example 88
4.21. Note_TextToVariable 89
4.22. Note_SubstringToVariable 89
4.23. Note_ReadStatus 90

4.24. Note_WhiteSpaceRemovalTool 91
4.25. Note_EditTool 91
4.26. Note_ReplaceText 94
4.27. Note_SetWorkFolder 94
5. Docking window 95
6. Reference materials 96

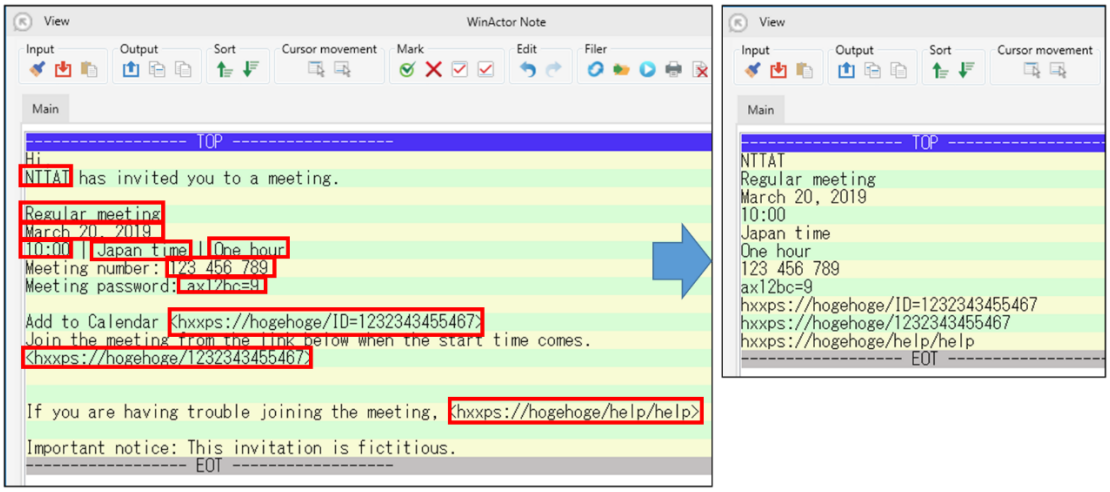
1. Introduction

WinActor Note is a text edit tool that comes with WinActor.

WinActor Note is a tool for processing "atypical texts" such as email texts into "typical texts" that are easily linked with other systems.

The left side of the figure below shows the email texts of the meeting invitation. Detailed information about the meeting is embedded in the texts. According to the texts on the left side of the figure below, the inviter of the meeting is "NTTAT." If you want to send the invitee's "NTTAT" as information to another system for system linkage, the sentence "has invited you to a meeting." will be the extra information.

The right side of the figure below shows the result of removing such unnecessary texts from the invitation texts and extracting only the information necessary for system linkage. If the information is processed into such a format, it becomes easier to create a system linkage scenario, for example, a scenario to transfer texts to the system one line at a time.



Atypical texts　　Typical texts

Figure 1-1. Information suitable for system linkage

WinActor Note has various functions to process "atypical texts" into "typical texts."

Through the tutorial in this manual, you can learn how to process texts using WinActor Note and how WinActor Note works with WinActor.

The steps to create a scenario and the tutorial corresponding to each step are as follows.

① Determine which information of "atypical texts" is required for system linkage.

→ 3.3.1Determining which information to be extracted

② Consider a procedure for processing "atypical texts" into "typical texts."

* Procedure for deleting information/text unnecessary for system linkage

→ 3.3Processing texts

③ Record the processing procedure as a WinActor Note macro.

→ 3.4Recording and editing a macro

④ Create a WinActor scenario.

→ 3.5 Creating a WinActor scenario

1.1. Trademarks

The names described below and other names of companies and products in this document are trademarks or registered trademarks of their respective companies. The TM, ®, and © marks are omitted in this document.

・Windows and Windows PowerShell are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries.
* The official name of Windows is Microsoft Windows Operating System.

・The name of CentOS is a trademark or registered trademark of CentOS ltd.
・Mac OS and OS X are trademarks of Apple Inc.
・Linux is a trademark or registered trademark of Mr. Linus Torvalds in Japan and other countries.

1.2. Notes on this manual

・ The copyright notice "Copyright © 2013-2025 NTT, Inc. & NTT ADVANCED TECHNOLOGY CORPORATION" attached to this manual and the provided software cannot be changed or deleted.
The copyright of this manual belongs to NTT, Inc. and NTT ADVANCED TECHNOLOGY CORPORATION.
・The descriptions in this manual assume that users understand Windows operations and functions. For information that is not described in this manual, see the documents provided by Microsoft.

2. Launching WinActor Note

Click "WinActor Note" in "Tool" of WinActor.

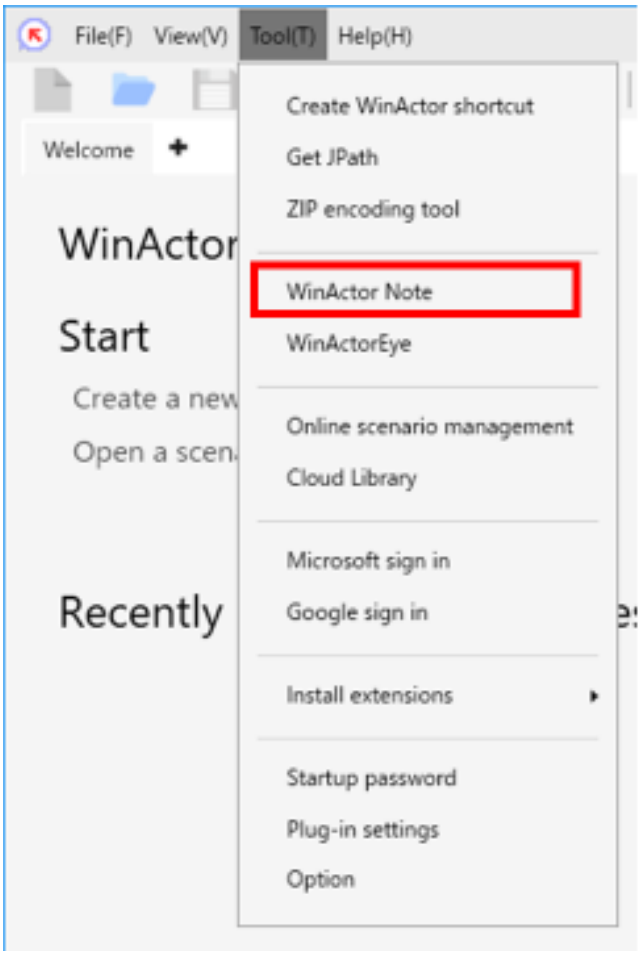


Figure 2-1. Launching WinActor Note

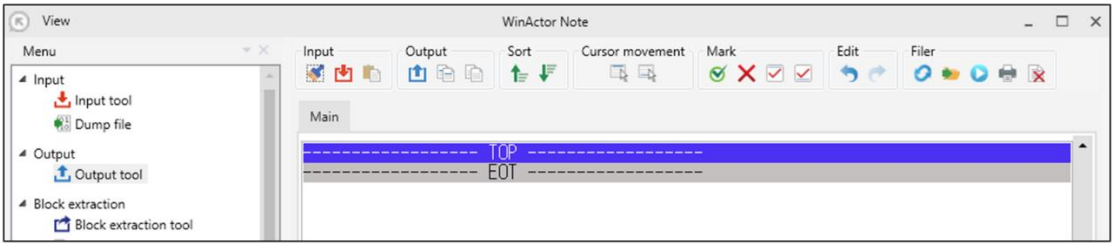


Figure 2-2. WinActor Note window

3. Scenario creation tutorial

3.1. Preparation

Create a folder for the tutorial and save the following texts to a file named "Meeting_request.txt."

```
Hi,
NTTAT has invited you to a meeting.

Regular meeting
March 20, 2019
10:00 | Japan time | One hour
Meeting number: 123 456 789
Meeting password: ax12bc=9

Add to Calendar <hxxps://hogehoge/ID=1232343455467>
Join the meeting from the link below when the start time comes.
<hxxps://hogehoge/1232343455467>


If you are having trouble joining the meeting, <hxxps://hogehoge/help/help>

Important notice: This invitation is fictitious.
```

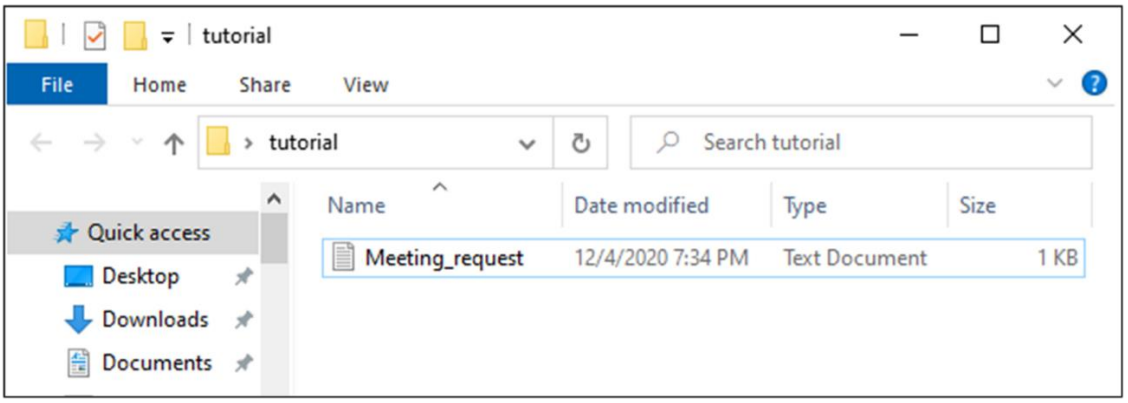


Figure 3-1. Preparing the folder for the tutorial

3.2. Loading a file and moving the cursor

The tutorial "Loading a file and moving the cursor" describes how to load a text file into WinActor Note and move the cursor on WinActor Note.

3.2.1. Loading a text file

Load a text file into WinActor Note.

Open "Input tool" from the "Input" menu.

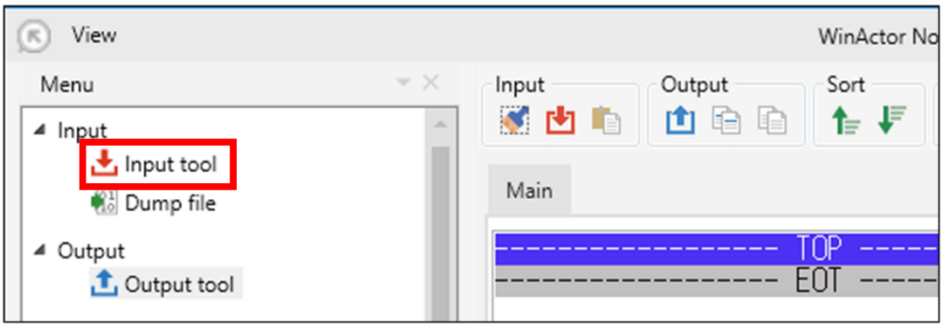


Figure 3-2. Opening "Input tool"

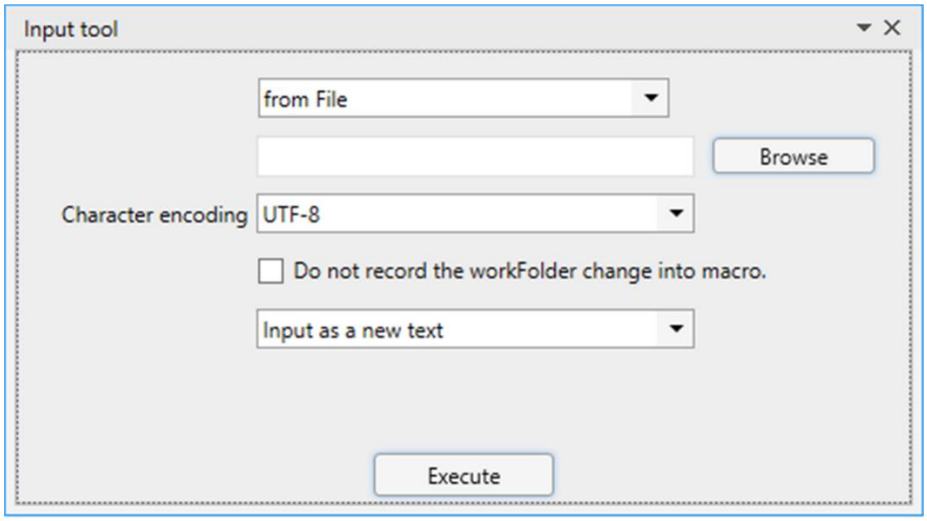


Figure 3-3. Input tool

First, select "from File." Then, click ‘Browse’ and select "Meeting_request.txt."
Select "Character encoding" according to the text file saving format.

* If the text file is saved in ShiftJIS format, select "MS932 (ShiftJIS)" for the character encoding.
* When opening a text file saved in ANSI format on Japanese Windows, select "MS932 (ShiftJIS)" for the character encoding.

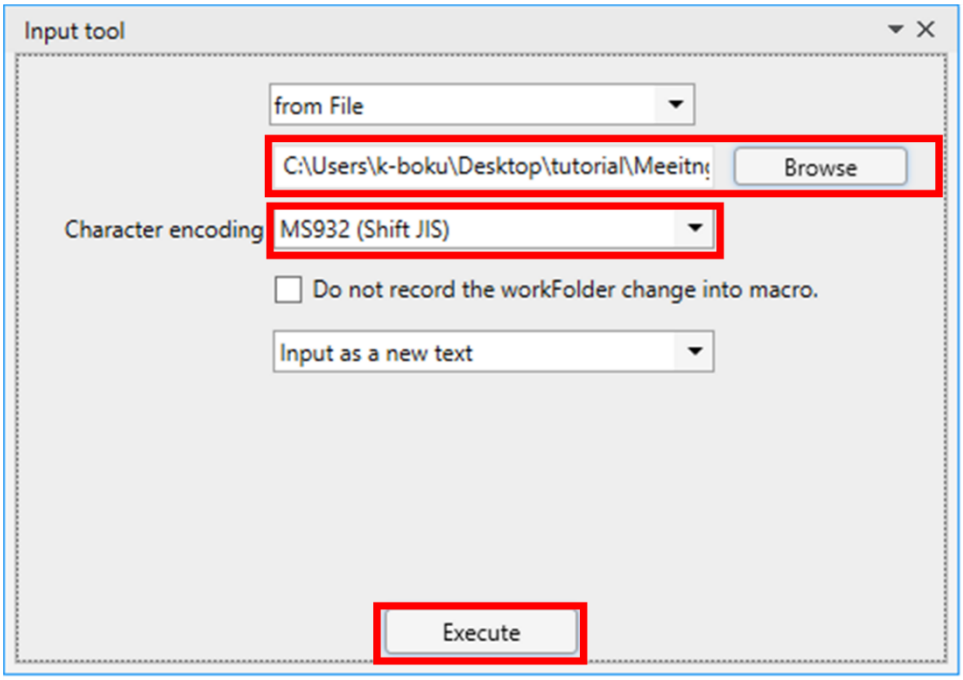


Figure 3-4. Loading the file

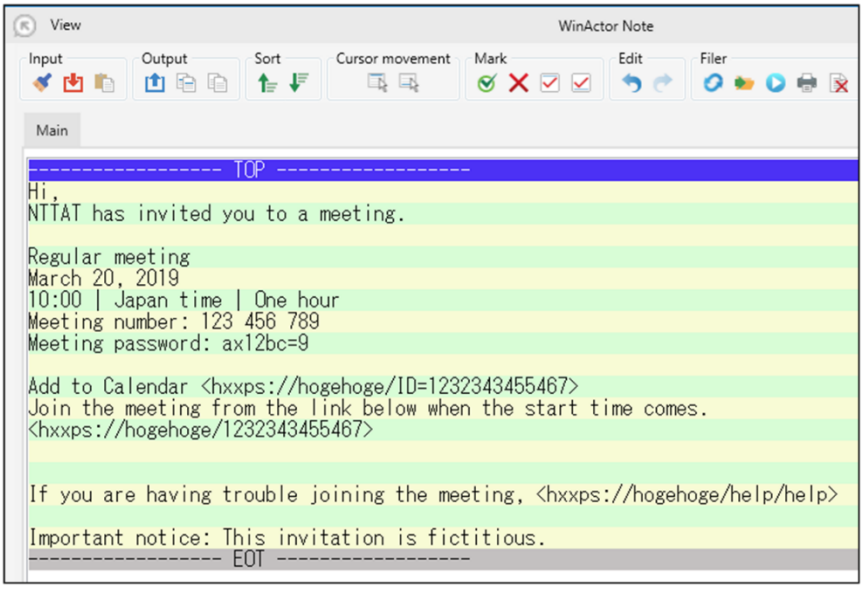
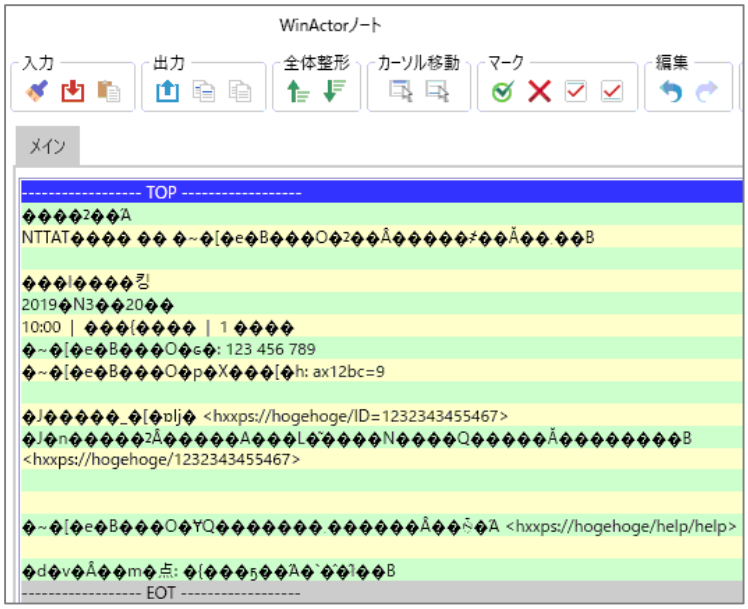


Figure 3-5. Loaded texts



The characters may be garbled if the character encoding is incorrectly selected.

Figure 3-6. Loaded texts (garbled)

3.2.2. Recommended character encoding

The character encoding "UTF-8," which is not affected by the OS settings and is supported by many editors, is recommended.

You can select "Encoding" when saving a text file with Notepad.
For "ANSI," which is selected by default in Notepad, the character encoding changes depending on the OS settings. If you save with "ANSI" on Japanese Windows, it will be saved as "MS932 (ShiftJIS)." "Unicode" will be saved as "UTF-16LE," "Unicode big endian" will be saved as "UTF-16BE," and "UTF-8" will be saved as "UTF-8."

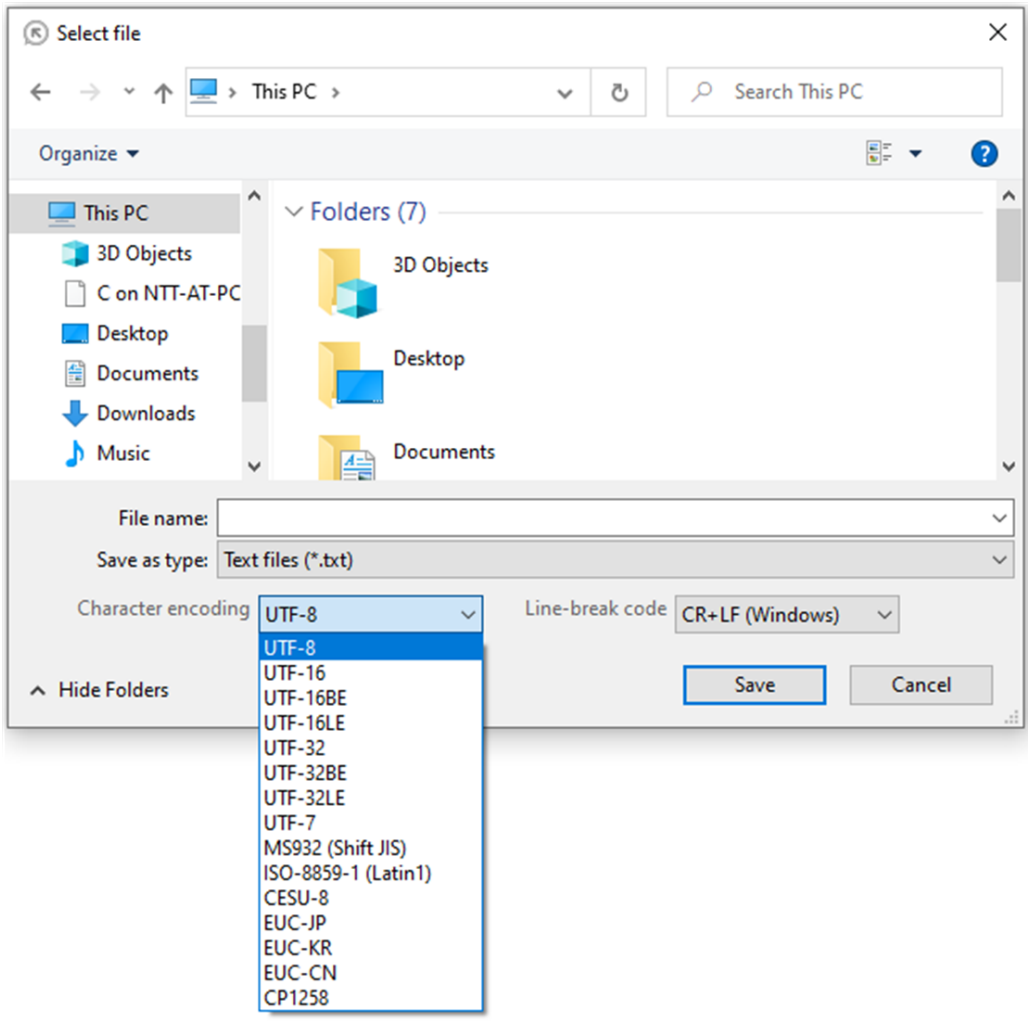


Figure 3-7. "Save As" window of Notepad

3.2.3. Cursor operation

Immediately after loading a text file into WinActor Note, the cursor position is "0" and the cursor is located at "TOP."

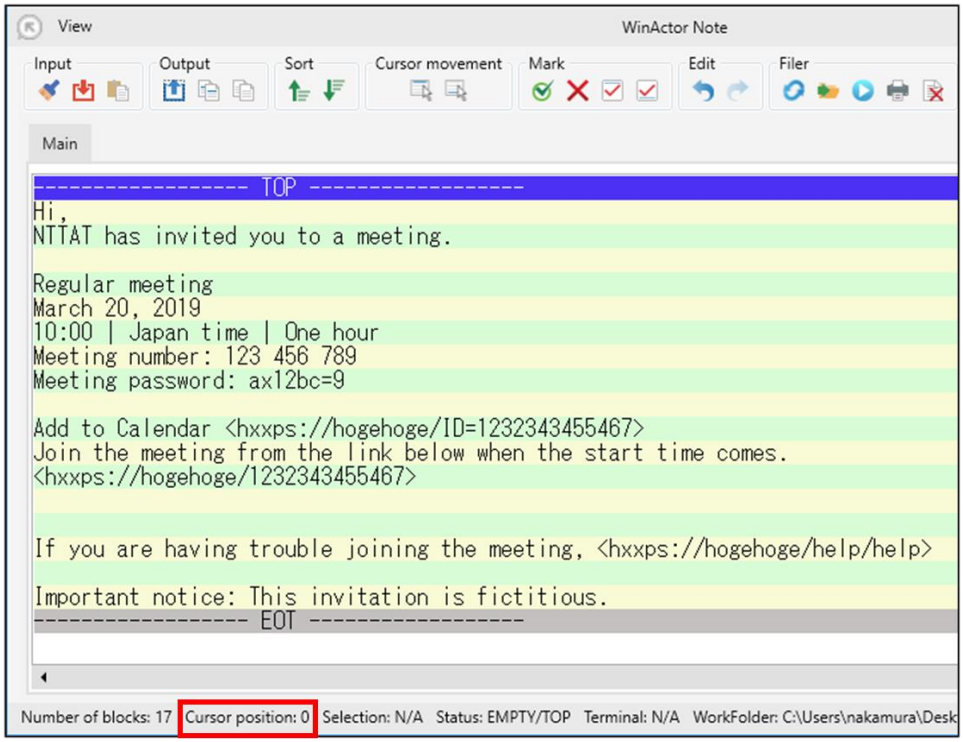


Figure 3-8. Immediately after loading a text file (Cursor position: 0)

Click "Select the next block" in the "Cursor movement" menu. The cursor position changes to "1" and the blue cursor moves to the first line of the texts.

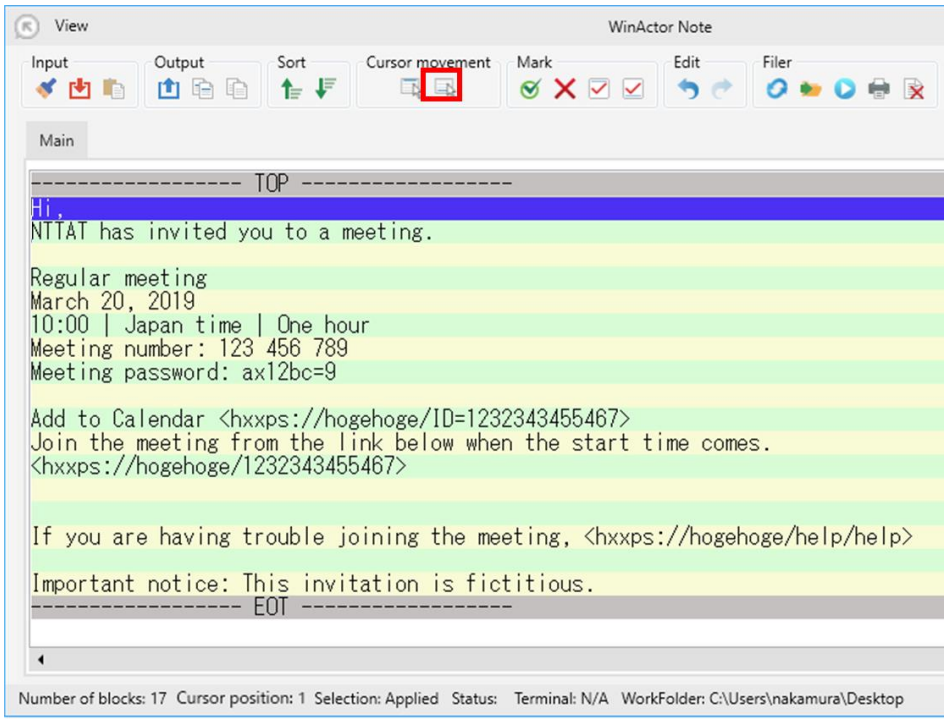


Figure 3-9. After moving the cursor (Cursor position: 1)

You can move the cursor back and forth by clicking "Select the previous block" and "Select the next block" in the "Cursor movement" menu.

There is another way of moving the cursor.

Click "Cursor movement tool" in the "Cursor movement" menu to display the "Cursor movement tool" window.

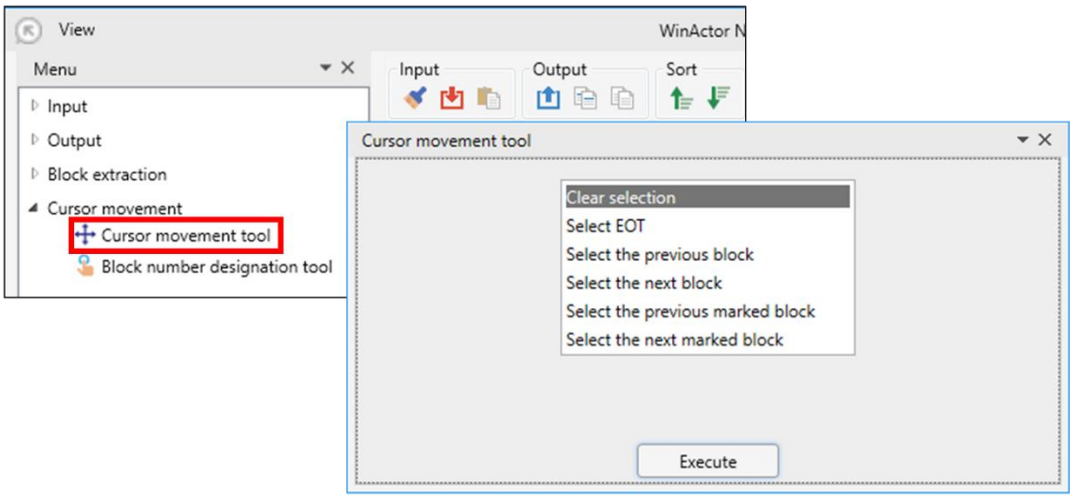


Figure 3-10. Cursor movement tool

You can move the cursor downward by selecting "Select the next block" and clicking ‘Execute’ in "Cursor movement tool."

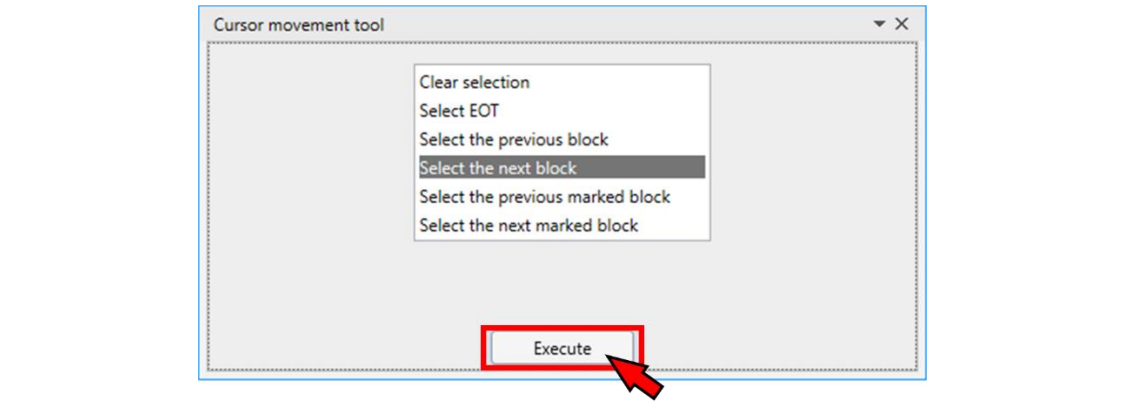


Figure 3-11. Moving the cursor using "Cursor movement tool"

Move to the cursor position "17" by using "Select the next block" of "Cursor movement tool." When the number of blocks is 17, the cursor position 17 is the last line of the loaded texts. If you execute "Select the next block" one more time, the cursor position changes to 18 and the cursor moves to the EOT (End Of Text).

Figure 3-12. Last line and EOT

When you execute "Clear selection" in "Cursor movement tool," the cursor position moves to TOP. When you execute "Select EOT," the cursor position moves to EOT.

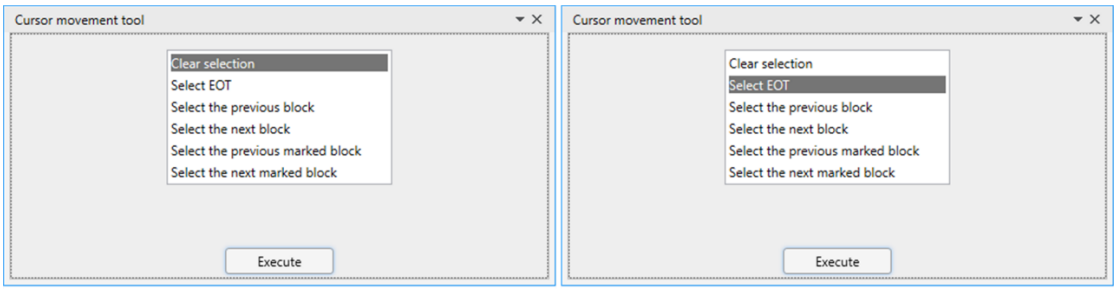


Figure 3-13. "Clear selection" and "Select EOT"

3.3. Processing texts

This section describes the procedure to process the texts of the meeting invitation into a format that WinActor can read easily.

In this tutorial, you can learn:

・Text processing using various functions of WinActor Note

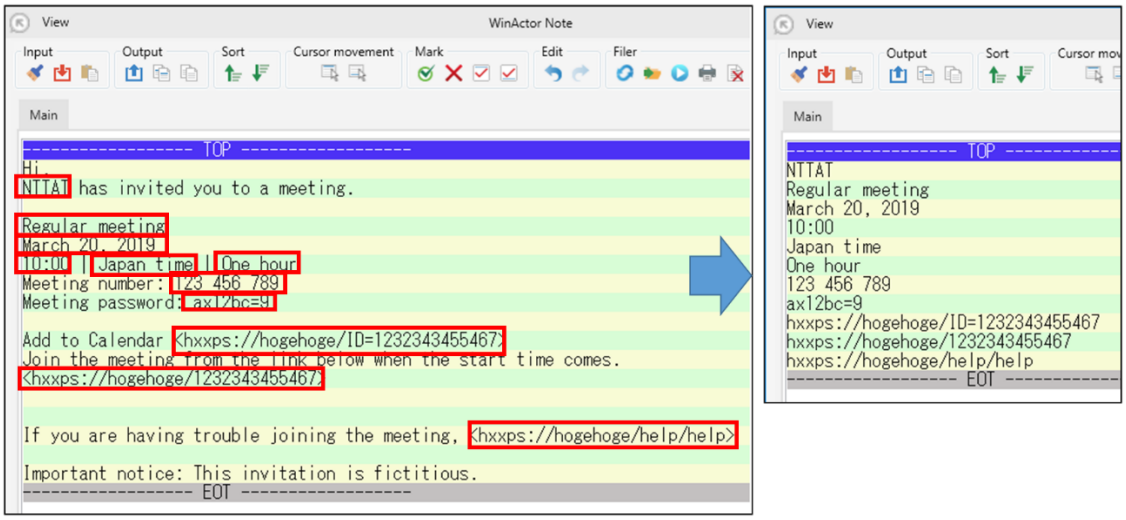


Figure 3-14. Image of text processing

3.3.1. Determining which information to be extracted

This tutorial shows a processing for extracting the information in the parts enclosed by the red frame in the figure below. There are various processing methods, and the method introduced in this tutorial is just an example.

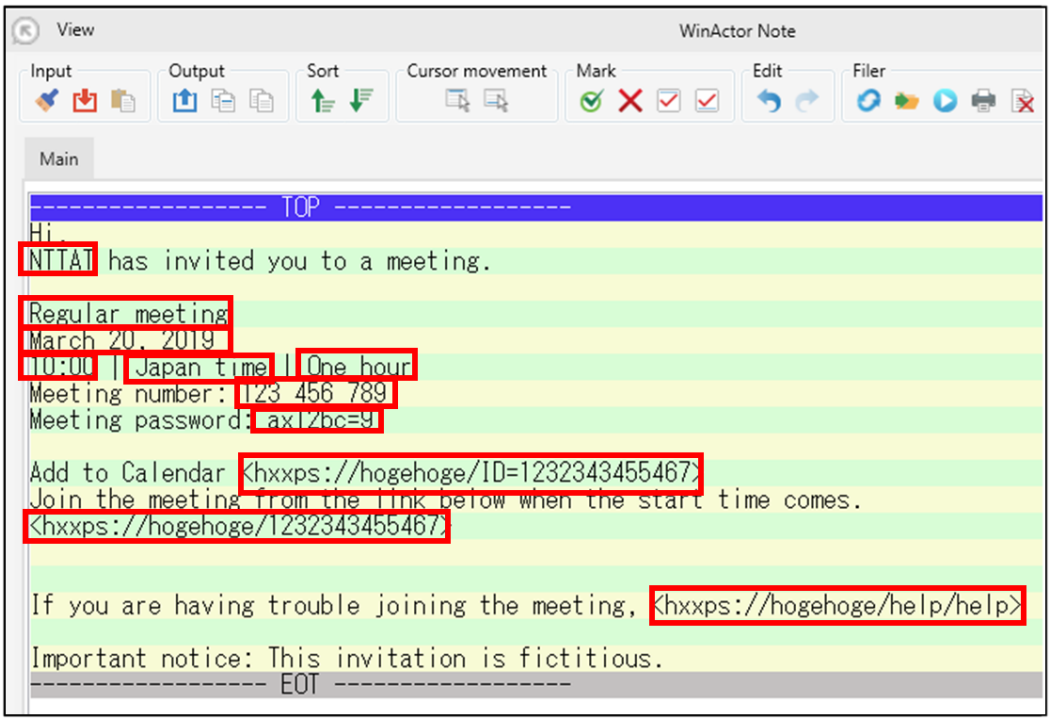


Figure 3-15. Information to be extracted

3.3.2. Deleting lines that do not contain the information to be extracted

First, delete "empty lines" (empty blocks) that do not contain the information to be extracted.

Open "Block extraction tool" from the "Block extraction" menu.

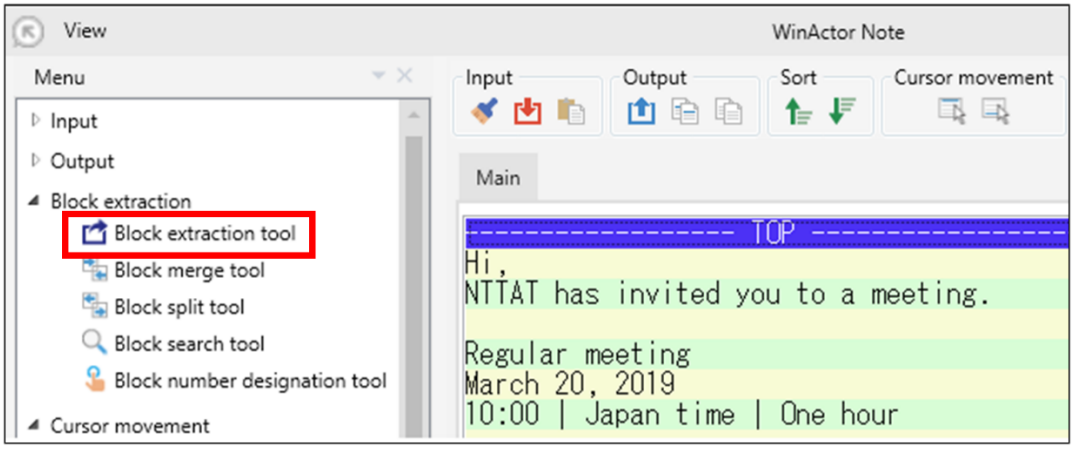


Figure 3-16. Opening "Block extraction tool"

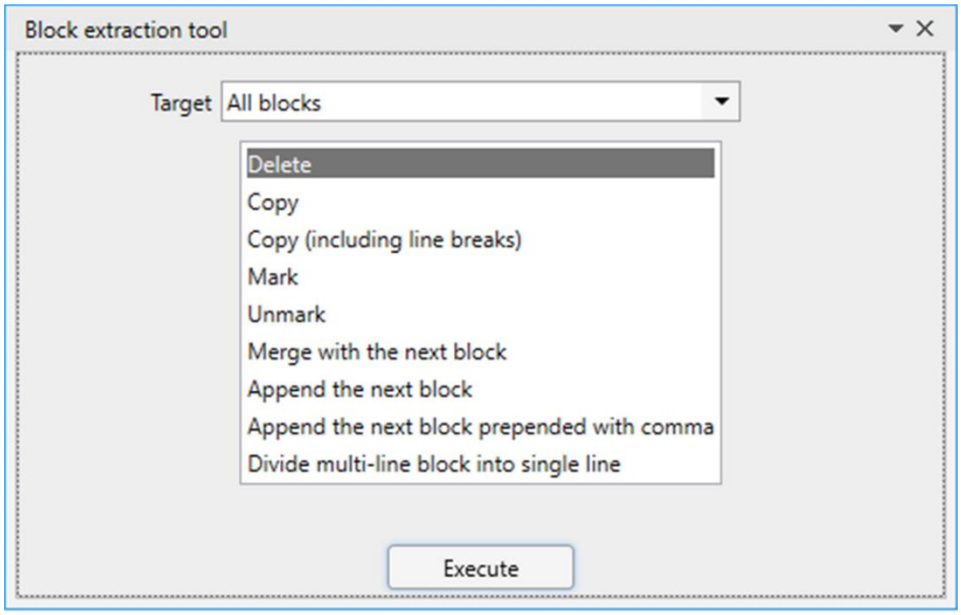


Figure 3-17. Block extraction tool

Select "Blocks that look empty" for the target and "Delete" for the operation, and click ‘Execute.’

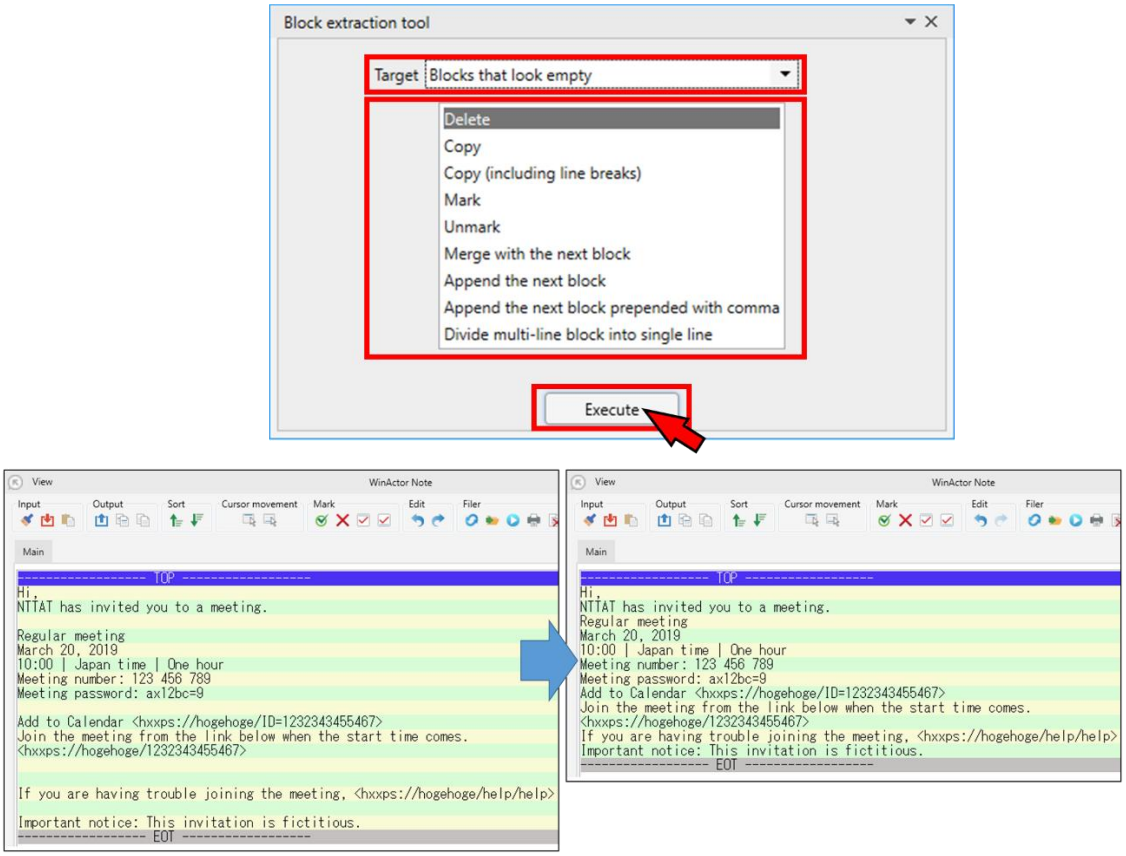


Figure 3-18. Deleting blocks that do not contain characters

Next, delete the line with "Hi," which is a block that does not contain the information to be extracted.

Open "Cursor movement tool" from the "Cursor movement" menu.

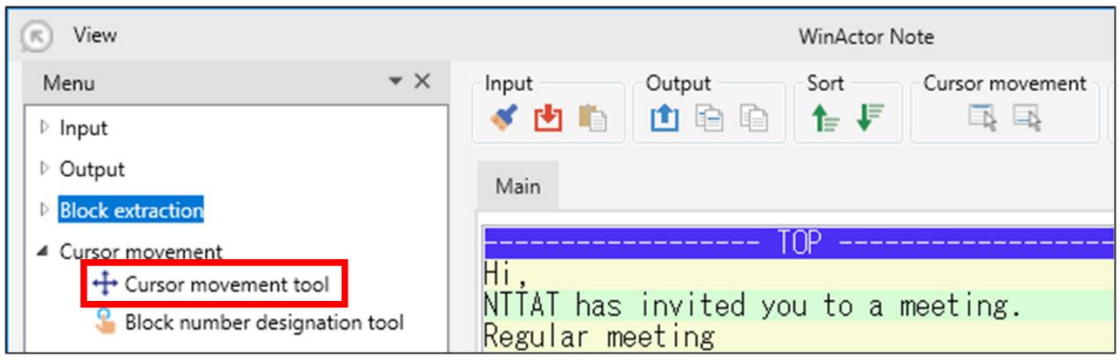


Figure 3-19. Opening "Cursor movement tool"

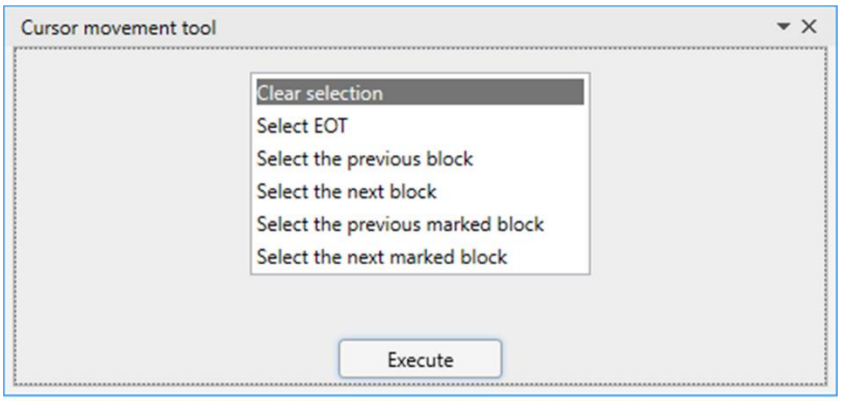


Figure 3-20. Cursor movement tool

To select the block containing "Hi," specify "Select the next block" and click ‘Execute’ in "Cursor movement tool."

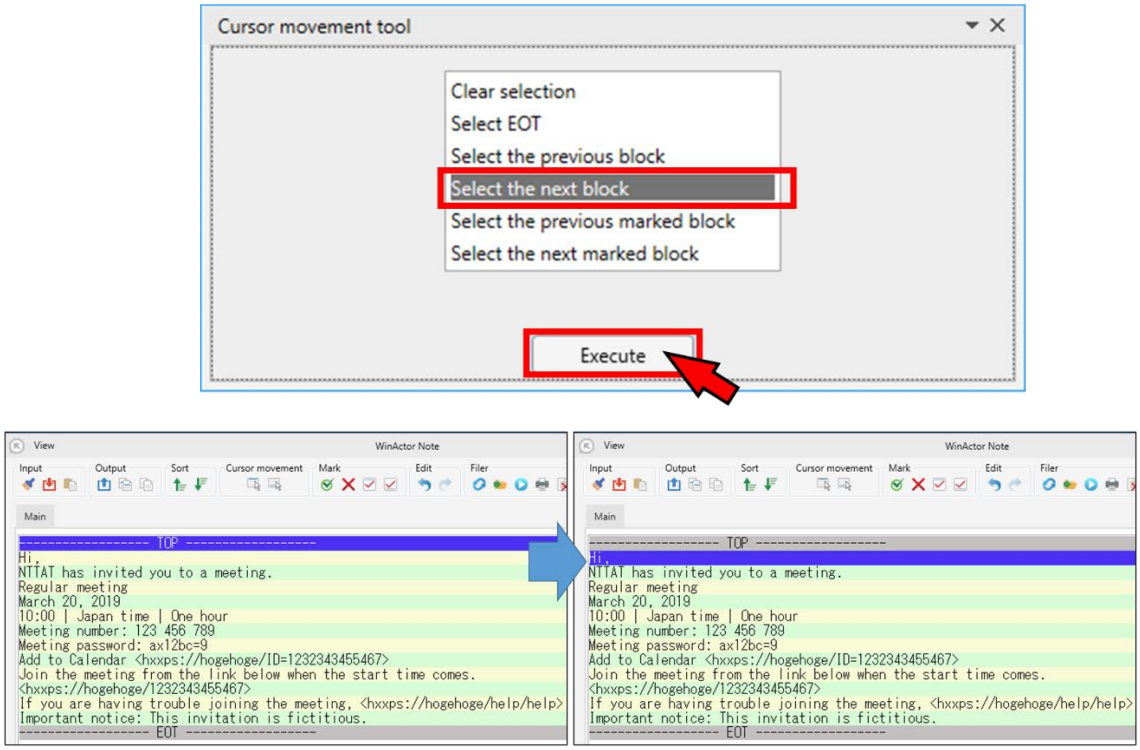


Figure 3-21. Selecting the block to be deleted

Next, in "Block extraction tool," specify "Selected block" for the target and "Delete" for the operation, and click ‘Execute.’

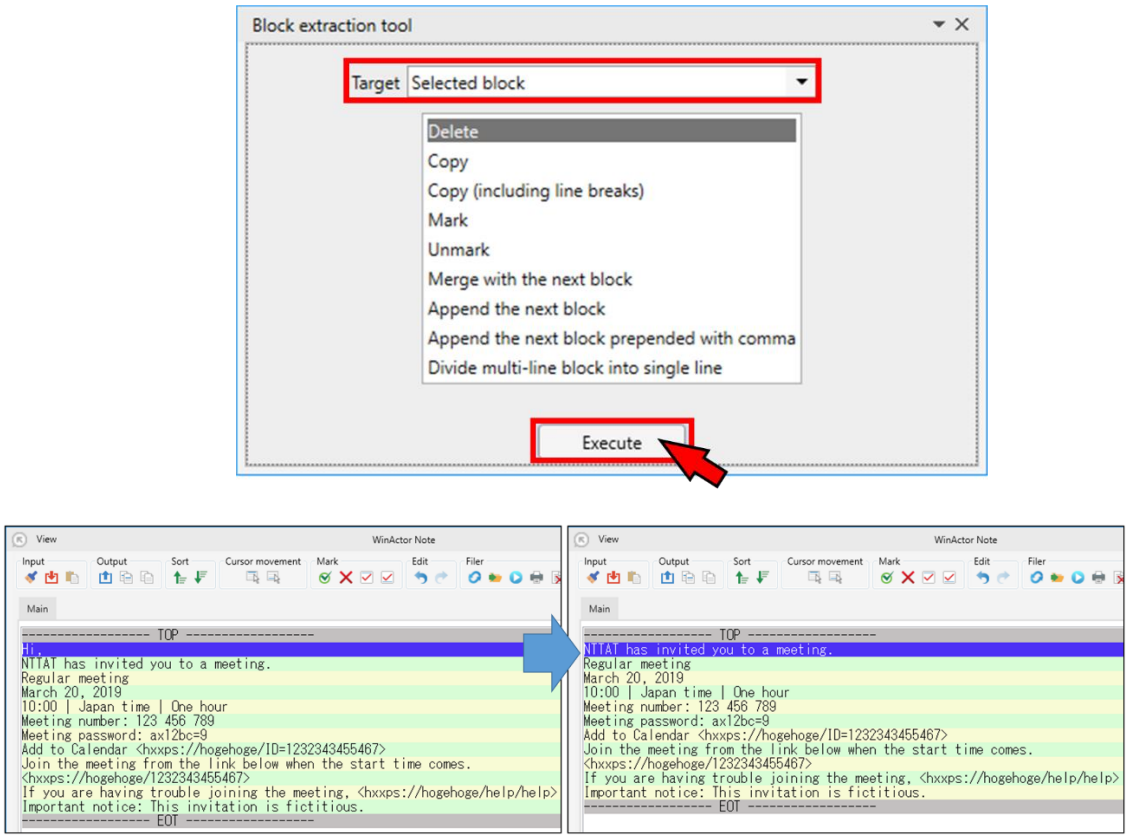


Figure 3-22. Deleting the unnecessary block

3.3.3. Editing a text in a line

To extract the information of the inviter, delete the part of "has invited you to a meeting." from "NTTAT has invited you to a meeting." to leave "NTTAT" only.

Open "Edit tool" from the "Edit" menu.

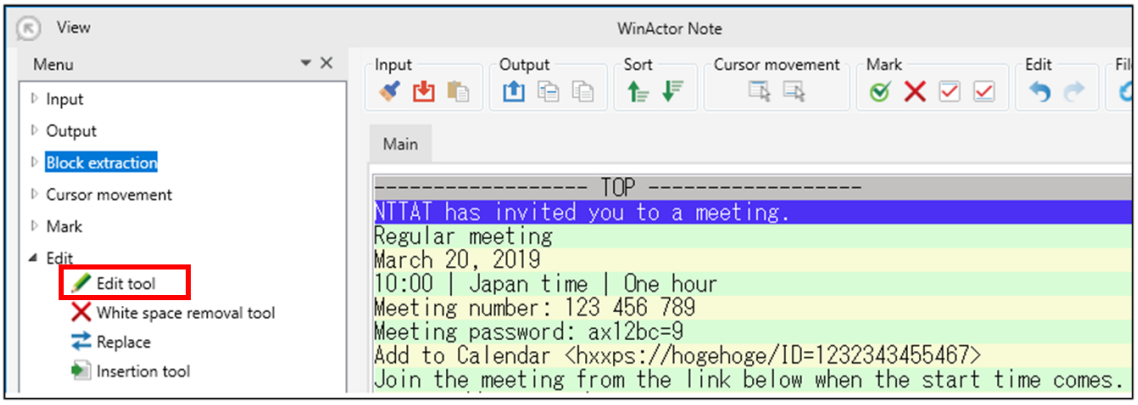


Figure 3-23. Opening "Edit tool"

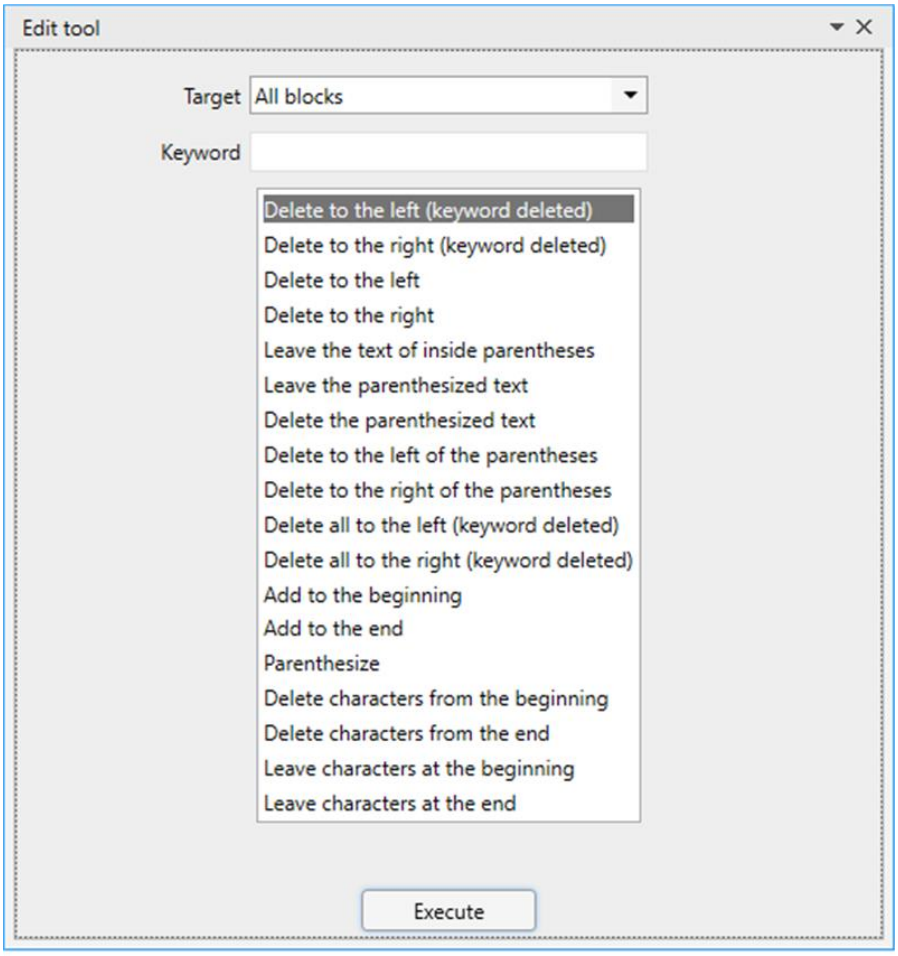
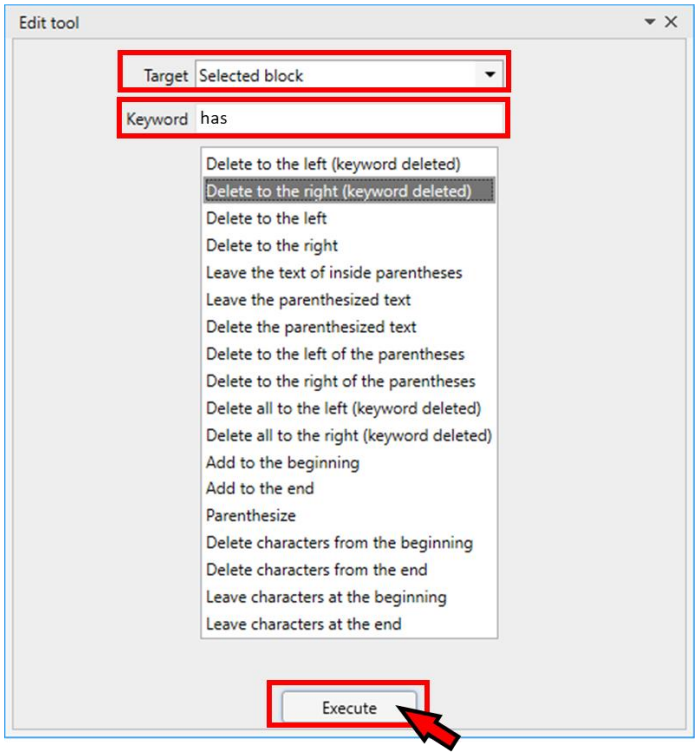


Figure 3-24. Edit tool

Specify "Selected block" for the target, enter "has" for the keyword, select "Delete to the right (keyword deleted)" for the operation, and click ‘Execute.’



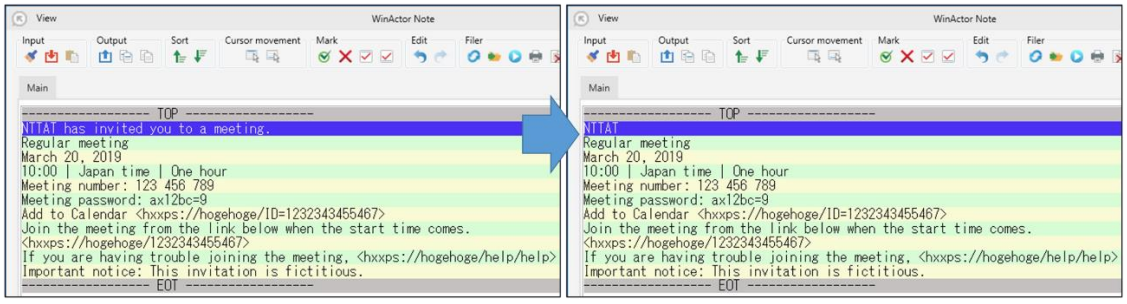


Figure 3-25. Editing the text in the line

3.3.4. Selecting a line away from the current position

Next, move the cursor to edit the line (block) with "10:00 | Japan time | One hour."

Open "Block number designation tool" from the "Cursor movement" menu.

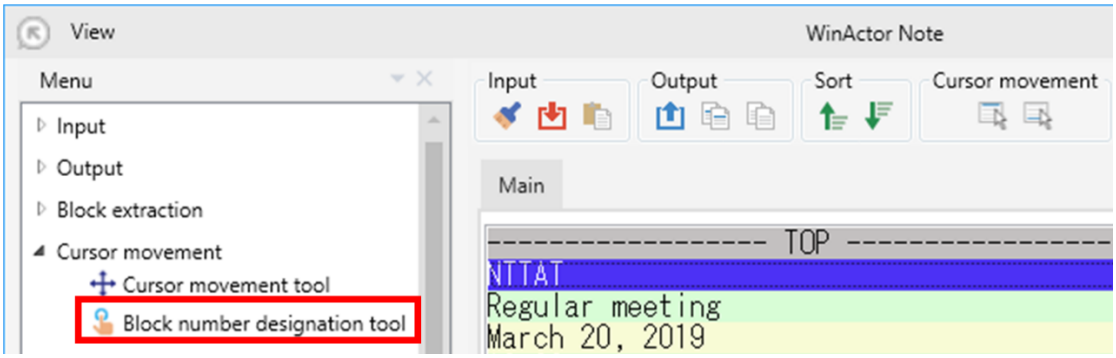


Figure 3-26. Opening "Block number designation tool"

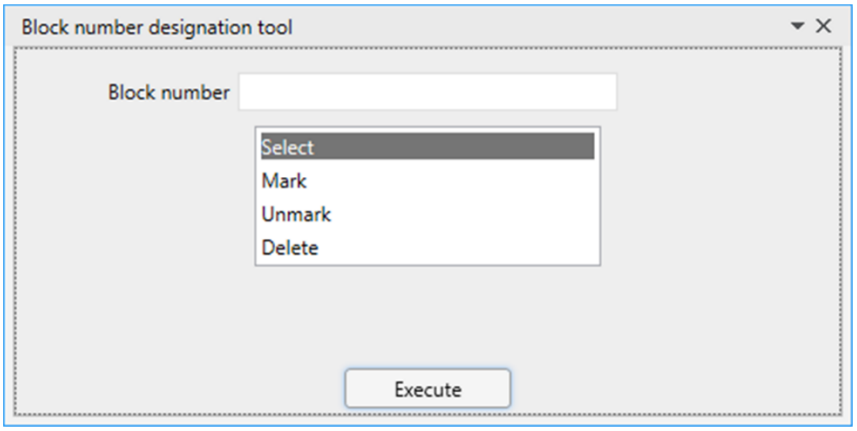


Figure 3-27. Block number designation tool

Select the "fourth" block using "Block number designation tool."

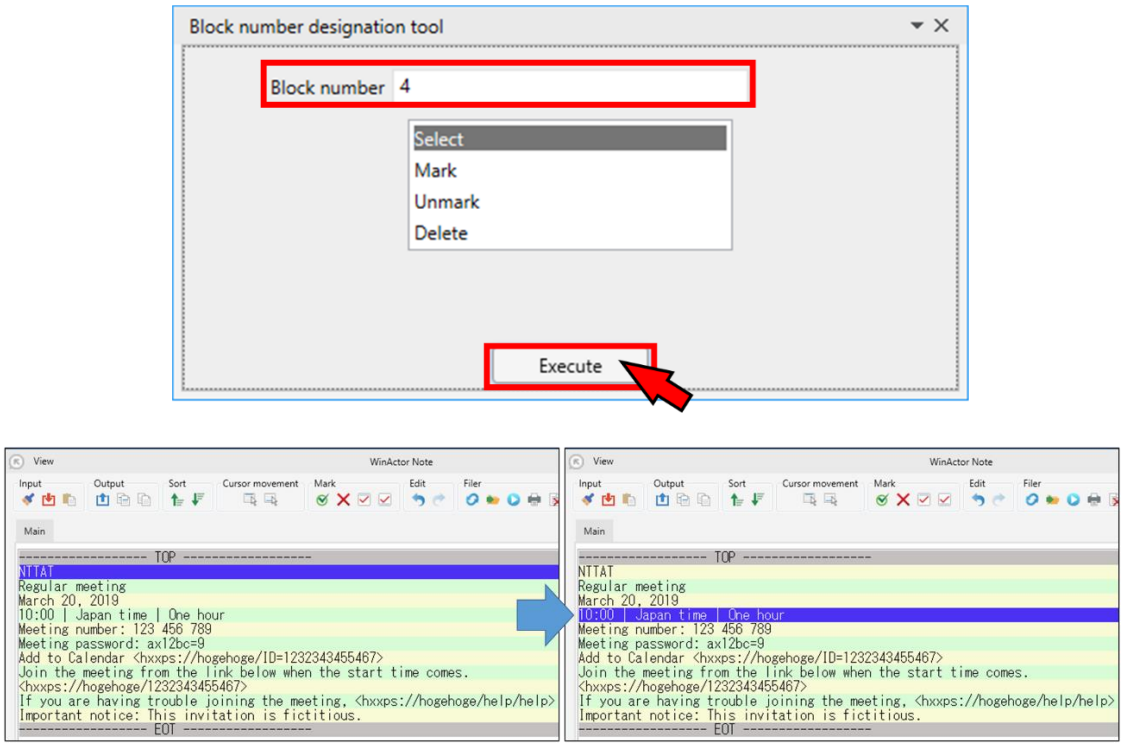


Figure 3-28. Selecting the line away from the current position

3.3.5. Splitting into multiple lines

Next, split the part "10:00 | Japan time | One hour" into three lines of "10:00," "Japan time," and "One hour."

Since it is separated by "|", specify "|" to split it.

Open "Block split tool" from the "Block extraction" menu.

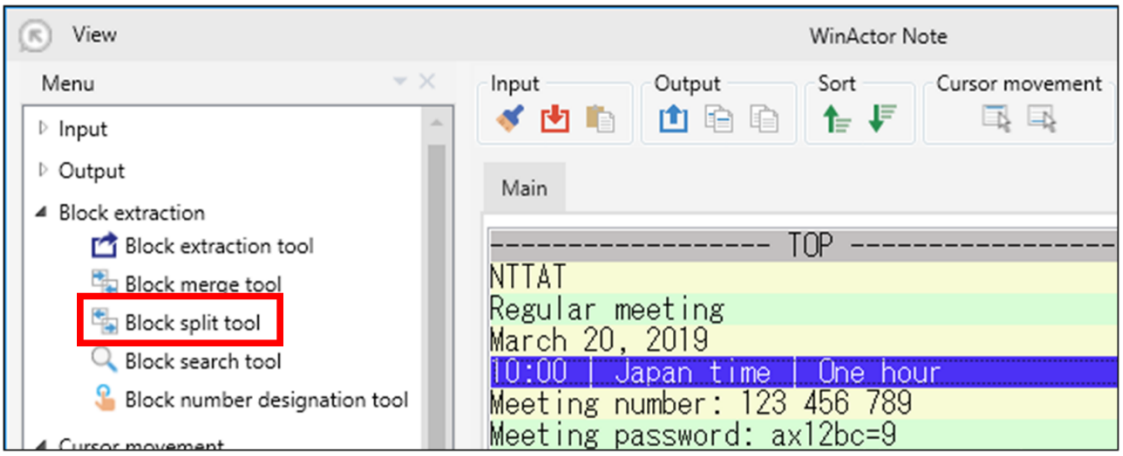


Figure 3-29. Opening "Block split tool"

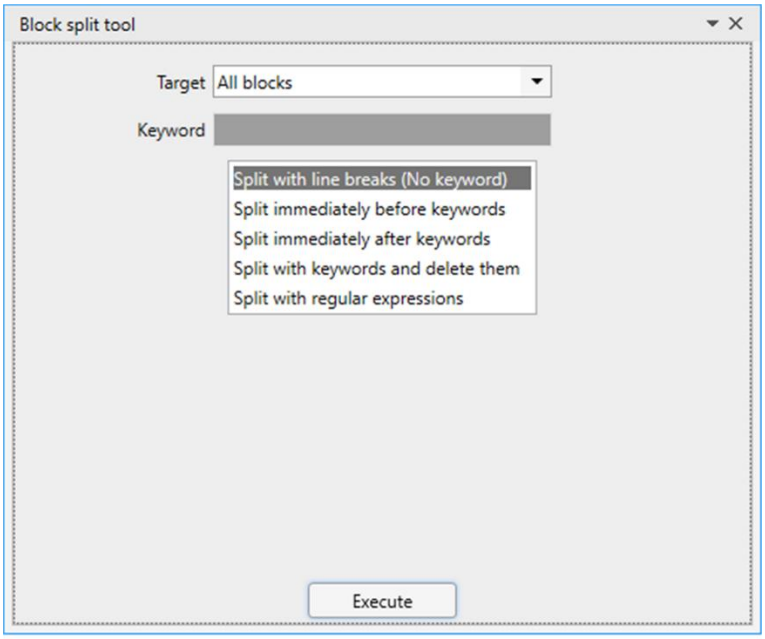


Figure 3-30. Block split tool

Specify "Selected block" for the target, "|" for the keyword, and "Split with keywords and delete them" for the operation, and click ‘Execute.’

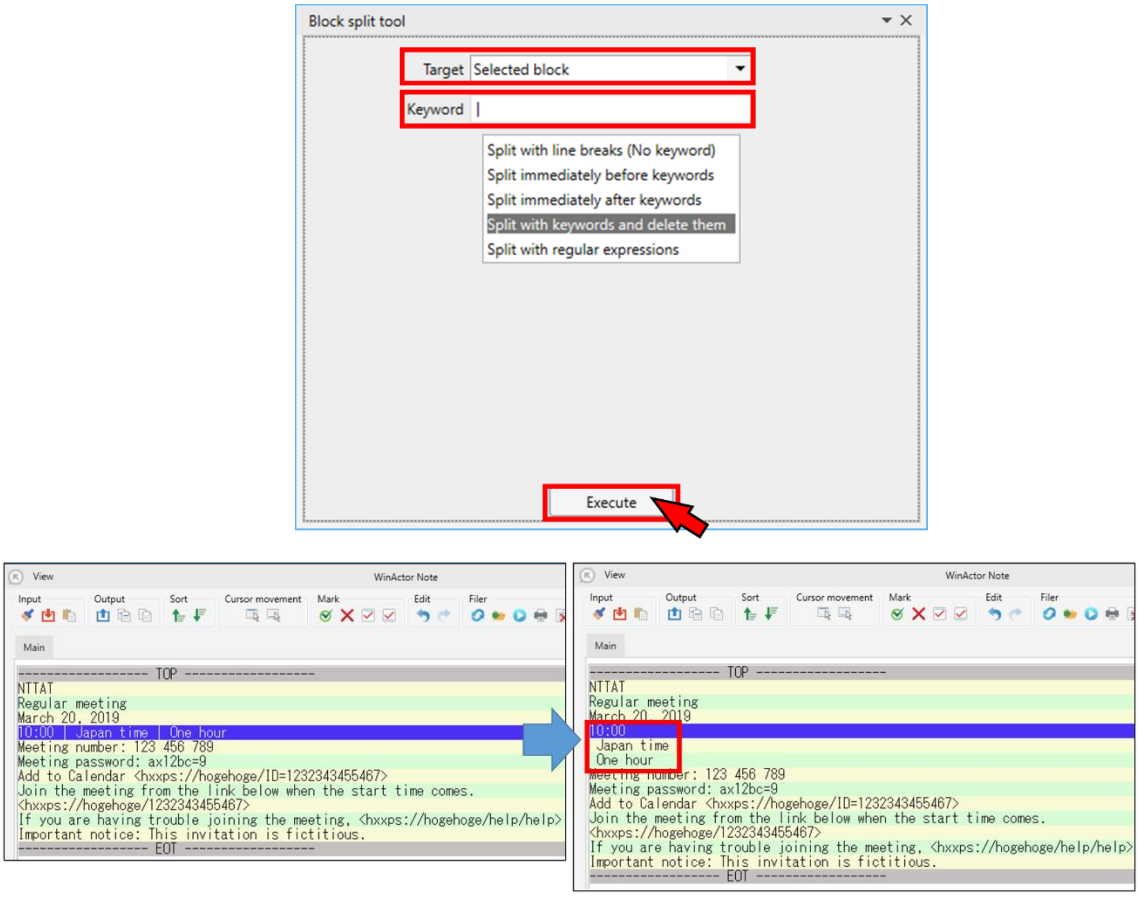


Figure 3-31. Splitting the block

3.3.6. Removing leading and trailing white spaces

The part "10:00 | Japan time | One hour" is now split into three lines. There are, however, leading and trailing white spaces in the information, such as "10:00 ," " Japan time," and " One hour." To remove the leading and trailing white spaces of each line, open "White space removal tool" from the "Edit" menu.

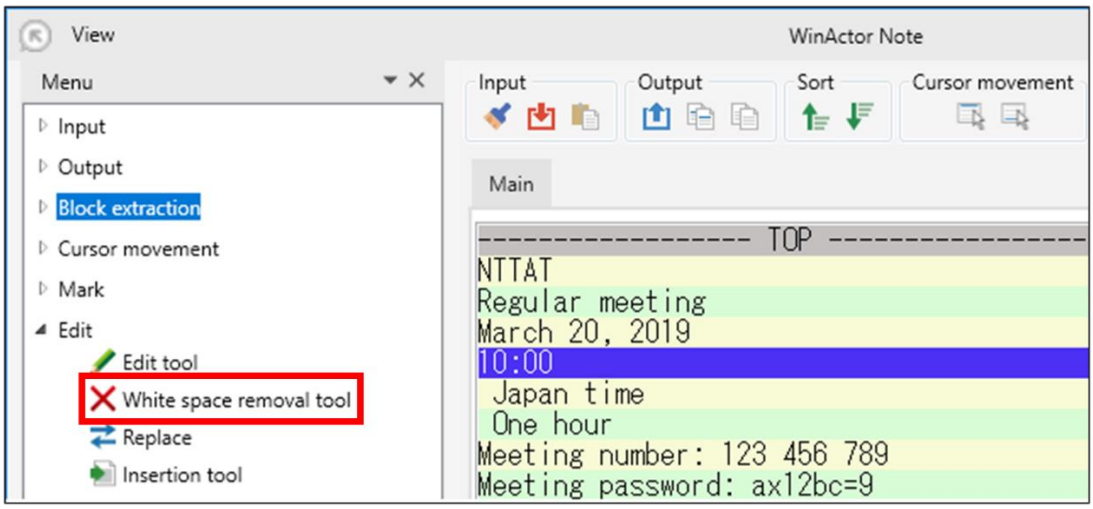


Figure 3-32. Opening "White space removal tool"

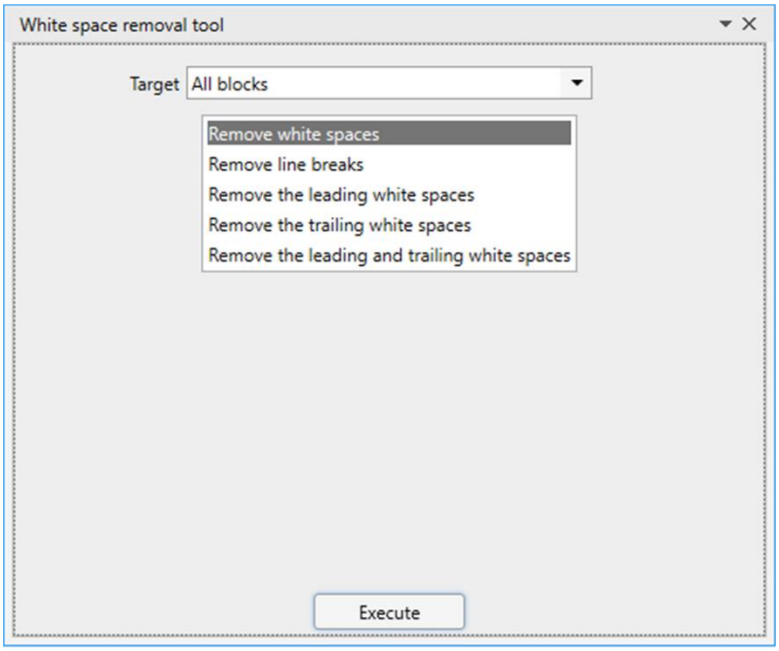


Figure 3-33. White space removal tool

In "White space removal tool," specify "All blocks" for the target and "Remove the leading and trailing white spaces" for the operation, and click ‘Execute.’

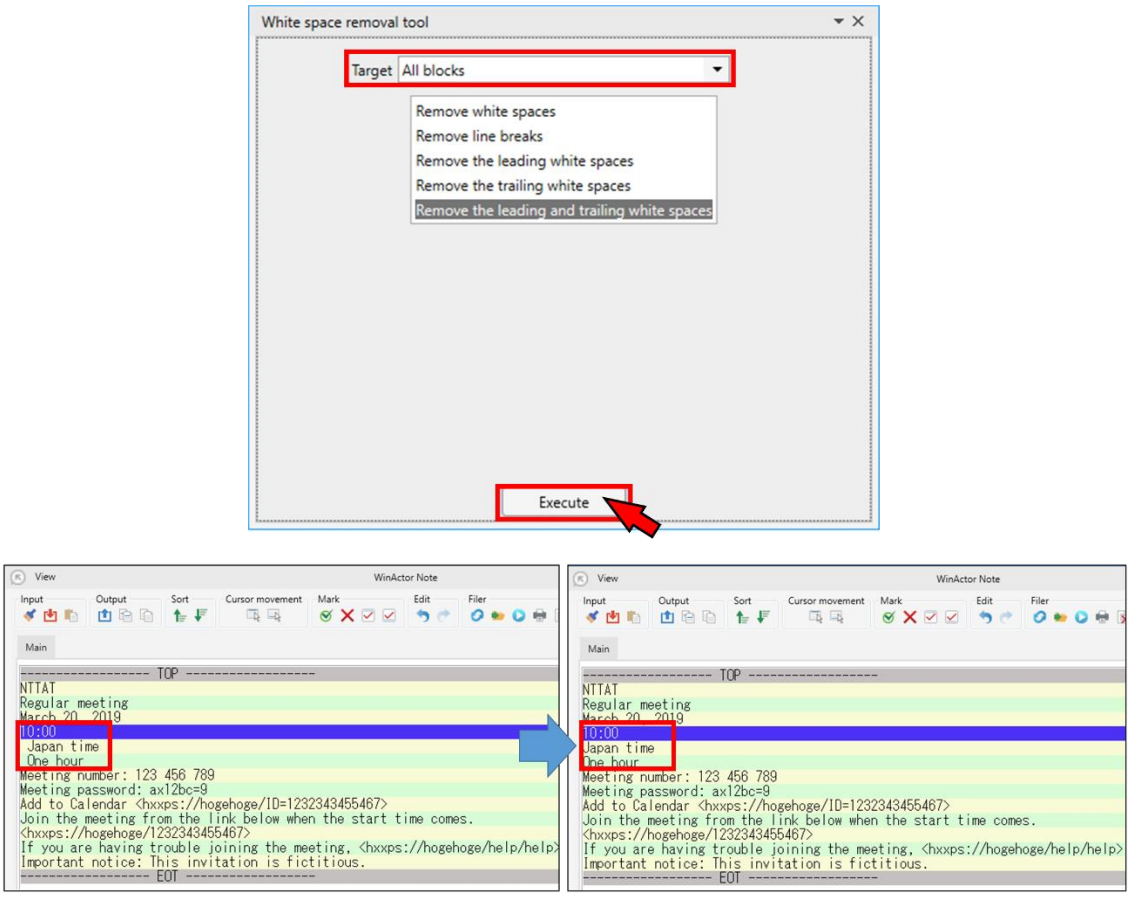


Figure 3-34. Removing the leading and trailing white spaces

3.3.7. Review of moving the cursor and editing a text in a line

From the texts below, extract the information of "Meeting number" and "Meeting password" in the red frame.

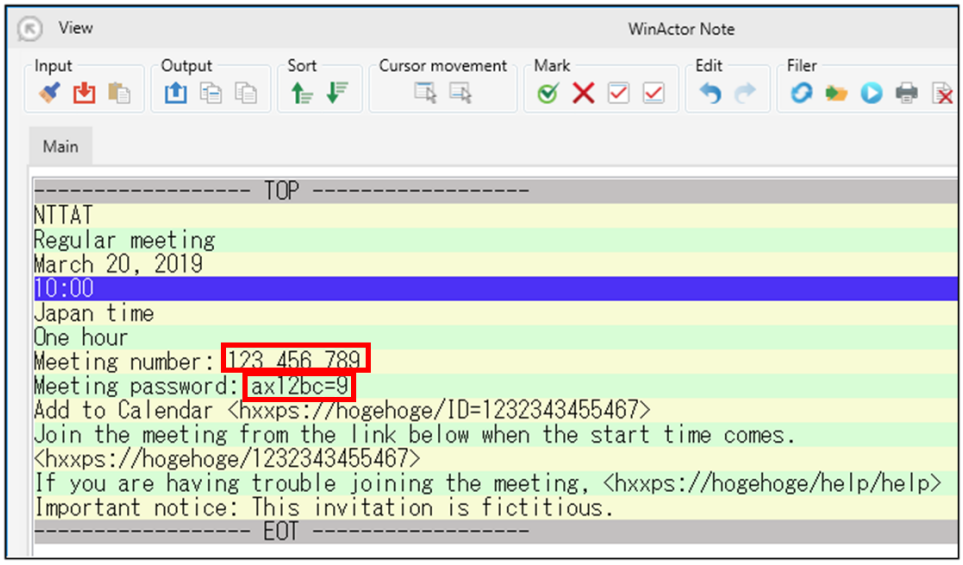
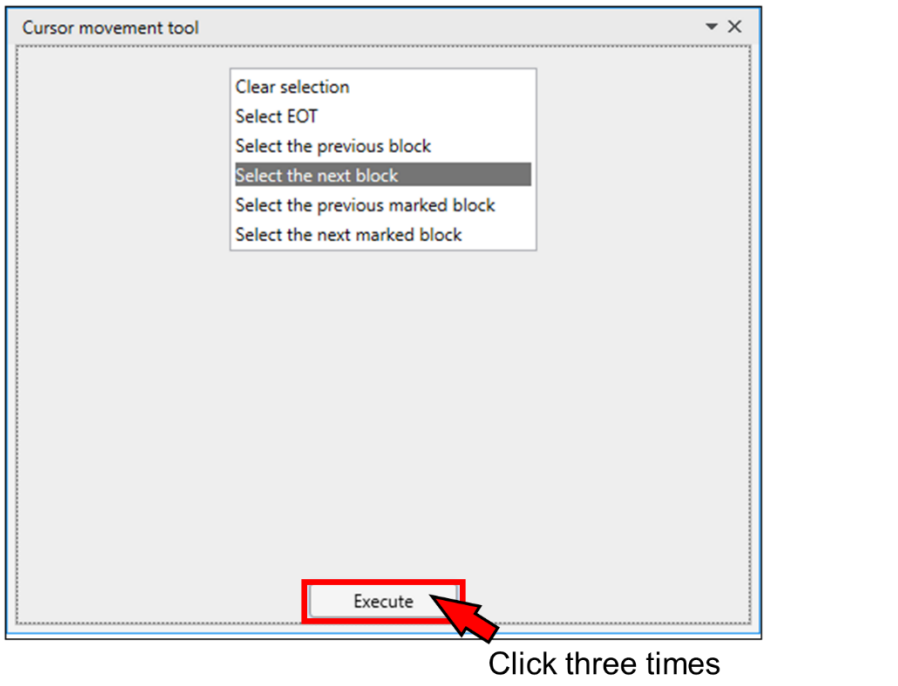


Figure 3-35. Text contents for review

In "Cursor movement tool," specify "Select the next block" and click ‘Execute’ three times to move the cursor to the position of "Meeting number."



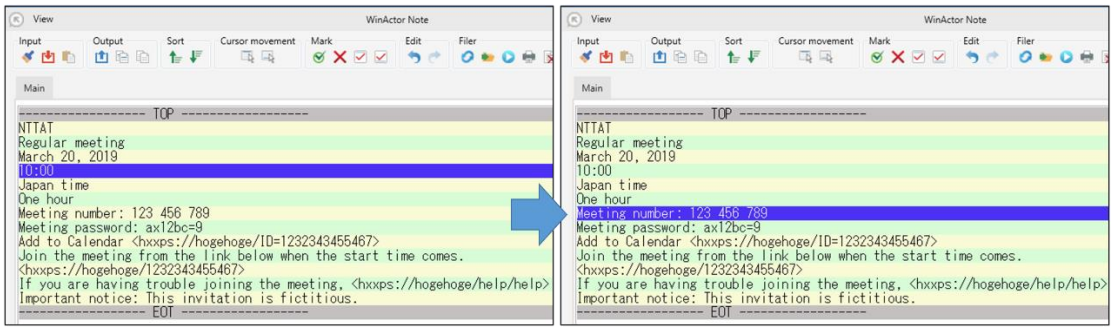


Figure 3-36. Moving the cursor

In "Edit tool," enter ":" for the keyword, specify "Selected block" for the target and "Delete to the left (keyword deleted)" for the operation, and click ‘Execute.’

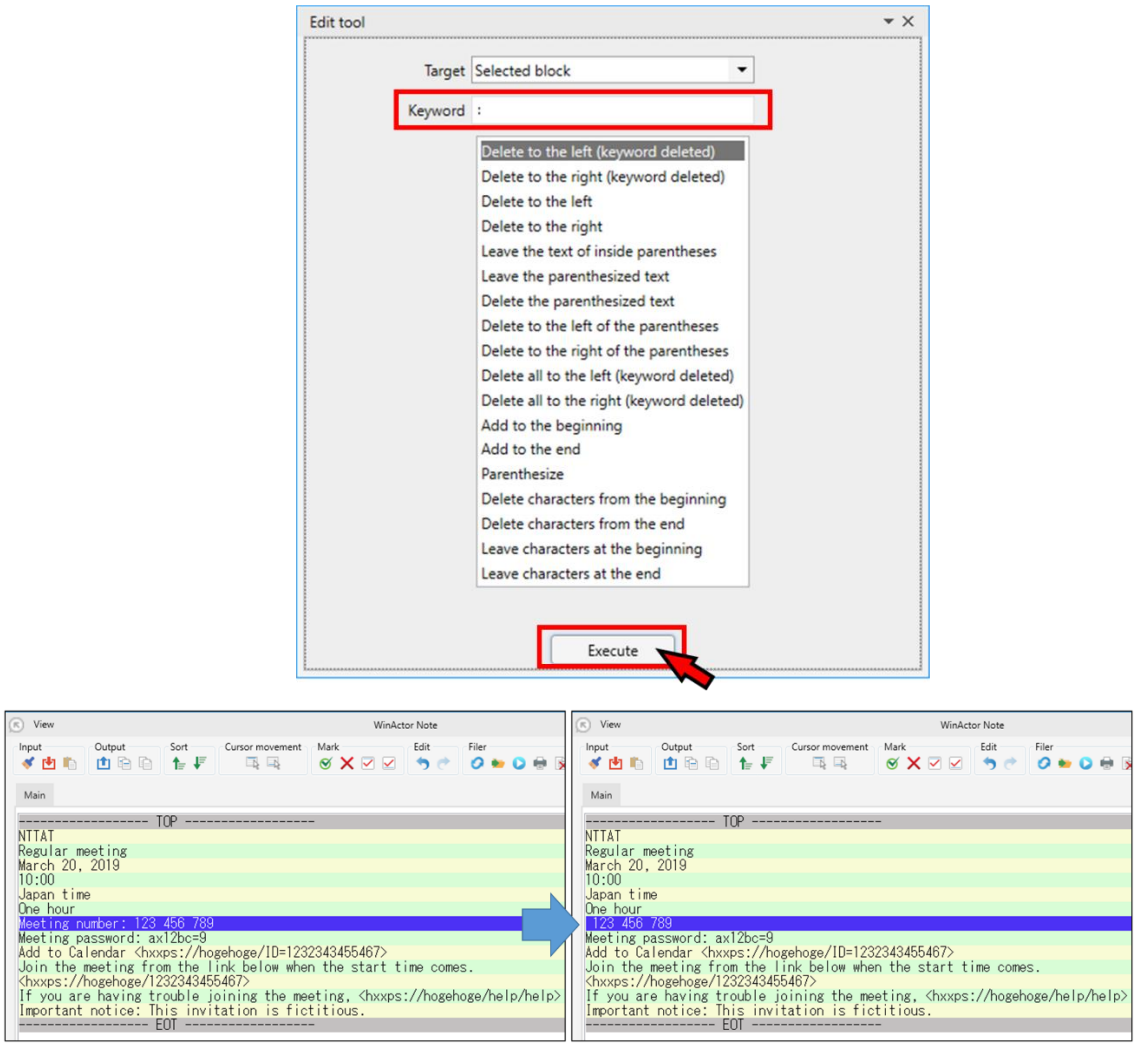


Figure 3-37. Review of editing the text in the line (Meeting number)

Extract the meeting password information as you did for the meeting number. In "Cursor movement tool," specify "Select the next block" and click ‘Execute’ to move the cursor to the position of "Meeting password." In "Edit tool," enter ":" for the keyword, specify "Selected block" for the target and "Delete to the left (keyword deleted)" for the operation, and click ‘Execute.’

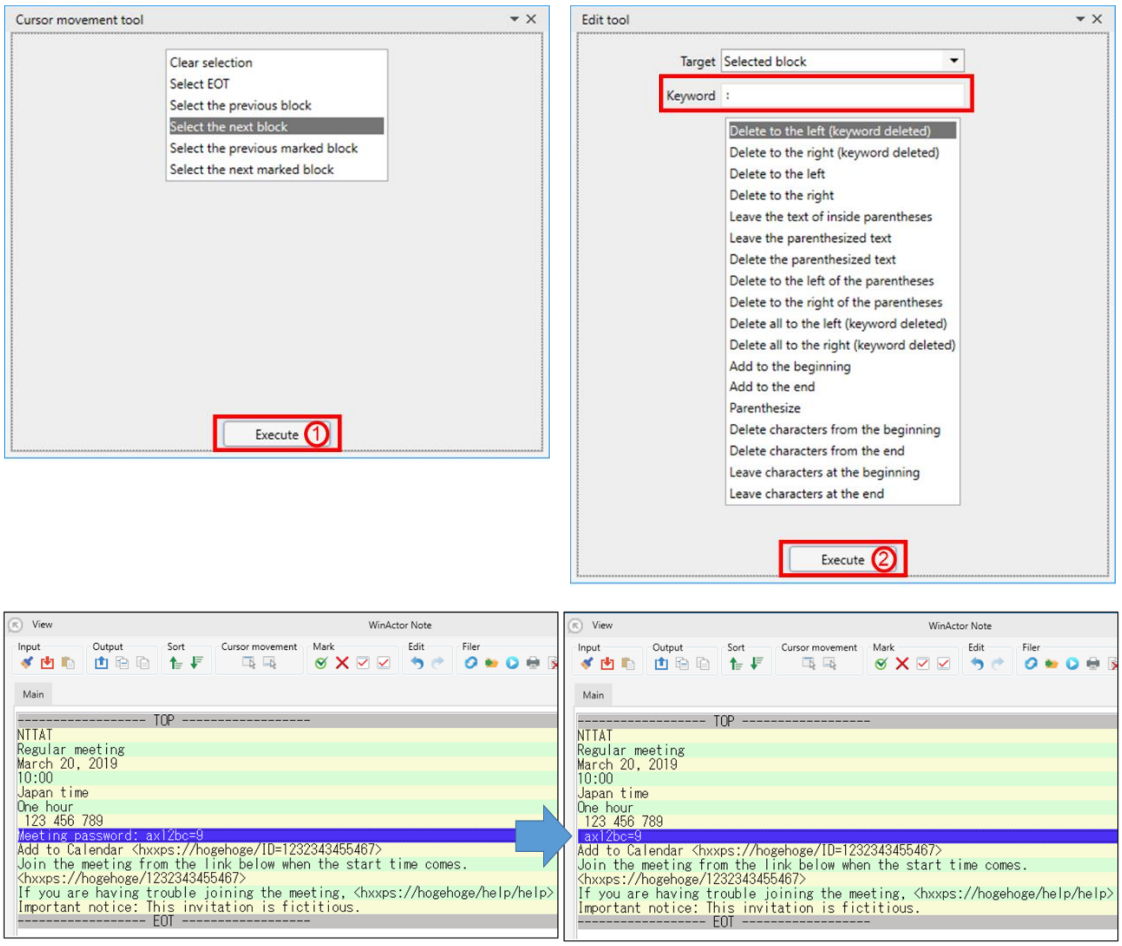


Figure 3-38. Review of editing the text in the line (Meeting password)

3.3.8. Extracting information enclosed in parentheses

Next, from the text "Add to Calendar <hxxps://hogehoge/ID=1232343455467>," extract the information "hxxps://hogehoge/ID=1232343455467" enclosed in parentheses "<>."

In "Cursor movement tool," specify "Select the next block" and click ‘Execute’ to move the cursor. Then, in "Edit tool," specify "Selected block" for the target and "Leave the text of inside parentheses" for the operation, enter "<>" for the parentheses, and click ‘Execute.’

* Enter the two characters of "<" (opening parenthesis) and ">" (closing parenthesis).

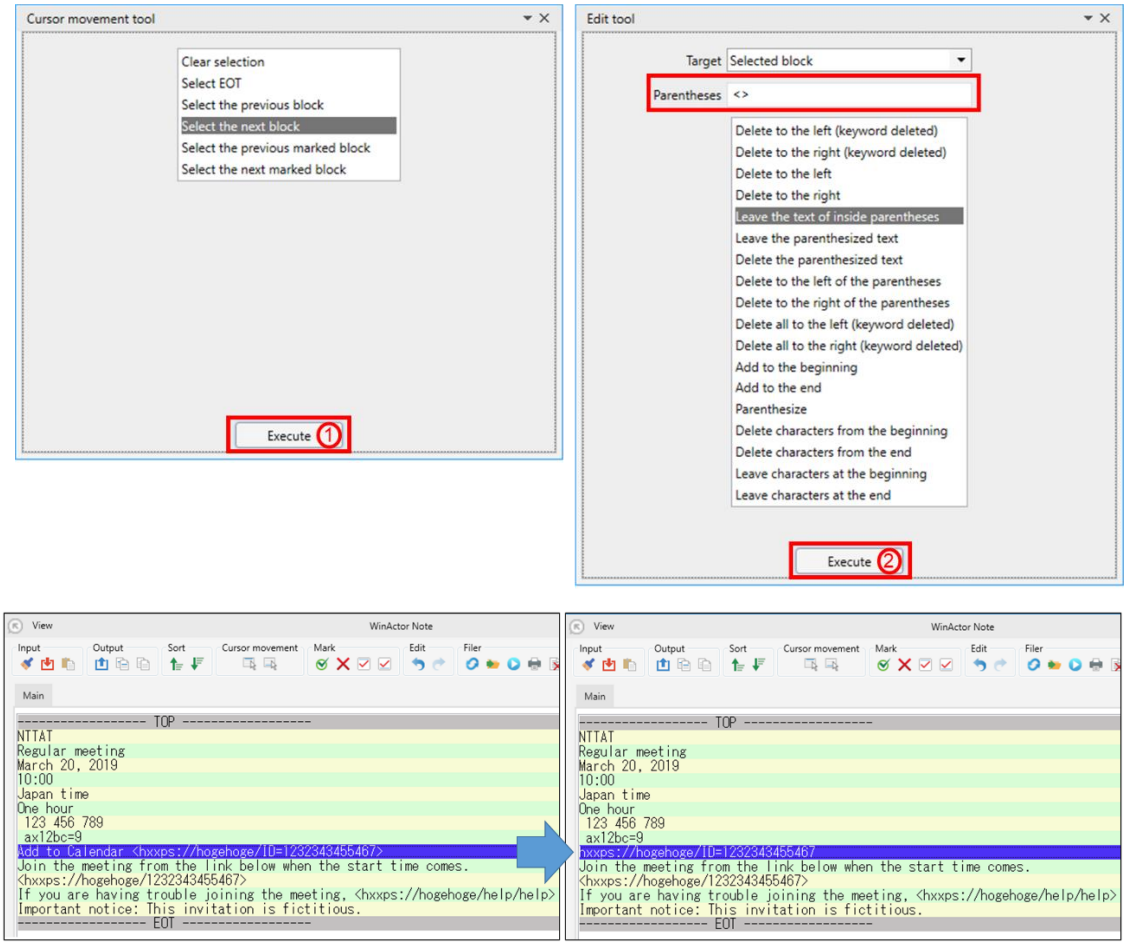


Figure 3-39. Extracting the information enclosed in parentheses

3.3.9. Practices

Practice 1.

For the lines " 123 456 789" and " ax12bc=9," remove the spaces at the beginning of the lines.

Practice 2.

Delete the lines "Join the meeting…" and "Important notice...."

Practice 3.

Extract the information "hxxps://…" enclosed in "<>."

The answers are on the next page.

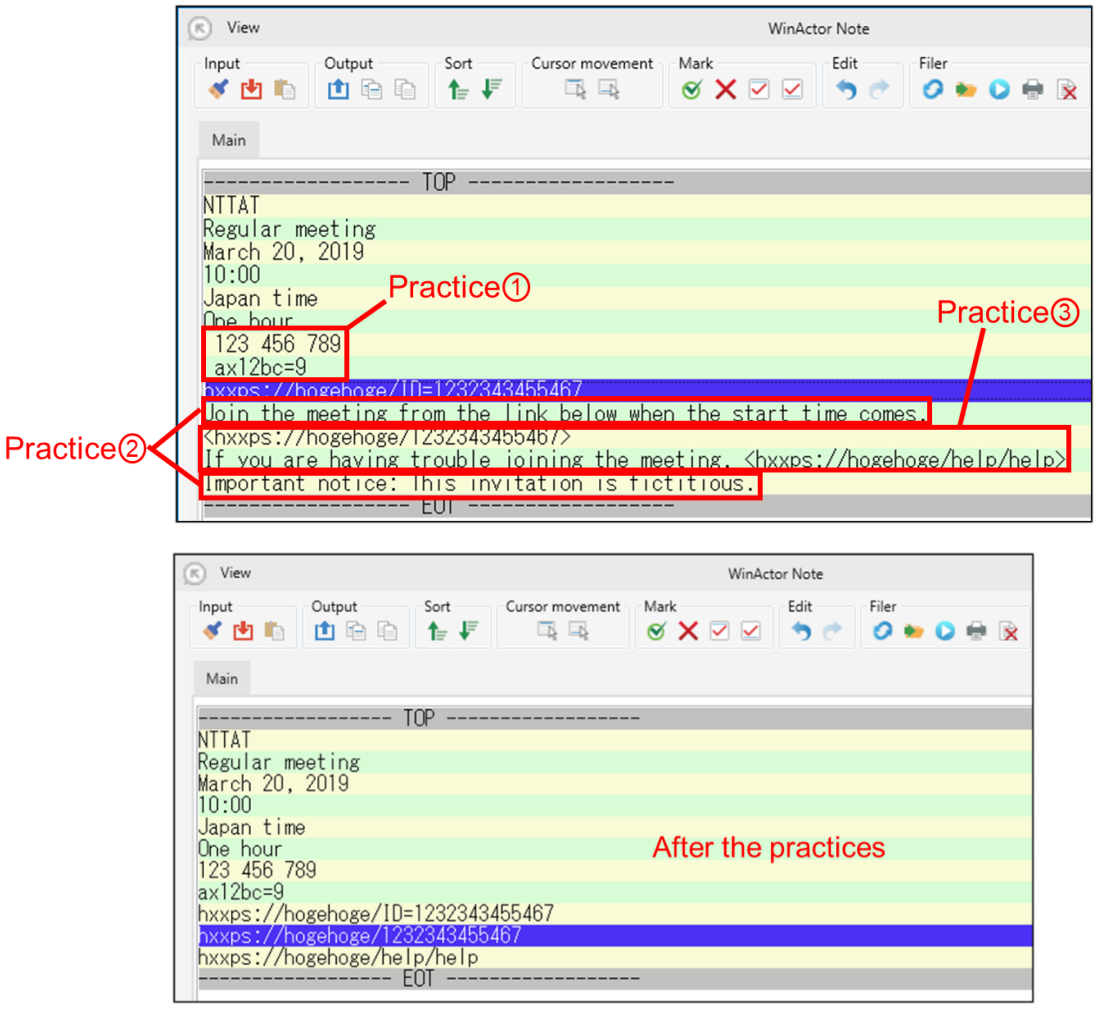


Figure 3-40. Texts for practices

Answer to Practice 1:

In "White space removal tool," specify "All blocks" for the target and "Remove the leading and trailing white spaces" for the operation, and click ‘Execute.’

Answer to Practice 2:

In "Cursor movement tool," specify "Select the next block" and click ‘Execute’ to move the cursor to the line to be deleted. Next, in "Block extraction tool," specify "Selected block" for the target and "Delete" for the operation, and click ‘Execute.’

* To move the cursor upward, execute "Select the previous block."

Answer to Practice 3:

In "Cursor movement tool," specify "Select the next block" or "Select the previous block" and click ‘Execute’ to move the cursor to the line containing "<hxxps://…>." Next, in "Edit tool," specify "Selected block" for the target and "Leave the text of inside parentheses" for the operation, enter "<>" for the parentheses, and click ‘Execute.’

* To move the cursor upward, execute "Select the previous block."

3.4. Recording and editing a macro

The section "3.3 Processing texts" describes the procedure to process the texts of the meeting invitation into the format that WinActor can easily read. This section introduces the procedure to record the text processing procedure as a macro.

In order for WinActor to execute the text processing, it is necessary to record the text processing procedure as a macro. The text processing is executed by running a WinActor Note macro from WinActor.

The procedure to link WinActor and WinActor Note is as follows.

・Record a WinActor Note macro
・Edit and save the WinActor Note macro
・Create a WinActor scenario
  Execute the following in the WinActor scenario:
  ① Run the WinActor Note macro
  ② Import the text processing result of WinActor Note into WinActor

If you make mistakes during the macro recording, wrong operations will be recorded. It is therefore recommended that you practice the operations for recording several times beforehand.

Practice the contents of "3.3 Processing texts" several times before proceeding to record the macro.

3.4.1. Starting the macro recording

Start recording a macro in the "Edit macro" pane. If the pane is not displayed, click "Edit macro" in the "View" menu to display the "Edit macro" pane.

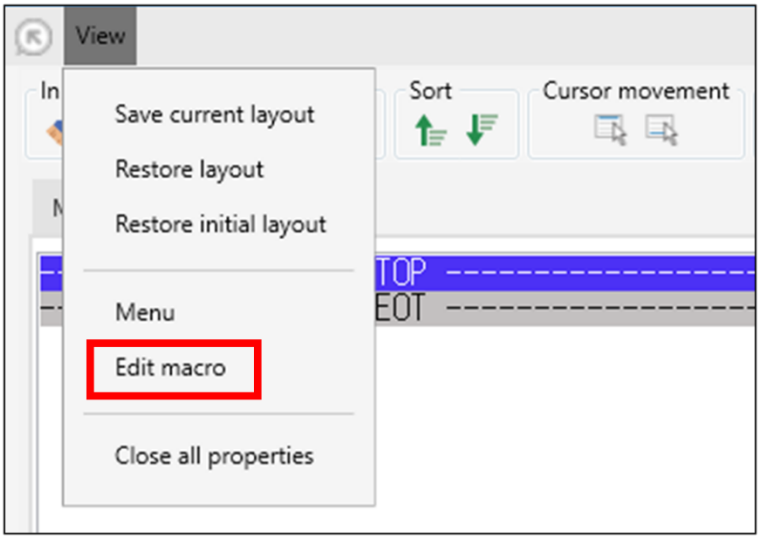


Figure 3-41. Opening the "Edit macro" pane

Start recording by clicking "Record" in the "Edit macro" pane.

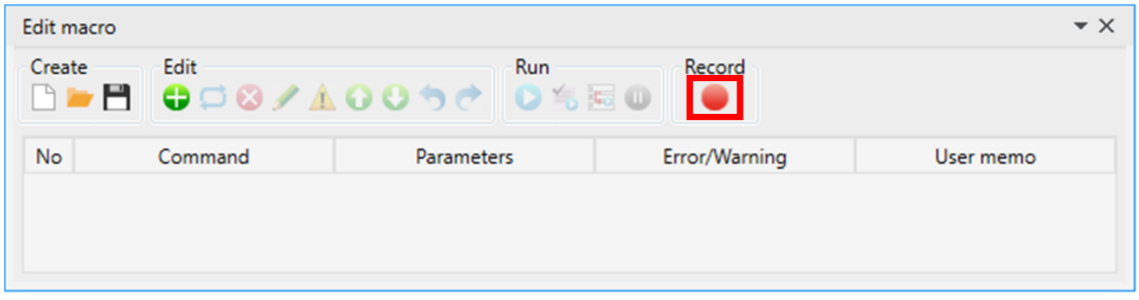


Figure 3-42. Starting the macro recording

3.4.2. Recording the text processing operations

Load the texts of the meeting invitation and process the texts by referring to the operations performed in "3.3 Processing texts."

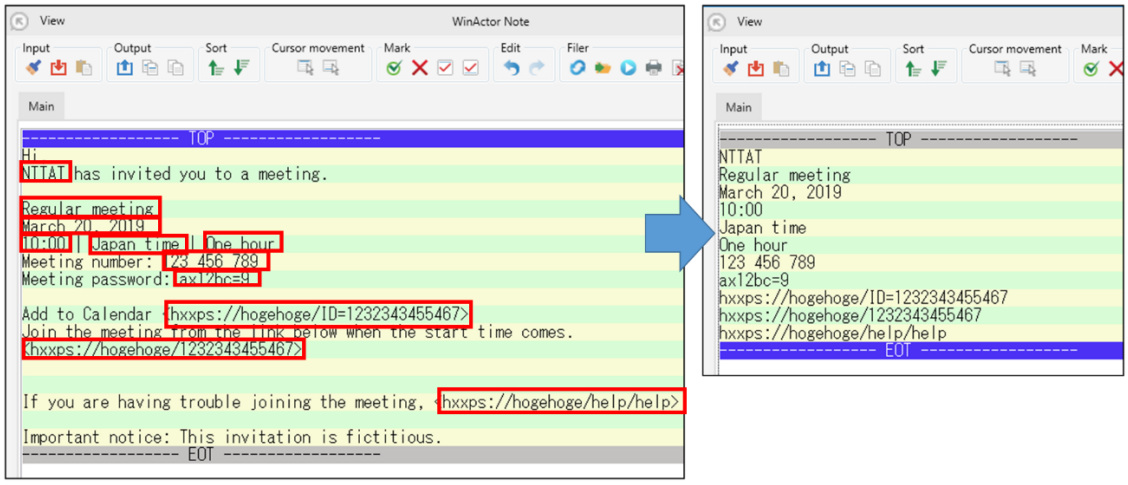


Figure 3-43. Image of processing the texts

3.4.3. Saving the created macro

When you execute the text processing operations, the steps of the text processing operations are being recorded in the macro area of the "Edit macro" pane. When you have finished recording, click "Record" to stop recording.

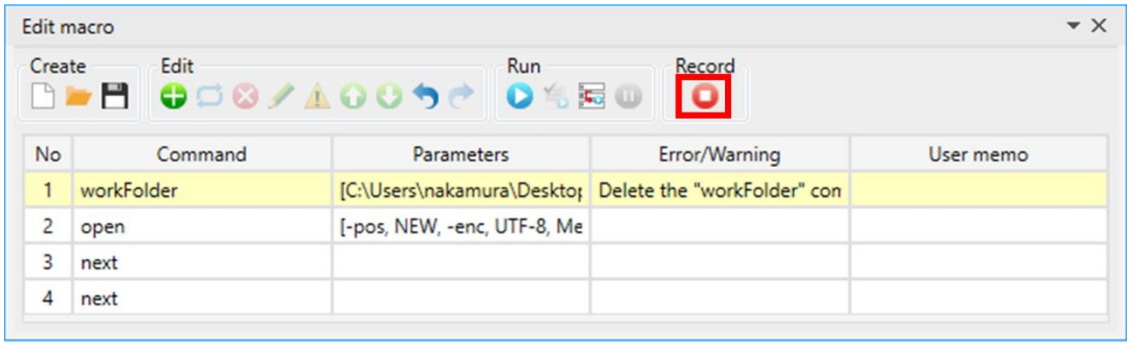


Figure 3-44. Macro for the text processing steps and stopping its recording

To save this macro for the text processing steps in a file, click "Save" in the Create menu of the "Edit macro" pane. Save it in the tutorial folder with the filename "Meeting_request_processing."
The file will be saved in JSON format.

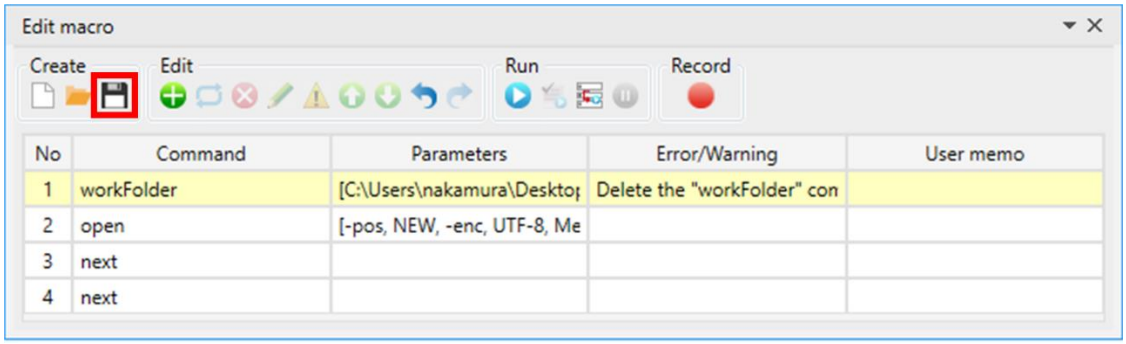


Figure 3-45. Saving the recorded macro

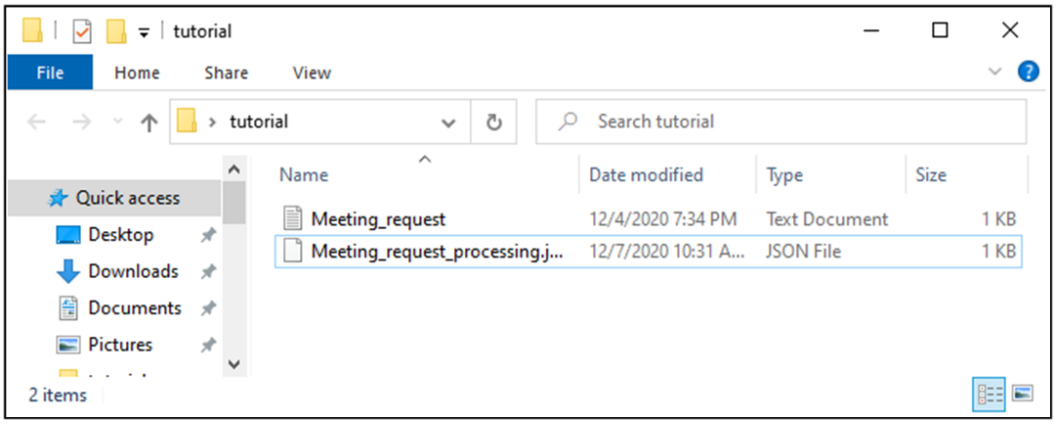


Figure 3-46. Tutorial folder after saving the macro

3.4.4. Editing the created macro

This tutorial introduces the procedure to instruct to load the file from the node of WinActor instead of the macro of WinActor Note.

In this procedure, the file loading part recorded in the macro is unnecessary and need to be deleted.

No.1 "workFolder" and No.2 "open" are the file loading part.

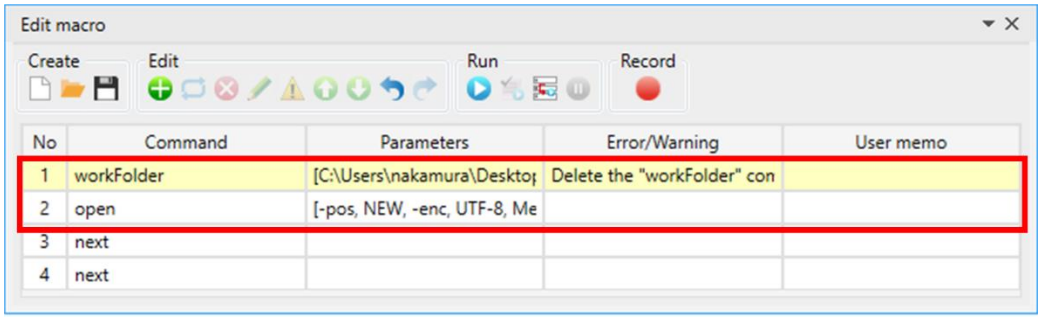


Figure 3-47. Part unnecessary for this tutorial

In the "Edit macro" pane, select No.1 and No.2 in the macro area and click "Delete" in the Edit menu.

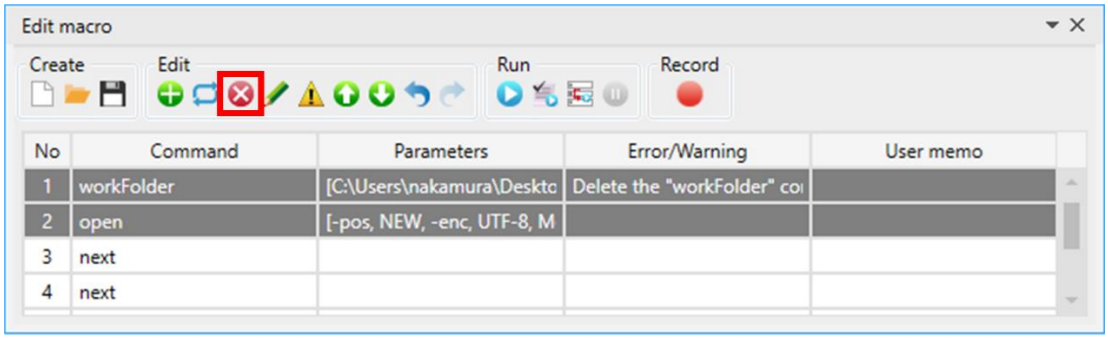


Figure 3-48. Editing the recorded macro

Save the edited result again. Overwrite the file with the filename "Meeting_request_processing."

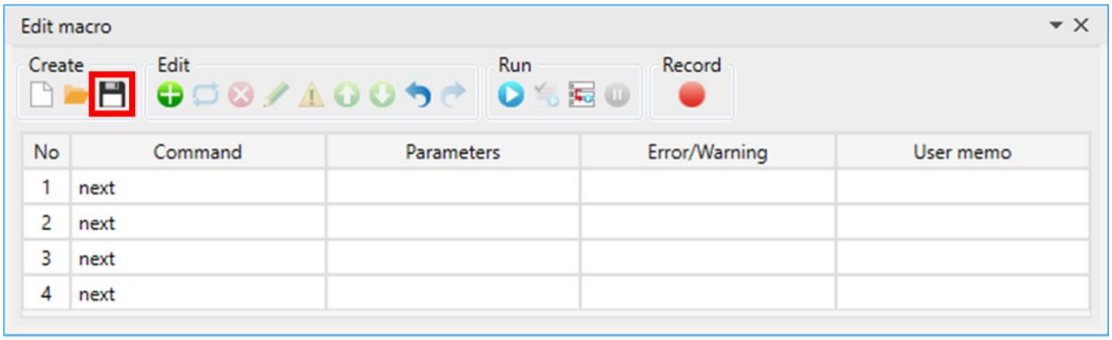


Figure 3-49. Saving the edited macro

3.4.5. Adding a macro loop

This subsection describes the procedure to run the created macro repeatedly.

If no text data is loaded in WinActor Note, load the prepared "Meeting_request.txt" file using "Input tool" according to "3.2.1 Loading a text file."

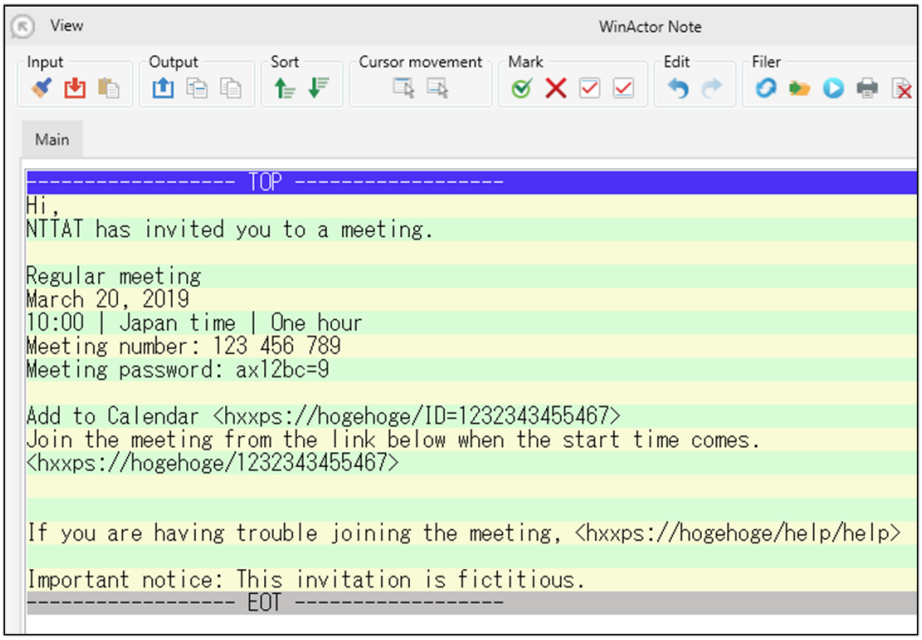


Figure 3-50. Loading Meeting_request.txt

Click "New" in the "Edit macro" pane to clear the macro area.

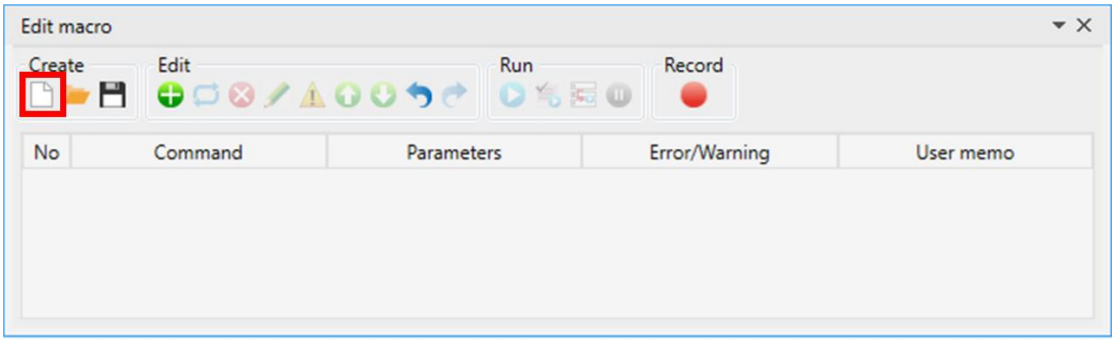


Figure 3-51. Clicking the 'New' button

Click "Record" in the "Edit macro" pane to start recording.

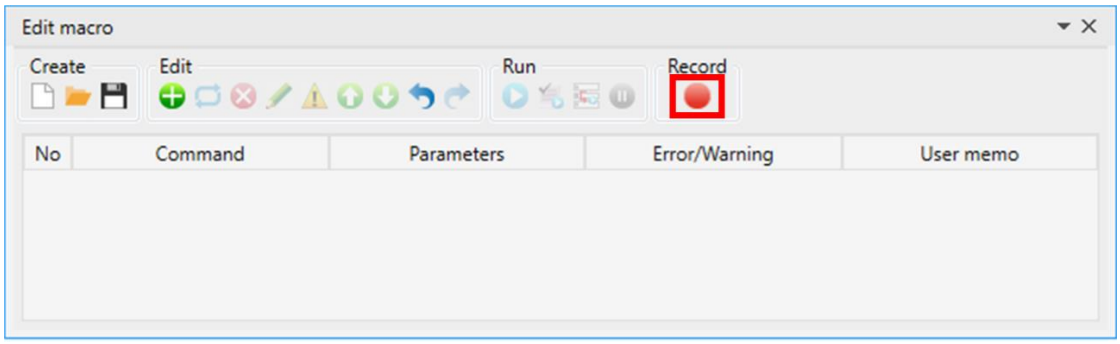


Figure 3-52. Starting the macro recording

Open "Cursor movement tool," specify "Select the next block," and click ‘Execute.’
The operation is recorded and the line with "next" is added to the "Edit macro" pane.

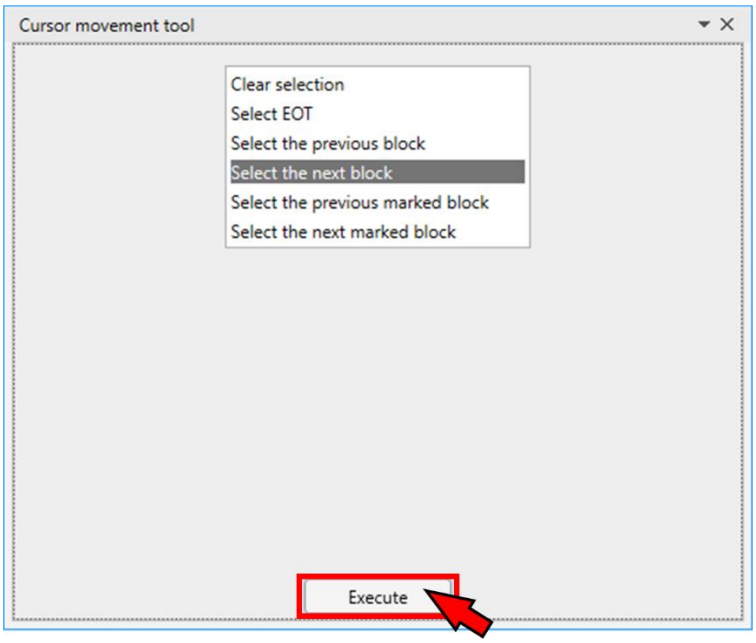


Figure 3-53. Executing "Cursor movement tool"

Click "Record" in the "Edit macro" pane to stop recording.

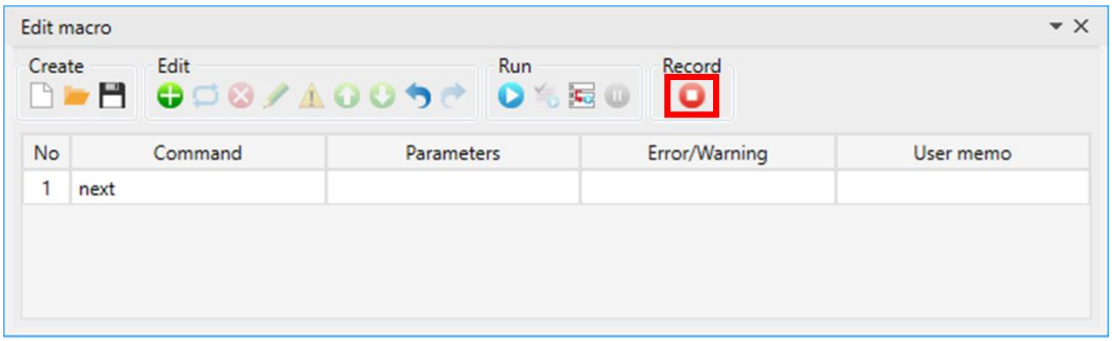


Figure 3-54. Stopping the macro recording

Select the line you want to repeat. Here, select the line with "next."

Click "Add loop" in the "Edit macro" pane to open the "Add loop" dialog.

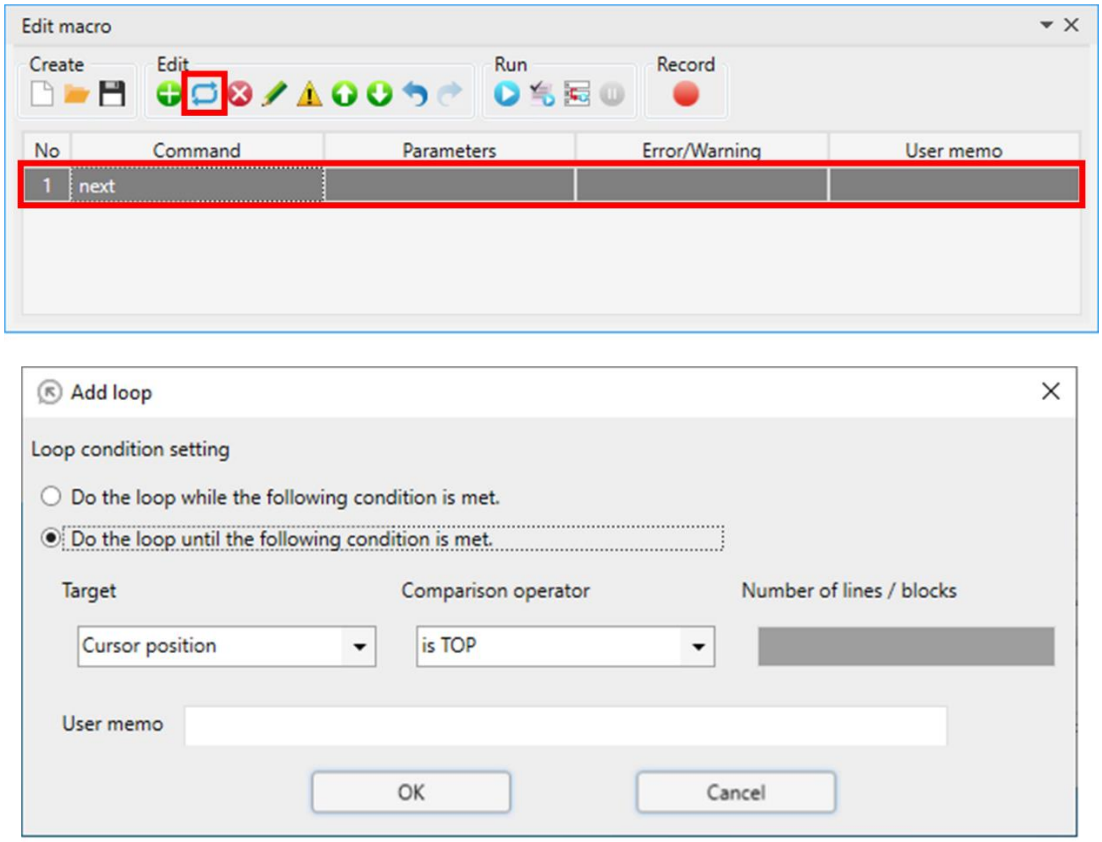


Figure 3-55. Opening the "Add loop" dialog

In the "Add loop" dialog, set the following conditions and click the ‘OK’ button.

- Check "Do the loop while the following condition is met."
- Select "Cursor position" for the target.
- Select "<" for the comparison operator.
- Enter "18" for the number of lines/blocks.

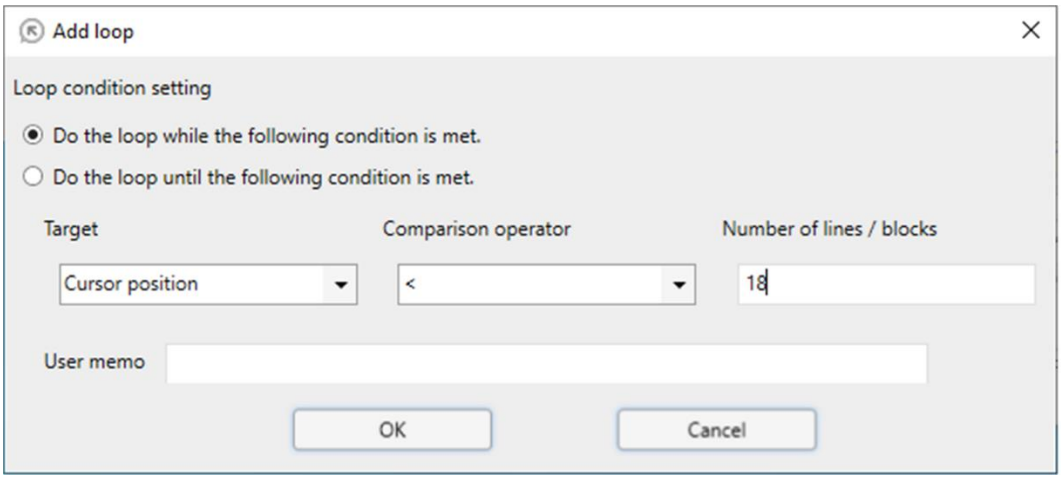


Figure 3-56. "Add loop" dialog

Click "Run all" in the "Edit macro" pane to run the macro.

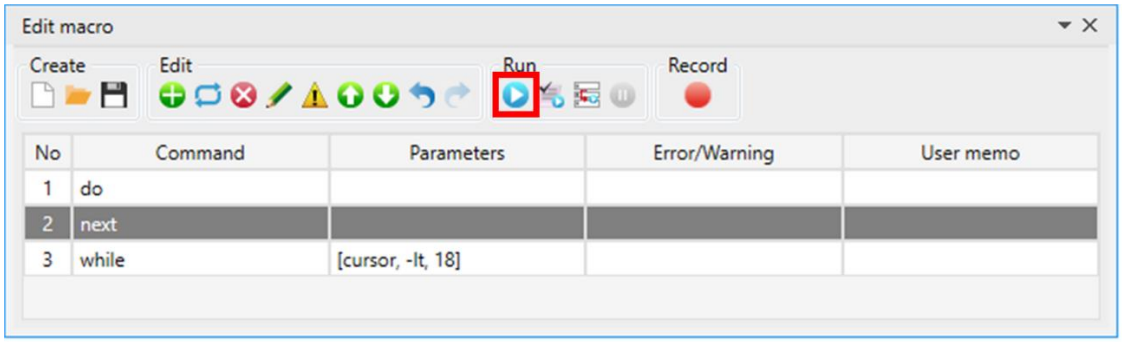


Figure 3-57. Clicking "Run all"

The operation is successful if "Select the next block" of "Cursor movement tool" is executed repeatedly and the cursor moves to "EOT."

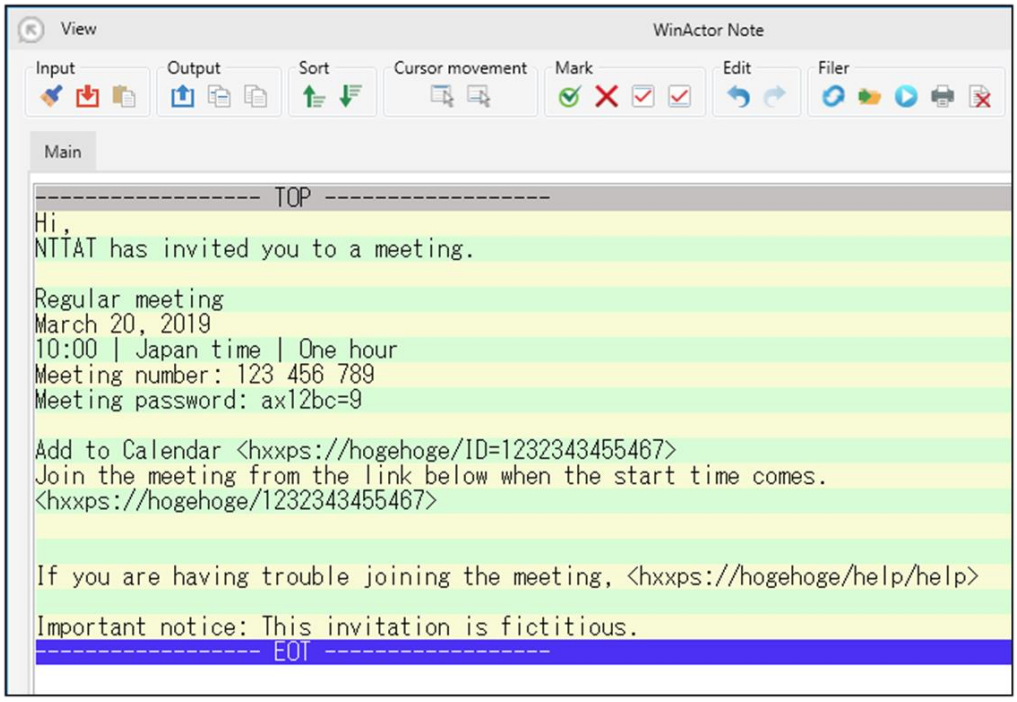


Figure 3-58. Result of running the loop

If the macro gets into an infinite loop and keeps running, you can stop it by clicking "Stop macro."

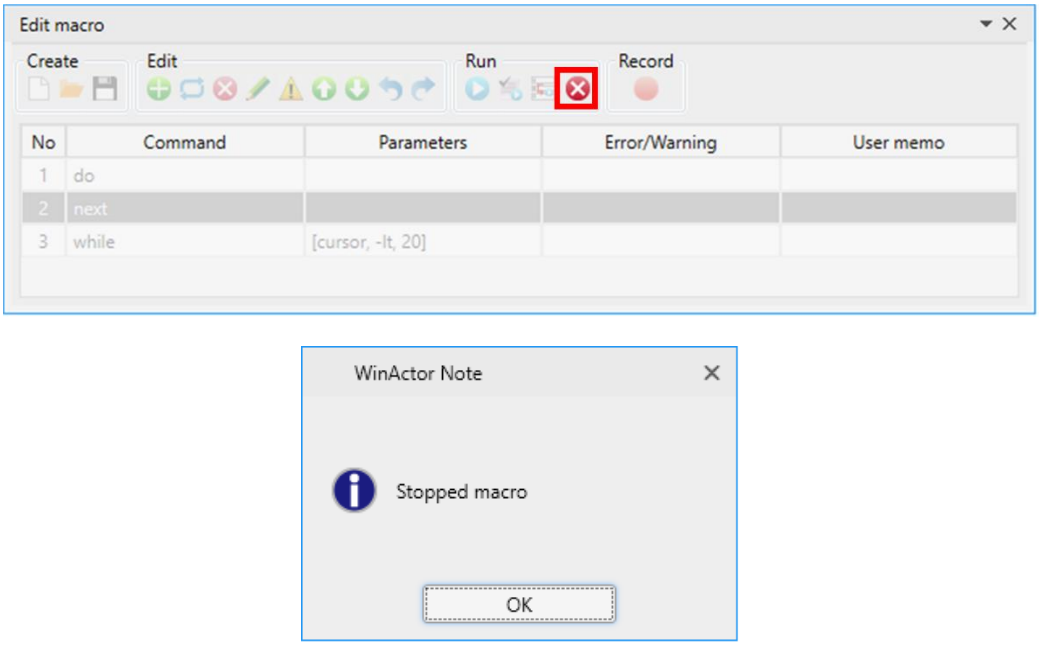


Figure 3-59. Stopping the running macro

That is all for adding the macro loop. Do not save the macro and proceed to the next section.

3.5. Creating a WinActor scenario

3.5.1. Creating a new WinActor scenario

Save a WinActor scenario as a new scenario file in the tutorial folder. Save it with the filename "Read_meeting_request_information." A file handled by a scenario should be saved in the folder where the scenario is saved so that the file can be accessed from the scenario by specifying only the filename (relative path).

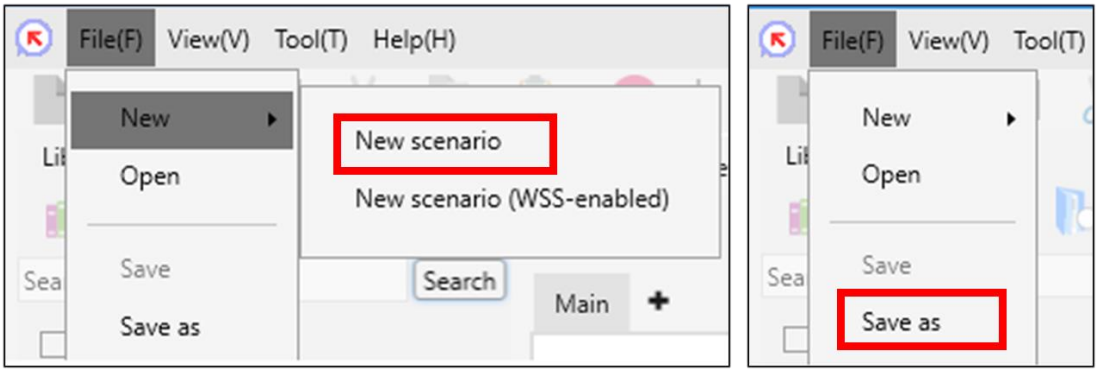


Figure 3-60. Creating a new scenario file

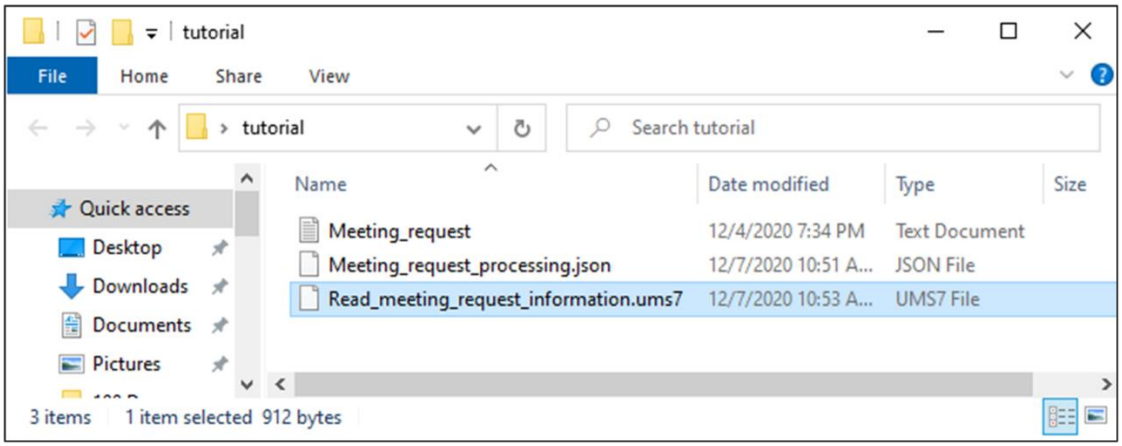


Figure 3-61. Tutorial folder after saving the WinActor scenario

3.5.2. Reading texts by the instruction from WinActor

Create the scenario so that WinActor instructs WinActor Note to read texts.

Place the "Note_ReadTextFile" library in the flowchart area of WinActor.

Library location:

Library -> NTTAT_vx.x.x/25_WinActor_Note/Note_ReadTextFile

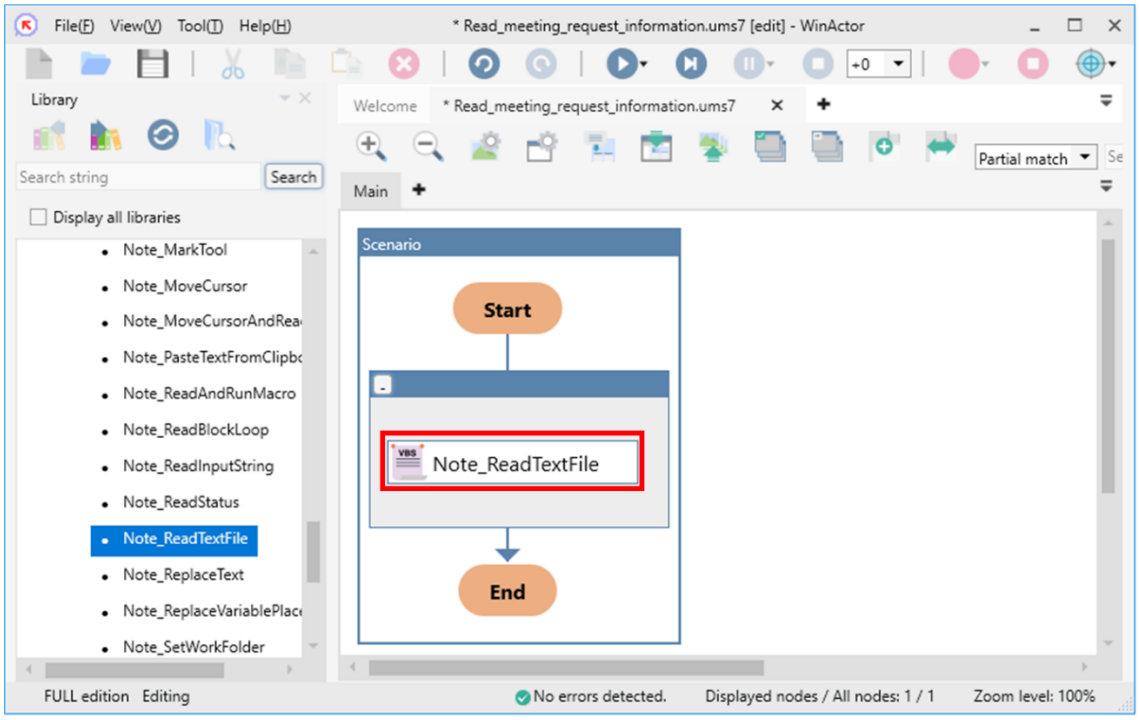


Figure 3-62. Placing the "Note_ReadTextFile" library

Open the "Note_ReadTextFile" library property.

Enter "Meeting_request.txt" in the filename and click the ‘Update’ button.

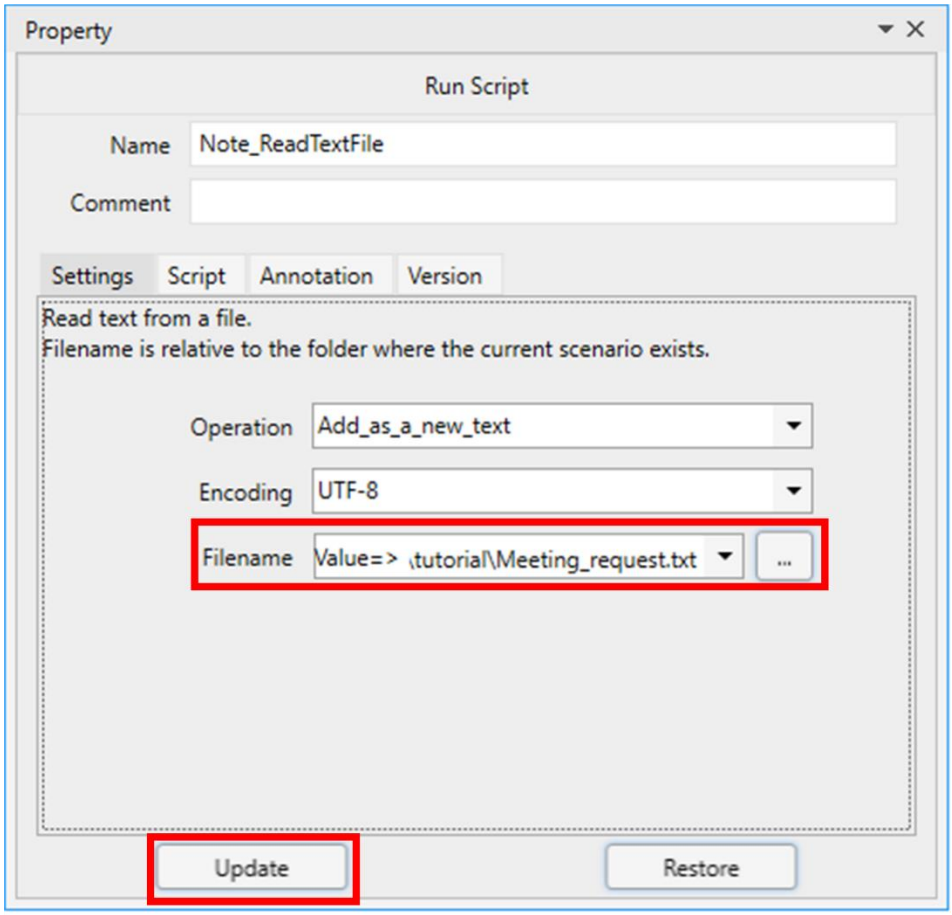


Figure 3-63. Editing the library property

After preparing the WinActor scenario for the text reading part, check the text reading operation once as a trial.

Clear WinActor Note and click the "Run scenario" button on the toolbar of WinActor.

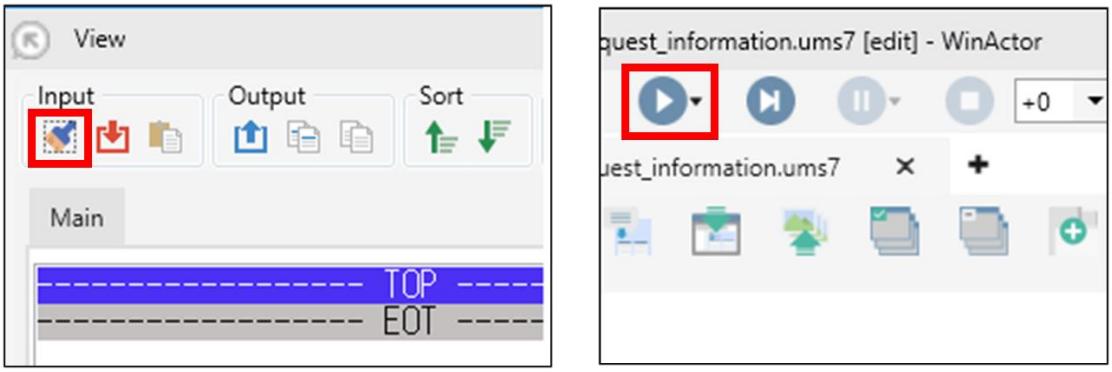


Figure 3-64. Checking the operation

The operation is successful if the texts are read into WinActor Note.

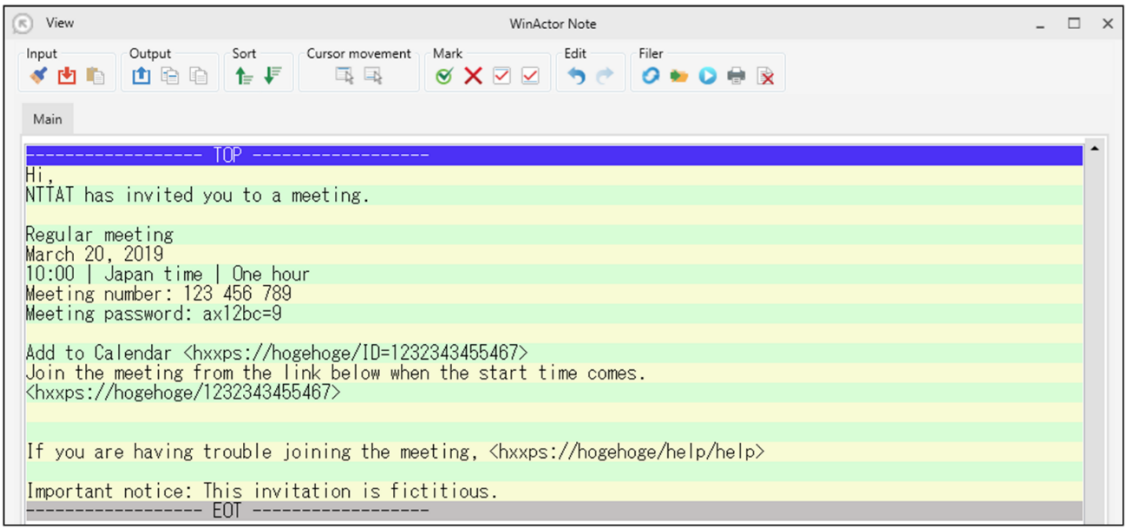


Figure 3-65. Confirming that the file is read into WinActor Note

Save the WinActor scenario frequently.

3.5.3. Processing texts by the instruction from WinActor

Run the WinActor Note macro from WinActor.

Place the "Note_ReadAndRunMacro" library in the flowchart area of WinActor.

Library location:

Library -> NTTAT_vx.x.x/25_WinActorNote/Note_ReadAndRunMacro

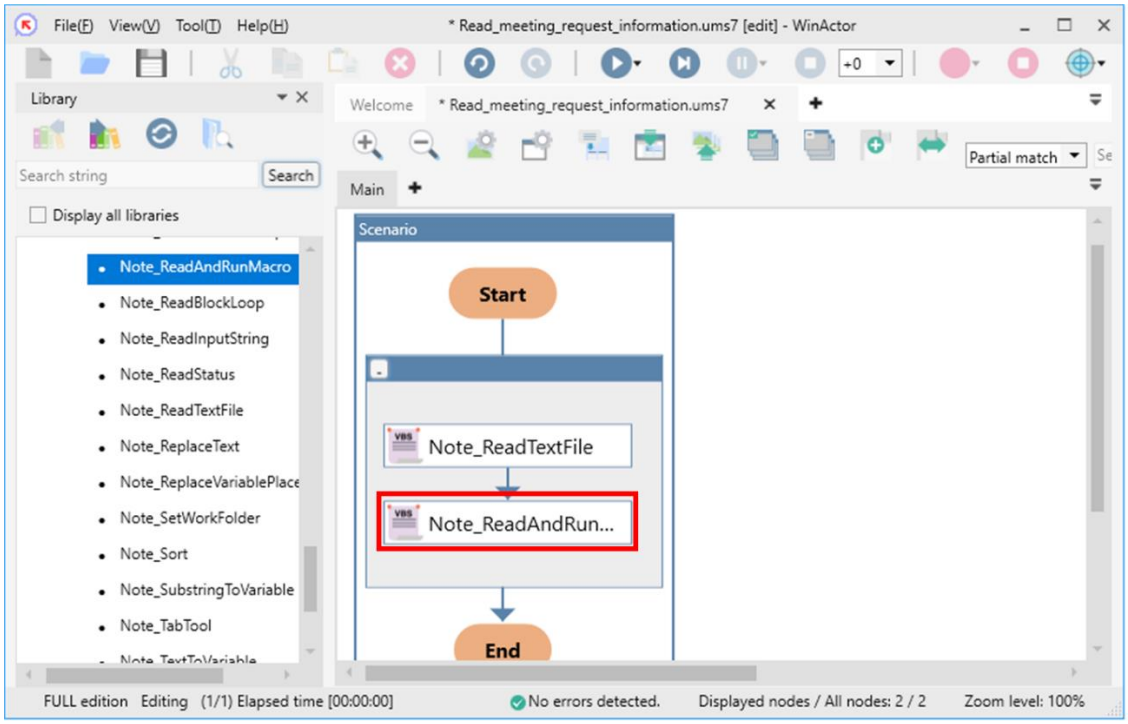


Figure 3-66. Placing the "Note_ReadAndRunMacro" library

Open the "Note_ReadAndRunMacro" library property.
Enter "Meeting_request_processing.json" in the filename and click the ‘Update’ button.

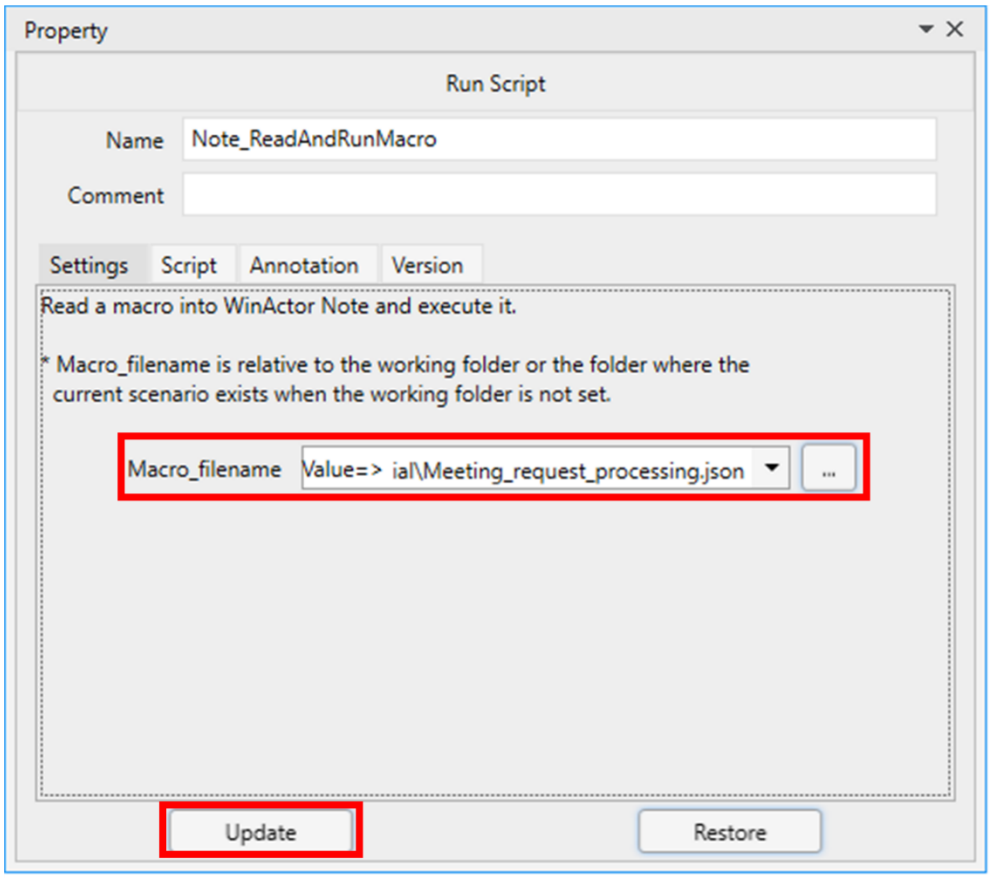


Figure 3-67. Editing the library property

Check the text processing operations.

Clear WinActor Note and click the "Run scenario" button on the toolbar of WinActor.

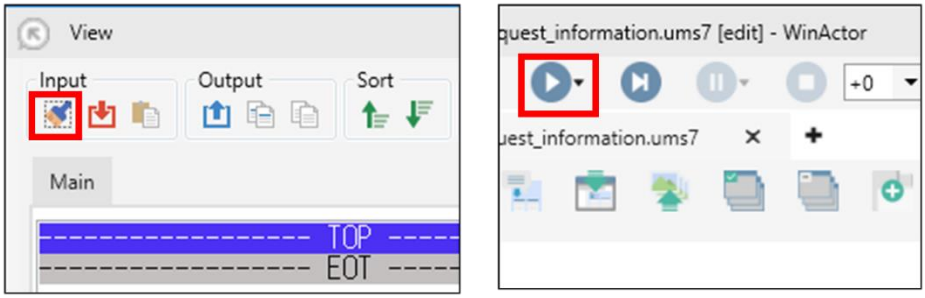


Figure 3-68. Checking the operation

The operations are successful if the text processing result is displayed in the text area of WinActor Note.

The point here is the position of the cursor.

In the next step, the texts will be read in order from the top. Therefore, the cursor should be moved to TOP by the instruction from the WinActor scenario.

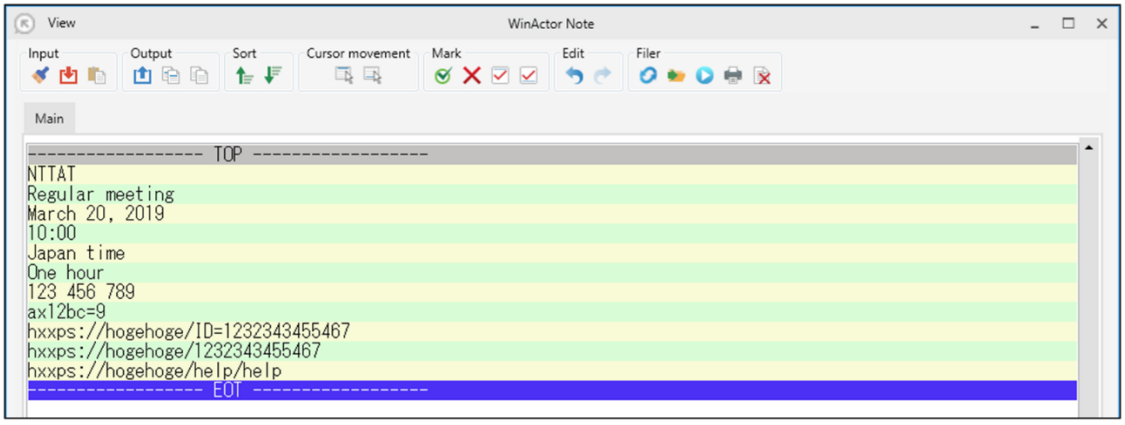


Figure 3-69. Confirming the text processing result on WinActor Note

Save the WinActor scenario frequently.

3.5.4. Moving the cursor to TOP by the instruction from WinActor

Create the scenario so that WinActor instructs WinActor Note to move the cursor to TOP.

Place the "Note_MoveCursor" library in the flowchart area of WinActor.

Library location:

Library -> NTTAT_vx.x.x/25_WinActorNote/Note_MoveCursor

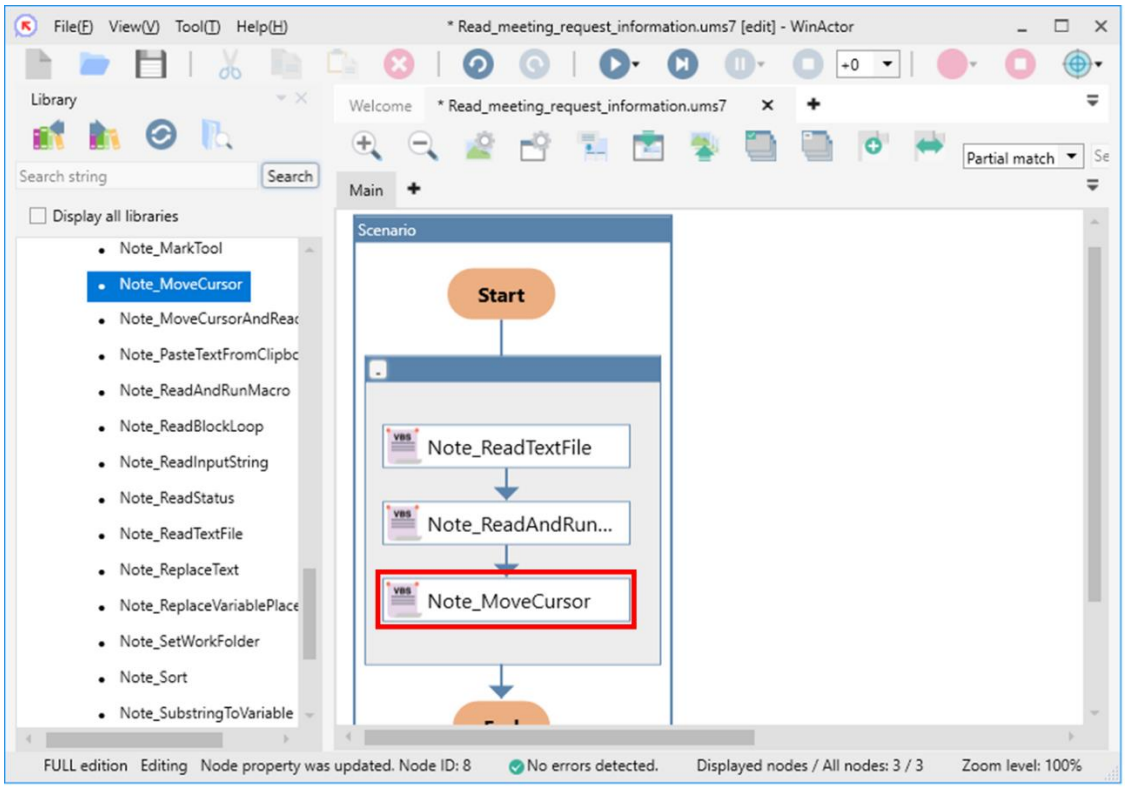


Figure 3-70. Placing the "Note_MoveCursor" library

Open the "Note_MoveCursor" library property.

Select "Clear" for the operation and click the ‘Update’ button.

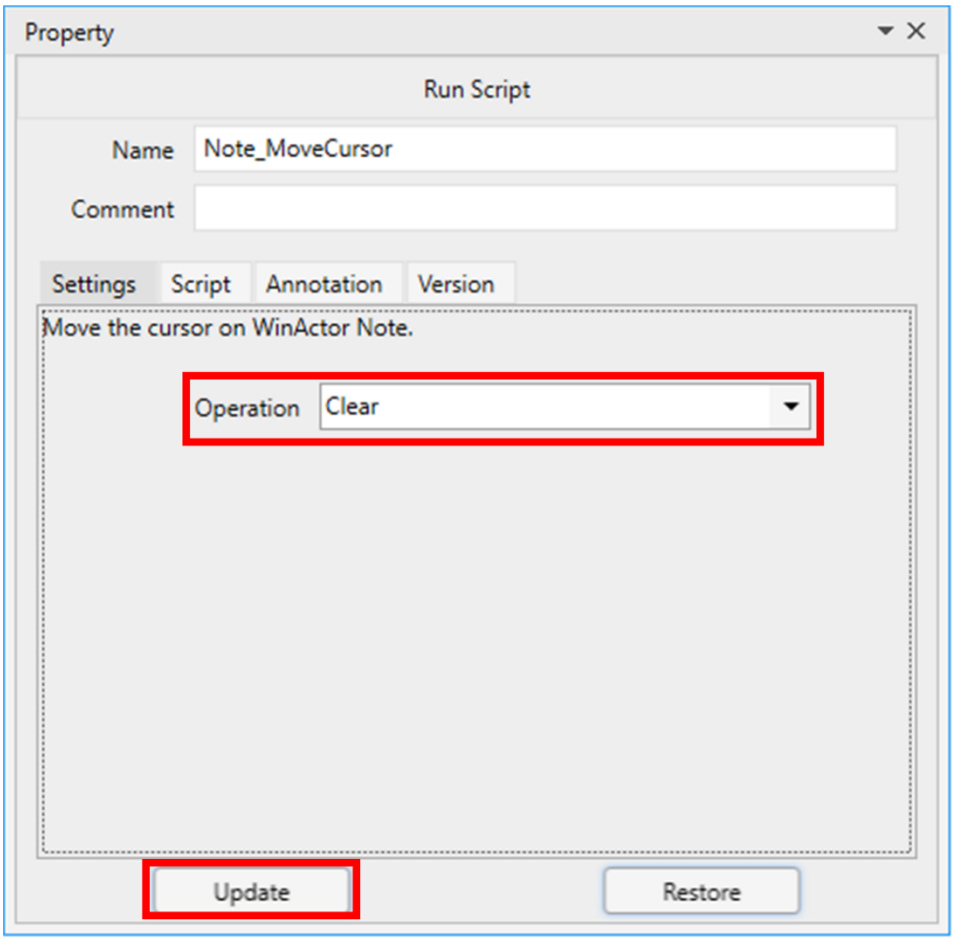


Figure 3-71. Editing the library property

Check the operation for moving the cursor.

Clear WinActor Note and click the "Run scenario" button on the toolbar of WinActor.

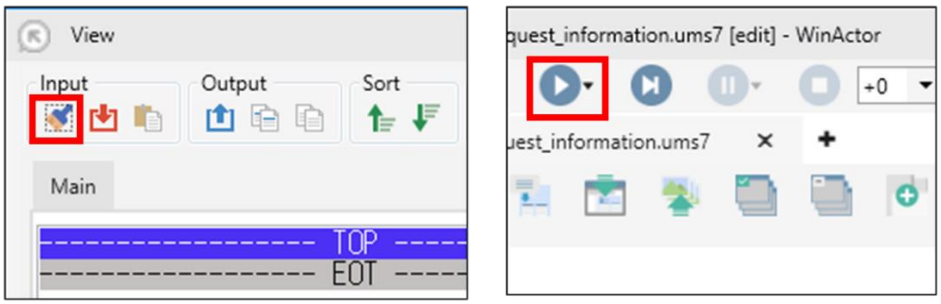


Figure 3-72. Checking the operation

The operation is successful if the texts are processed in the text area of WinActor Note and "Cursor position" on the status bar shows 0.

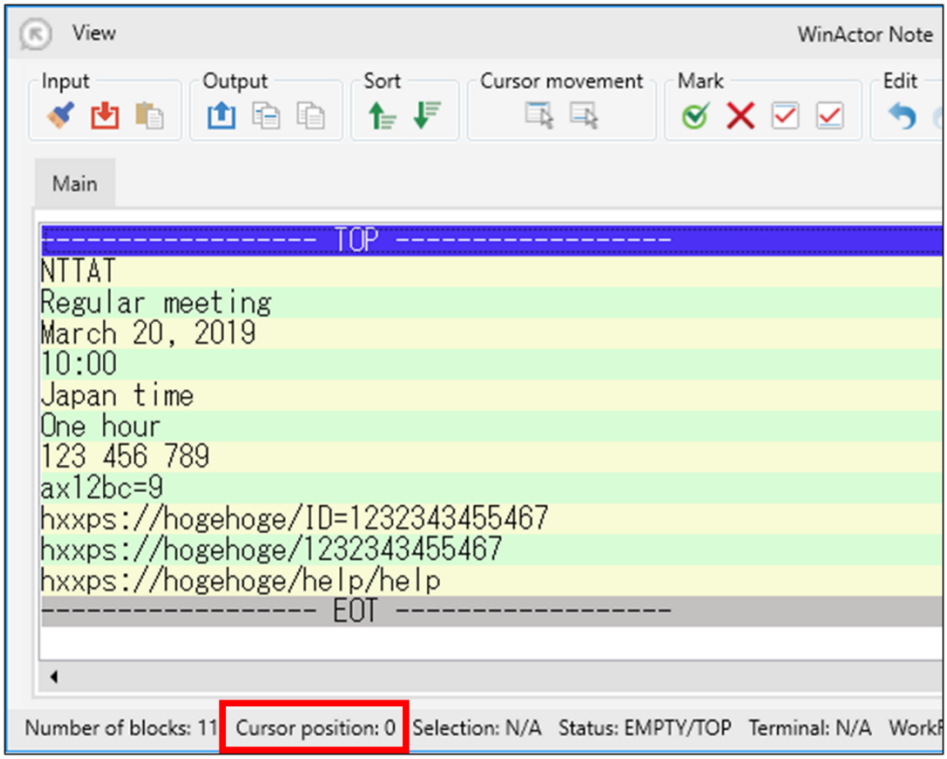


Figure 3-73. Confirming "Cursor position: 0" on WinActor Note

Save the WinActor scenario frequently.

3.5.5. Reading information from WinActor Note

Add the operation to read texts on WinActor Note and save them as variables in WinActor.

Place the "Note_MoveCursorAndReadText" library in the flowchart area of WinActor.

Library location:
Library -> NTTAT_vx.x.x/25_WinActorNote/Note_MoveCursorAndReadText

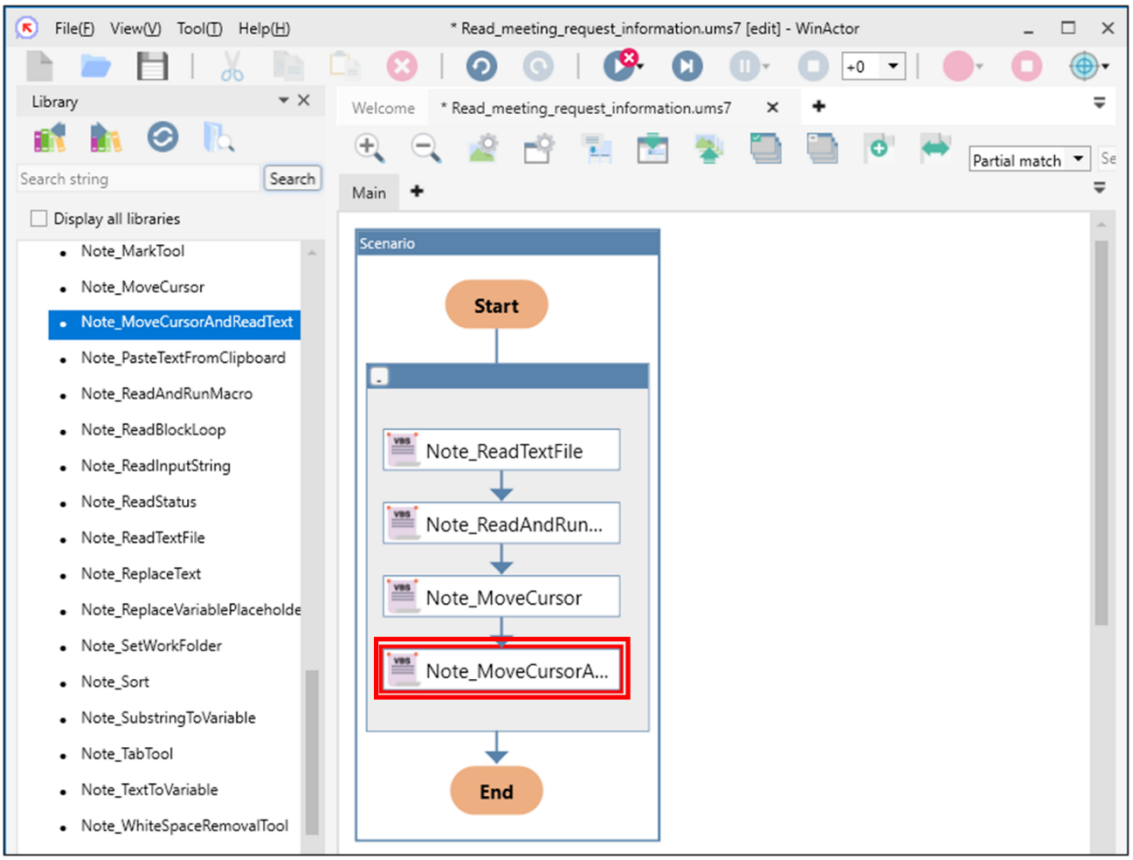


Figure 3-74. Placing the "Note_MoveCursorAndReadText" library

Since the information on the first line of WinActor Note is the "requester" of the meeting, set the information in the first line to be read in the variable "Requester."

Open the "Note_MoveCursorAndReadText" library property.
Specify "Select_the_next_block" for the operation, enter "Requester" for the name and the variable for text, and click the ‘Update’ button.
When the ‘Update’ button is clicked, a new variable "Requester" is registered in the Variable list pane of WinActor.

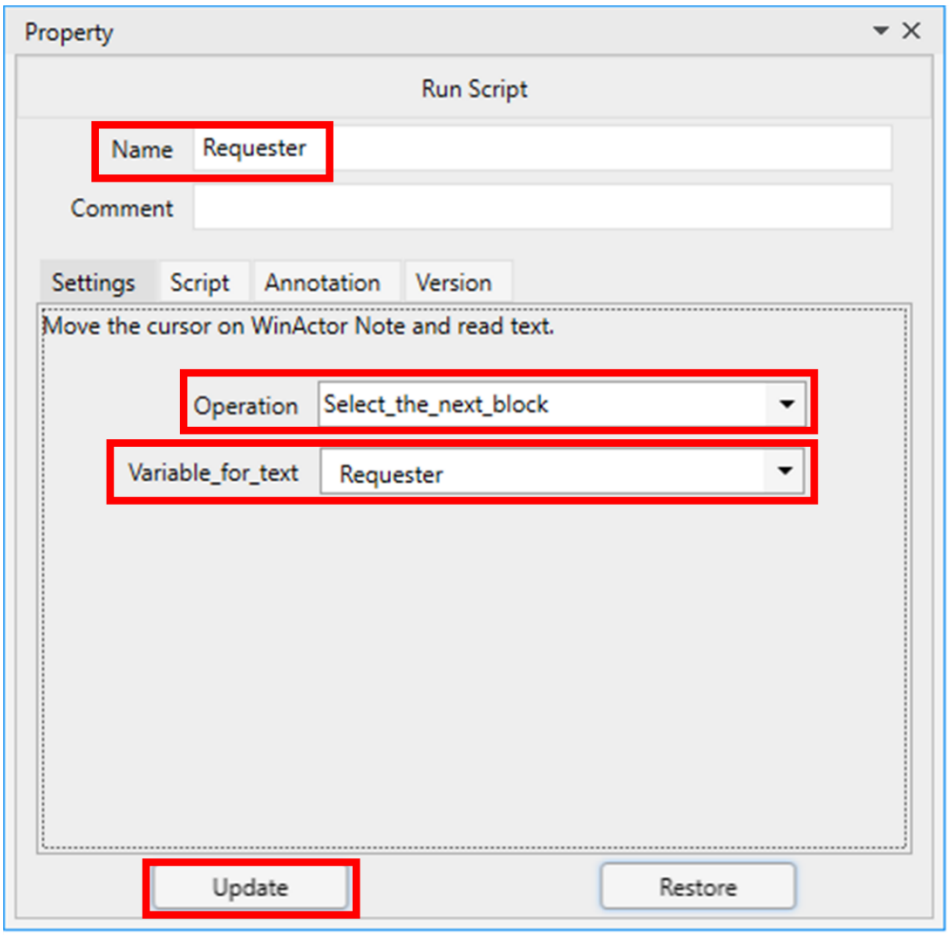


Figure 3-75. Editing the library property

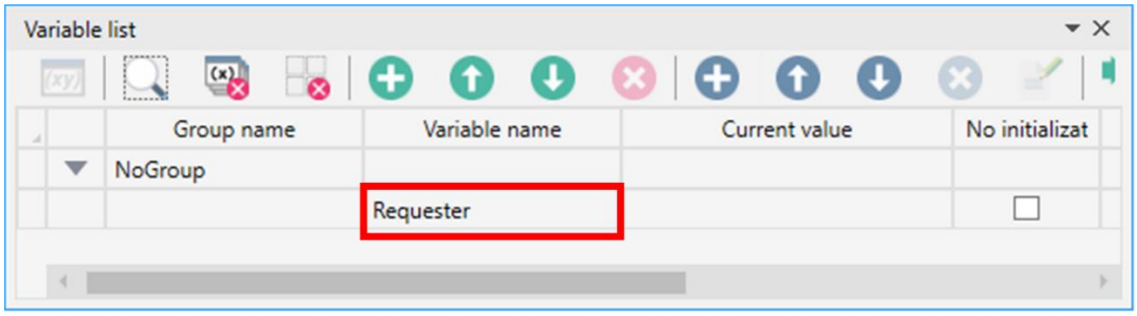


Figure 3-76. "Requester" is registered in the Variable list pane

Save the WinActor scenario frequently.

3.5.6. Practice

The section "3.5.5　Reading information from WinActor Note" describes how to create the scenario for the operation to read the requester information from WinActor Note. In this subsection, create the scenario for the operations to read the information in the block No.2 to 11 with the same procedure.

Table 3-1. Information to be read

Block No.	Variable name	Current value to be inserted at runtime
1	Requester	NTTAT
2	Meeting_name	Regular meeting
3	Date	March 20, 2019
4	Time	10:00
5	Time_zone	Japan time
6	Meeting_duration	One hour
7	Meeting_ID	123 456 789
8	Meeting_password	ax12bc=9
9	URL_for_registration	hxxps://hogehoge/ID=1232343455467
10	URL_to_join_the_meeting	hxxps://hogehoge/1232343455467
11	URL_for_contact	hxxps://hogehoge/help/help

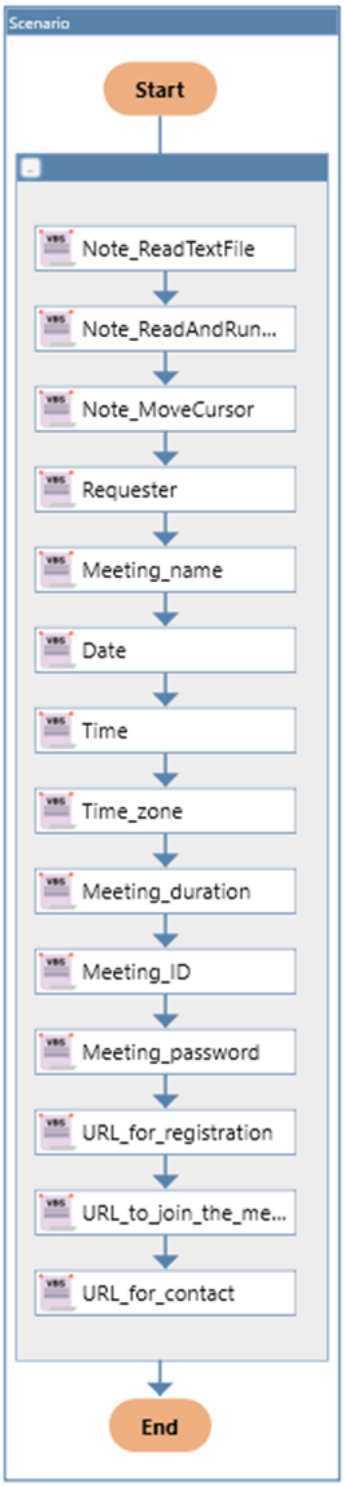
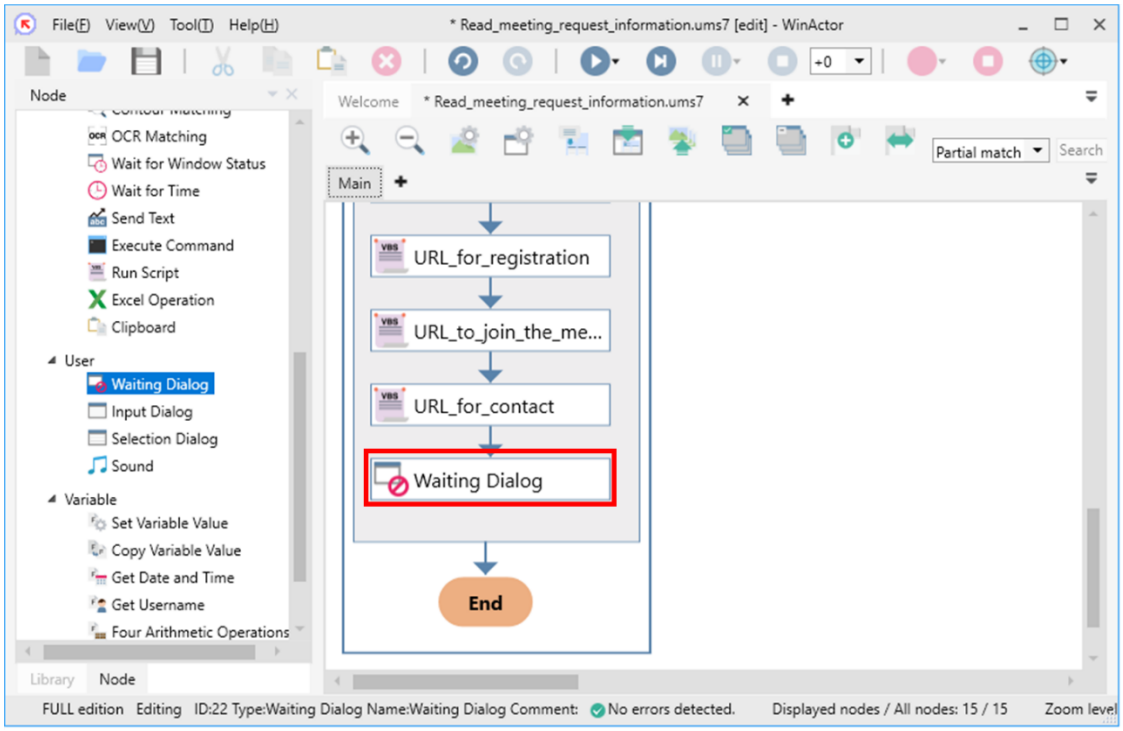


Figure 3-77. Scenario after the practice is completed

3.5.7. Checking the operations for reading information into variables

Place "Waiting Dialog" at the end of the scenario, and click the "Run scenario" button on the toolbar of WinActor.



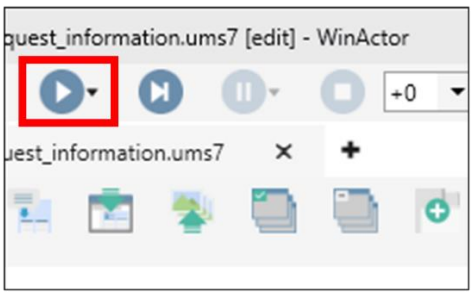


Figure 3-78. Running the text processing results reading

The operations are successful if you can confirm that the information is read into the Variable list pane of WinActor.

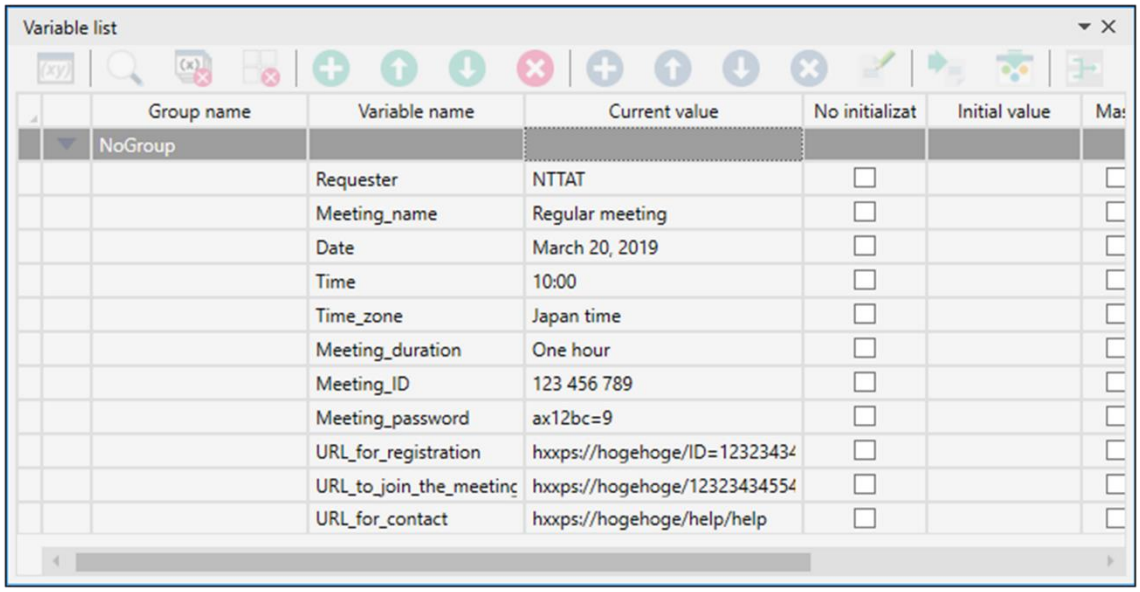


Figure 3-79. Result of reading the text processing results

Lastly, stop the running scenario of WinActor and complete the tutorial.

Figure 3-80. Stopping the WinActor scenario to complete the tutorial

3.6. Operation mode

There are three operation modes for setting the window display of WinActor Note: "Hidden," "View," and "Edit."

It is set to be in the "Hidden" mode when WinActor is launched and in the "Edit" mode when WinActor Note is launched (2. Launching WinActor Note). The operation mode can be changed by using the library "4.19. Note_ChangeMode."

3.6.1. Hidden

Hides the window of WinActor Note.

3.6.2. View

Displays the window of WinActor Note. You can check operations when running a scenario. On the "View" mode window, the menu bar is hidden and the edit operations are disabled.

3.6.3. Edit

Displays the window of WinActor Note. On the "Edit" mode window, the menu bar is displayed and the edit operations are enabled. However, it will be switched to the "View" mode to prevent editing when running a scenario.

3.6.4. Precautions when running a scenario

When running a scenario, resources of WinActor Note will be cleared under the following conditions.

① If the operation mode is not "Edit" at the start of the scenario
② If the operation mode is not "Edit" at the end of the scenario
③ At the end of loop execution using a data list

If the operation mode is "Edit" and you close WinActor Note with the x button at the upper right of the window, the conditions ① and ② become invalid.

4. Library and property list

This chapter introduces the user libraries related to WinActor Note.

4.1. Note_ReadBlockLoop

This library is to read a block one by one from a selected position for each block on WinActor Note.

The text data in the read block will be input to the variable "text" of WinActor and output as a display message.

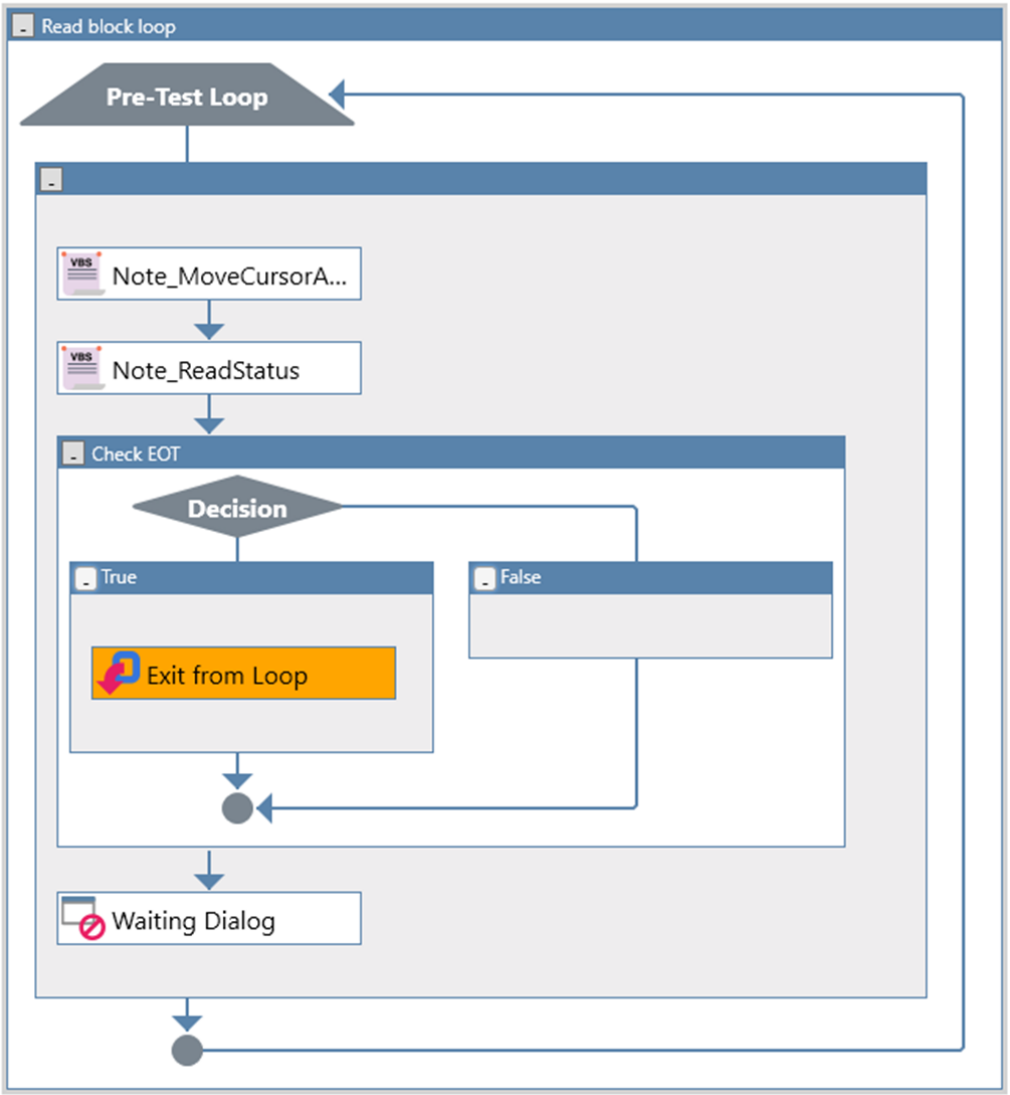


Figure 4-1. Details of the scenario of "Note_ReadBlockLoop"

4.2. Note_MoveCursorAndReadText

This library is to move the cursor on WinActor Note and read a text at the destination.

Table 4-1. "Note_MoveCursorAndReadText" library property items

Property item	Description	
Operation	Select a destination to move the cursor.	
	Select_the_previous_block	The cursor moves to one block above the currently selected block.
	Select_the_next_block	The cursor moves to one block below the currently selected block.
	Select_the_previous_marked_block	The cursor moves to the nearest marked block upward from the currently selected block.
	Select_the_next_marked_block	The cursor moves to the nearest marked block downward from the currently selected block.
Variable_for_text	Specify a variable name where the read text will be input.	

4.3. Note_MoveCursor

This library is to move the cursor on WinActor Note.

Table 4-2. "Note_MoveCursor" library property items

Property item	Description	
Operation	Select a destination to move the cursor.	
	Clear	The cursor moves to TOP (Block 0).
	Select_EOT	The cursor moves to EOT (End Of Text), the block below the last block.
	Select_previous_block	The cursor moves to one block above the currently selected block.
	Select_next_block	The cursor moves to one block below the currently selected block.

	Select_previous_mark	The cursor moves to the nearest marked block upward from the currently selected block.
	Select_next_mark	The cursor moves to the nearest marked block downward from the currently selected block.

4.4. Note_InsertTextFromClipboard

This library is to insert strings on the clipboard into strings in target blocks on WinActor Note.

Table 4-3. "Note_InsertTextFromClipboard" library property items

Property item	Description
Target	Specify target blocks to insert strings.
Format	Specify either "Linewise" or "JSON_format" as a format of texts to be inserted.

4.4.1. Example 1

WinActor Note

```
Hi, {0}.

The meeting room has been booked.
Please join the meeting.

Date: {1}
Time: {2} - {3}
Location: {4}
```

Clipboard (linewise)

```
NTTAT
2019/2/1
10:00
11:00
Tokyo
```

WinActor Note after the insertion

```
Hi, NTTAT.

The meeting room has been booked.
Please join the meeting.

Date: 2019/2/1
Time: 10:00 - 11:00
Location: Tokyo
```

4.4.2. Example 2

WinActor Note

```
HI, {Name}.

The meeting room has been booked.
Please join the meeting.

Date: {Date}
Time: {Start time} - {End time}
Location: {Location}
```

Clipboard (JSON format)

```
{
        "Name" : "NTTAT",
        "Date" : "2019/2/1",
        "Start time" : "10:00",
        "End time" : "11:00",
        "Location" : "Tokyo"
}
```

WinActor Note after the insertion

```
Hi, NTTAT.

The meeting room has been booked.
Please join the meeting.

Date: 2019/2/1
Time: 10:00 - 11:00
Location: Tokyo
```

4.5. Note_PasteTextFromClipboard

This library is to paste texts from the clipboard into WinActor Note.

Table 4-4. "Note_PasteTextFromClipboard" library property item

Property item	Description
Operation	Specify where the texts pasted from the clipboard to be inserted.

4.6. Note_CopyToClipboard

This library is to copy strings in target blocks on WinActor Note to the clipboard.

Table 4-5. "Note_CopyToClipboard" library property items

Property item	Description
Target	Specify blocks to be copied.
Line-break_code	Specify whether to include line breaks at the end of texts or not when copying texts to the clipboard.

4.7. Note_Sort

This library is to sort texts on WinActor Note.

Table 4-6. "Note_Sort" library property item

Property item	Description
Sort_method	Select either "Ascending_order" or "Descending_order" to sort texts based on the text of each block.

4.8. Note_TabTool

This library is to operate a tab of WinActor Note.

Table 4-7. "Note_TabTool" library property items

Property item	Description
Tab_name	Specify a tab to be operated.
Operation	Specify an operation to be performed for the tab.

Table 4-8. Operations in "Note_TabTool"

Operation	Description
Add	Creates a new tab by specifying a tab name. The newly created tab will be selected. An error will occur if a tab with the same name exists.
Select(raise_error_if_not_selectable)	This is to select a tab specified by the tab name so that you can edit the texts in the text area of that tab. An error will occur if the specified tab does not exist.
Select(add_if_not_selectable)	This is to select a tab specified by the tab name so that you can edit the texts in the text area of that tab. If the specified tab does not exist, the tab will be added and selected.
Delete	Deletes a tab specified by the tab name. An error will occur if the specified tab does not exist. The "Main" tab cannot be deleted.
Delete_tabs_other_than_the_main_tab	Deletes all tabs except the "Main" tab. It is not necessary to specify the tab name.

4.9. Note_WriteTextFile

This library is to write strings in target blocks on WinActor Note to a text file.
Specify a filename with a relative path from the folder where the scenario file is located.

Table 4-9. "Note_WriteTextFile" library property items

Property item	Description
Target	Specify a range of blocks for which texts are to be written.
Line-break code	Specify whether to include line breaks at the end of the texts or not when writing the texts to a file.
Line-break_code_type	Specify a type of the line-break code for the texts to be written to a file.
Encoding	Specify an encoding for the texts to be written to a file.
Filename	Specify a name of a text file to write the texts to.

4.10. Note_ReadTextFile

This library is to read a text file into WinActor Note.
Specify a filename with a relative path from the folder where the scenario file is located.

Table 4-10. "Note_ReadTextFile" library property items

Property item	Description
Operation	Specify where the texts read from a text file to be inserted.
Encoding	Specify an encoding for reading the texts.
Filename	Specify a name of a text file to be read.

4.11. Note_InsertTextFromFile

This library is to insert strings from a file into strings in target blocks on WinActor Note.

Table 4-11. "Note_InsertTextFromFile" library property items

Property item	Description
Target	Specify target blocks to insert strings.
Format	Specify either "Linewise" or "JSON_format" as a format of the texts to be inserted.
Filename	Specify a name of a text file to be inserted.

	(Referenced by a relative path from the folder where the scenario file is located)
Encoding	Specify an encoding for reading the texts to be inserted.

4.11.1. Example 1

WinActor Note

```
Hi, {0}.

The meeting room has been booked.
Please join the meeting.

Date: {1}
Time: {2} - {3}
Location: {4}
```

File (linewise)

```
NTTAT
2019/2/1
10:00
11:00
Tokyo
```

WinActor Note after the insertion

```
Hi, NTTAT.

The meeting room has been booked.
Please join the meeting.

Date: 2019/2/1
Time: 10:00 - 11:00
Location: Tokyo
```

4.11.2. Example 2

WinActor Note

```
Hi, {Name}.

The meeting room has been booked.
Please join the meeting.

Date: {Date}
Time: {Start time} - {End time}
Location: {Location}
```

File (JSON format)

```
{
        "Name" : "NTTAT",
        "Date" : "2019/2/1",
        "Start time" : "10:00",
        "End time" : "11:00",
        "Location" : "Tokyo"
}
```

WinActor Note after the insertion

```
Hi, NTTAT.

The meeting room has been booked.
Please join the meeting.

Date: 2019/2/1
Time: 10:00 - 11:00
Location: Tokyo
```

4.12. Note_BlockSplitTool

This library is to split blocks on WinActor Note that meet specified conditions.

Table 4-12. "Note_BlockSplitTool" library property items

Property item	Description
Target	Specify a range of blocks to be split.
Keyword	Specify a keyword to be used as a position to split. (It is not referenced when "Split_with_line_breaks(No_keyword)" is specified for the split method.)
Split_method	Specify which of the implemented split methods you want to use to split.

4.12.1. Example 1

WinActor Note

Apple, Orange, Banana

Split operation

Keyword: ","　Split method: "Split_immediately_after_keywords"

WinActor Note after the split

Apple, Orange, Banana

4.12.2. Example 2

WinActor Note

Apple, Orange, Banana

Split operation

Keyword: ","　Split method: "Split_with_keywords_and_delete_them"

WinActor Note after the split

Apple Orange Banana

4.13. Note_BlockExtractionTool

This library is to extract targeted blocks on WinActor Note.

Table 4-13. "Note_BlockExtractionTool" library property items

Property item	Description
Target	Specify a range of blocks to be extracted.
Operation	Specify an operation (delete, copy, merge blocks, etc.) to be performed for the specified target blocks.

4.14. Note_BlockSearchTool

This library is to search blocks on WinActor Note that meet specified conditions.

Table 4-14. "Note_BlockSearchTool" library property items

Property item	Description
Keyword	Specify a string to be searched.
Search_condition	Specify a search condition for the entered keyword.
Search_method	Specify a search range for the specified keyword or an operation to be performed on the blocks that match the search condition.

4.15. Note_BlockNumberDesignationTool

This library is to specify a block by number and perform an operation on the specified block on WinActor Note. You can specify a range by entering numbers as "2-5,3-6,-2,3-."

* If "2-5" is specified, the block numbers from 2 to 5 will be the operation target. If "-2" is specified, the block number 2 and lower will be the operation target. If "3-" is specified, the block number 3 and higher will be the operation target.
* You cannot specify a range when specifying "Select" for the operation.

Table 4-15. "Note_BlockNumberDesignationTool" library property items

Property item	Description
Block_number	Specify a number for a block to be operate. (Multiple blocks can be specified by separating numbers with ",")
Operation	Specify an operation to be performed for the specified block(s).

4.16. Note_BlockMergeTool

This library is to merge blocks on WinActor Note that meet specified conditions.

Table 4-16. "Note_BlockMergeTool" library property items

Property item	Description
Keyword	Specify a keyword that serves as a separator for merging blocks.
Search_condition	Specify a condition for the specified keyword for merging blocks. (Example: Merge blocks that "Contain" the keyword)

4.17. Note_ReadAndRunMacro

This library is to read a macro into WinActor Note and run the macro.

Table 4-17. "Note_ReadAndRunMacro" library property item

Property item	Description
Macro_filename	Specify a name of a macro file to be read with an absolute or relative path. If a relative path is specified, it will be a relative path from the following folder. If a work folder is set: Work folder. If a work folder is not set: The folder where the scenario currently in progress is saved.

4.18. Note_MarkTool

This library is to mark or unmark blocks on WinActor Note.

Table 4-18. "Note_MarkTool" library property item

Property item	Description
Operation	Specify whether to mark or unmark the block at the current cursor position, or specify one of the operations based on whether the blocks are marked or not.

4.19. Note_ChangeMode

This library is to change the operation mode of WinActor Note. For each operation mode, see "3.6 Operation mode."

Table 4-19. "Note_ChangeMode" library property item

Property item	Description
Mode	Hidden ... Changes the operation mode to "Hidden." View ... Changes the operation mode to "View." Edit ... Changes the operation mode to "Edit." It switches to the "View" mode to prevent editing when a scenario is running.

4.20. Note_ReplaceVariablePlaceholder

This library is to insert strings of WinActor variables into strings in target blocks on WinActor Note.

Table 4-20. "Note_ReplaceVariablePlaceholder" library property item

Property item	Description
Target	Specify target blocks to insert variables.

4.20.1. Example

WinActor Note

Hi, {Name}.

The meeting room has been booked.
Please join the meeting.

Date: {Date}
Time: {Start time} - {End time}
Location: {Location}

Variables on WinActor

Variable name	Current value
Name	NTTAT
Date	2019/2/1
Start time	10:00
End time	11:00
Location	Tokyo

WinActor Note after the insertion

Hi, NTTAT.

The meeting room has been booked.
Please join the meeting.

Date: 2019/2/1
Time: 10:00 - 11:00
Location: Tokyo

4.21. Note_TextToVariable

This library is to import a string in target blocks on WinActor Note into a specified variable.

Table 4-21. "Note_TextToVariable" library property items

Property item	Description
Target	Specify blocks for which a text is to be imported to a variable.
Line-break_code	Specify whether to include a line break at the end of the text or not when getting the texts to be imported.
Line-break_code_type	Specify a line-break code when the text to be imported contain a line break.
Variable_for_text	Specify a variable to which the acquired text to be output.

4.22. Note_SubstringToVariable

This library is to import a part of text on WinActor Note into a specified variable.

Table 4-22. "Note_SubstringToVariable" library property items

Property item	Description
Start_character	Specify a range of characters to be imported.
End_character	Specify a range of characters to be imported.
Variable_for_substring	Specify a variable to which the acquired text to be output.

4.23. Note_ReadStatus

This library is to get the state of WinActor Note.

Table 4-23. "Note_ReadStatus" library property items

Property item	Description
Selection	Stores "false" when the cursor is at TOP (above the first line) or EOT (below the last line), and stores "true" when the cursor is at the text part.
Cursor_position	Stores the numerical value of the block number where the cursor is positioned. 0 when the cursor is at TOP. The number of blocks + 1 when the cursor is at EOT.
Number_of_blocks	Stores the number of blocks of text currently loaded into WinActor Note in the numerical value.
MARK	Stores "true" if the cursor is at the marked block and "false" otherwise. Stores "false" when the cursor is at TOP or EOT.
EMPTY	Stores "true" if the text in the cursor position is empty and "false" otherwise. Stores "true" when the cursor is at TOP or EOT.
TOP	Stores "true" if the cursor is at TOP and "false" otherwise.
EOT	Stores "true" if the cursor is at EOT and "false" otherwise.

4.24. Note_WhiteSpaceRemovalTool

This library is to remove white spaces in target blocks on WinActor Note.

Table 4-24. "Note_WhiteSpaceRemovalTool" library property items

Property item	Description
Target	Specify a range of blocks for which white spaces are to be removed.
Operation	Specify a type of operation to remove white spaces (Example: Remove_white_spaces, Remove_line_breaks, Remove_the_leading_white_spaces, etc.).

4.25. Note_EditTool

This library is to edit target blocks on WinActor Note.

Table 4-25. "Note_EditTool" library property items

Property item	Description
Target	Specify blocks for editing.
Keyword	Specify a keyword to be used in a condition when performing a specified operation.
Operation	Specify a type of edit operation to be performed for the targeted blocks.

Table 4-26. Edit operations in "Note_EditTool"

Edit operation	Description
Delete_to_the_left	Deletes text data (including the keyword part) located to the left of the specified keyword. If the specified keyword appears more than once, the text data to the left of the first (leftmost) keyword will be deleted.
Delete_to_the_right	Deletes text data (including the keyword part) located to the right of the specified keyword. If the specified keyword appears more than once, the text data to the right of the last (rightmost) keyword will be deleted.
Delete_to_the_left(excluding_keyword)	Deletes text data (not including the keyword part) located to the left of the specified keyword. If the specified keyword appears more than once, the text data to the left of the first (leftmost) keyword will be deleted.
Delete_to_the_right(excluding_keyword)	Deletes text data (not including the keyword part) located to the right of the specified keyword. If the specified keyword appears more than once, the text data to the right of the last (rightmost) keyword will be deleted.
Leave_the_text_of_inside_parentheses	Leaves only a text inside parentheses (eg, "", (), <>, etc.) specified as a keyword and deletes the rest of the text including parentheses.
Leave_the_parenthesized_text	Leaves a text enclosed in parentheses (eg "", (), <>, etc.) specified as a keyword and deletes the rest of the text.
Delete_the_parenthesized_text	Deletes a text enclosed in parentheses (eg "", (), <>, etc.) specified as a keyword and leaves the rest of the text.
Delete_to_the_left_of_the_parentheses	Deletes all text data to the left of a text enclosed in parentheses (eg "", (), <>, etc.) specified as a keyword.
Delete_to_the_right_of_the_parentheses	Deletes all text data to the right of a text enclosed in parentheses (eg "", (), <>, etc.) specified as a keyword.
Delete_all_to_the_left	Deletes text data (including the keyword part) located to the left of the specified keyword. If the specified keyword appears more than once, the text data to the left of the last (rightmost) keyword will be deleted.
Delete_all_to_the_right	Deletes text data (including the keyword part) located to the right of the specified keyword. If the specified keyword appears more than once, the text data to the right of the first (leftmost) keyword will be deleted.
Add_to_the_beginning	Adds a text specified as a keyword at the beginning of a text of each target block.
Add_to_the_end	Adds a text specified as a keyword at the end of the text of each target block.

Parenthesize	Encloses the entire text data of each target block using parentheses (eg "", (), <>, etc.) specified as a keyword.

4.26. Note_ReplaceText

This library is to replace a string in target blocks on WinActor Note.

Table 4-27. "Note_ReplaceText" library property items

Property item	Description
Target	Specify a range of blocks for which a string is to be replaced.
Regular_expression	Specify whether to use a regular expression when entering a string before replacement.
Before	Specify a string to be replaced.
After	Specify a string after replacement.

4.27. Note_SetWorkFolder

This library is to set a work folder of WinActor Note.

Table 4-28. "Note_SetWorkFolder" library property items

Property item	Description
Specify_the_folder	Select a folder to be set as a work folder from "Folder_name" (any folder) and "Scenario_folder." If "Scenario_folder" is selected, the folder (scenario folder) where the scenario currently in progress is saved will be specified.
Folder_name	Specify a folder to be set as a work folder with an absolute or relative path. If a relative path is specified, the scenario folder will be the starting point. If "Scenario_folder" is selected in Specify_the_folder, the input contents will be replaced with the scenario folder.

5. Docking window

For descriptions of the docking window, see "WinActor Note Operation Manual."

6. Reference materials

Table 6-1 shows the materials referenced in this manual.

Table 6-1. Reference materials

No.	Material name
1	WinActor Operation Manual
2	WinActor User Library Sample Manual
3	WinActor Note Operation Manual
4	WinActor Note Terminal Function Scenario Creation Manual

WinActor Note

Text Processing Scenario Creation Manual

NTT ADVANCED TECHNOLOGY CORPORATION

WA7-N- 20250603