

# WinActor Note

# Terminal Function Scenario Creation Manual

# NTT ADVANCED TECHNOLOGY CORPORATION

Copyright © 2013-2025 NTT, Inc. & NTT ADVANCED TECHNOLOGY CORPORATION

#### Contents

1.	Intro	oduc	tion	1
1	.1.	Abo	ut this document	1
1	.2.	Trac	demarks	2
1	.3.	Not	es on this manual	2
2.	Sce	nario	o creation tutorial	3
2	.1.	Pre	paration	3
2	2.2.	Hov	v to create a scenario	4
2	2.3.	Sce	nario creation procedure	5
2	2.4.	Sce	nario overview	8
2	2.5.	Cre	ating the WinActor Note macro	9
	2.5.	1.	Clearing the data on the WinActor Note window	9
	2.5.	2.	Processing using "SSH tool"	. 10
	2.5.	3.	Text processing	. 14
	2.5.	4.	Saving the text data on the WinActor Note window to a file	. 16
	2.5.	5.	Saving the macro	. 17
2	2.6.	Cre	ating the WinActor scenario	. 18
	2.6.	1.	Calling the created macro	. 19
	2.6.	2.	Adding the processing of judgment	. 22
	2.6.	3.	Adding the processing of receiving the file	. 28
	2.6.	4.	Adding the processing of creating the folder for saving logs	. 32
	2.6.	5.	Adding the processing of moving the log file	. 47
2	2.7.	Che	cking the operations	. 55
3.	Libr	ary a	nd property list	. 56
3	5.1.	She	II tool	. 56
	3.1.	1.	Note_OpenPowerShell	. 58
	3.1.	2.	Note_OpenCommandPrompt	. 58
	3.1.	3.	Note_ExecuteCommand	. 58
	3.1.	4.	Note_ClosePowerShellOrCommandPrompt	. 58
3	5.2.	SSF	ł tool	. 59
	3.2.	1.	Connection settings	. 61
	3.2.	2.	Note_OpenSSHClient	. 66
	3.2.	3.	Note_OpenSSHClient(KnownHosts)	. 67
	3.2.	4.	Note_ExecuteCommand(SSHClient)	. 68
	3.2.	5.	Note_CloseSSHClient	. 68

	3.2.6.	Note_SendFile(SCP)	. 69
	3.2.7.	Note_SendFile(SCP_KnownHosts)	. 70
	3.2.8.	Note_ReceiveFile(SCP)	. 71
	3.2.9.	Note_ReceiveFile(SCP_KnownHosts)	. 72
	3.2.10.	Note_SendControlCode(SSHClient)	. 73
3	3.3. Telr	net tool	. 74
	3.3.1.	Note_OpenTeInetClient	. 75
	3.3.2.	Note_ExecuteCommand(TelnetClient)	. 76
	3.3.3.	Note_CloseTeInetClient	. 76
	3.3.4.	Note_SendControlCode(TelnetClient)	. 76
4.	Docking	window	. 78
5.	Referen	ce materials	. 79

# 1. Introduction

#### 1.1. About this document

This document is the manual for creating scenarios for terminal functions that work in cooperation with WinActor Note.

The following terminal functions are provided:

- <u>Windows shell function</u> Executes commands on Windows PowerShell or Command Prompt
- SSH/SCP client functions

Connects to, executes commands on, and disconnects from an SSH server Sends/receives a file with an SSH server (SCP)

Telnet client function

Connects to, executes commands on, and disconnects from a Telnet server

By using the terminal functions in combination with the macro functions of WinActor Note, it is possible to perform detailed controls according to the result of executing commands.

In Chapter 2, you will create a scenario to execute commands by connecting to an SSH server built on CentOS and get log files according to the output result of the commands. Through the tutorial in this manual, you can learn the terminal functions of WinActor Note.

For how to use WinActor Note, see the materials No.3 and 4 in Table 5-1. Reference materials.

# 1.2. 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.

# 1.3. 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. Scenario creation tutorial

# 2.1. Preparation

The environment used in this tutorial is shown in Table 2-1. Environment used in this tutorial. It is assumed that the SSH protocol connection confirmation has been completed in advance between the SSH server and the computer on which a WinActor scenario runs.

Target	Item	Description	Settings in this tutorial
Server	Server OS	CentOS Linux	-
		release 7.6.1810	
	SSH server	OpenSSH 7.4p1	IP address: 192.168.56.2
			Port number: 22
	Operation verification	Any	Username: user
	account		Password: ax12bc=9
Computer	IP address	Any	IP address: 192.168.56.1
on which	Scenario folder	Any	C:\Terminal_function_scenario
WinActor	Password file	Any	Store the file in the scenario
runs		However, store the	folder with the password:
		file in the scenario	ax12bc=9 and the filename:
		folder.	cihperPassword.json.
			For details, see 2.3.
	Windows	Edition:	
		Windows 10 Pro	
		Japanese version	

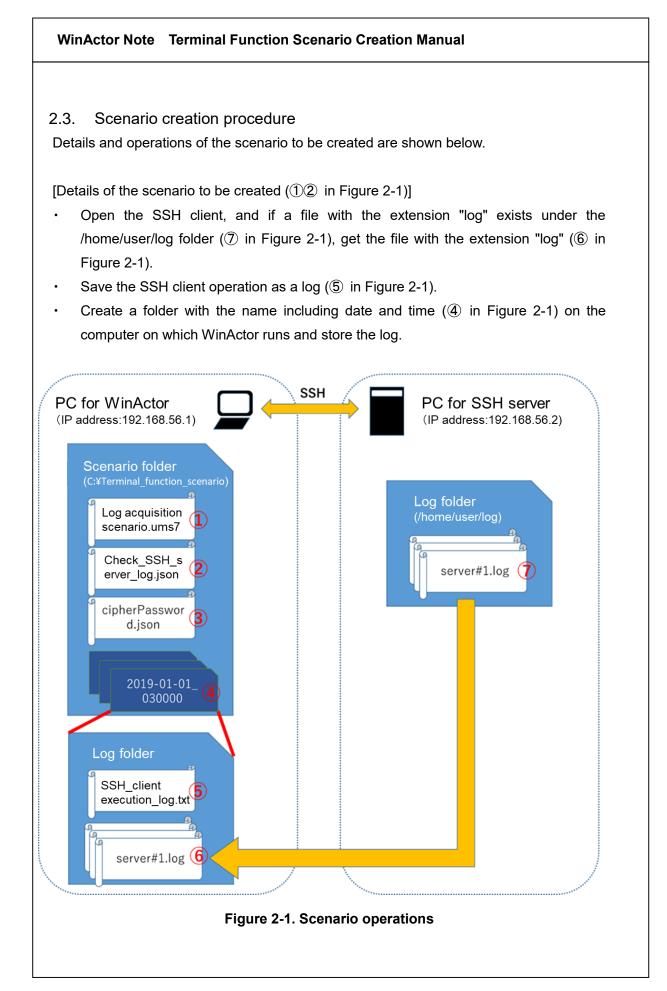
# Table 2-1. Environment used in this tutorial

#### 2.2. How to create a scenario

There are two ways to create a scenario using the terminal functions.

- 1 Create a scenario using the user libraries only
- 2 Use the macro functions of WinActor Note

As in the scenario you will create in Section 2.3, when processing operations continuously on WinActor Note, you can efficiently create the scenario by recording a macro on WinActor Note, checking the operations, and then running the created macro using the "Note\_ReadAndRunMacro" library.



No. inFile or folderDescriptionFigure2-1		Description
1	Log acquisition scenario.ums7	This is a WinActor scenario to be created in this tutorial. Here, the scenario with WinActor Ver.7 is used as an example.
2	Check_SSH_server_log.json	This is a WinActor Note macro to be created in this tutorial.
3	cipherPassword.json	This is a password file for logging in to the SSH server. <u>Create this in advance by referring to</u> <u>the section "Password file generation tool" in</u> <u>the material No.3 in Table 5-1.</u>
4	2019-01-01_030000, etc.	This is a folder for storing logs created by running the scenario of ①. This folder will be created on the computer on which WinActor runs. The date and time when the folder was created becomes the folder name. In the case of the left, it means 3:00:00 on January 1, 2019.
5	SSH_client_execution_log.txt	The contents displayed in WinActor Note when running the macro of ② are saved in this file. In the scenario created in this tutorial, the login message to the SSH server and the execution and result of the commands remain as information.
6	server#1.log, etc.	This is a file with the extension "log" acquired from the computer that runs the SSH server by running the scenario of ①.
7	server#1.log, etc.	This is a file with the extension "log" that exists on the computer that runs the SSH server. <u>The</u> <u>scenario to be created in this tutorial aims to</u> <u>get this file. If the file does not exist on the SSH</u> <u>server, create the file in advance.</u> Text data of less than 10 characters is used as the content of the file.

#### Table 2-2. Description of each file and folder

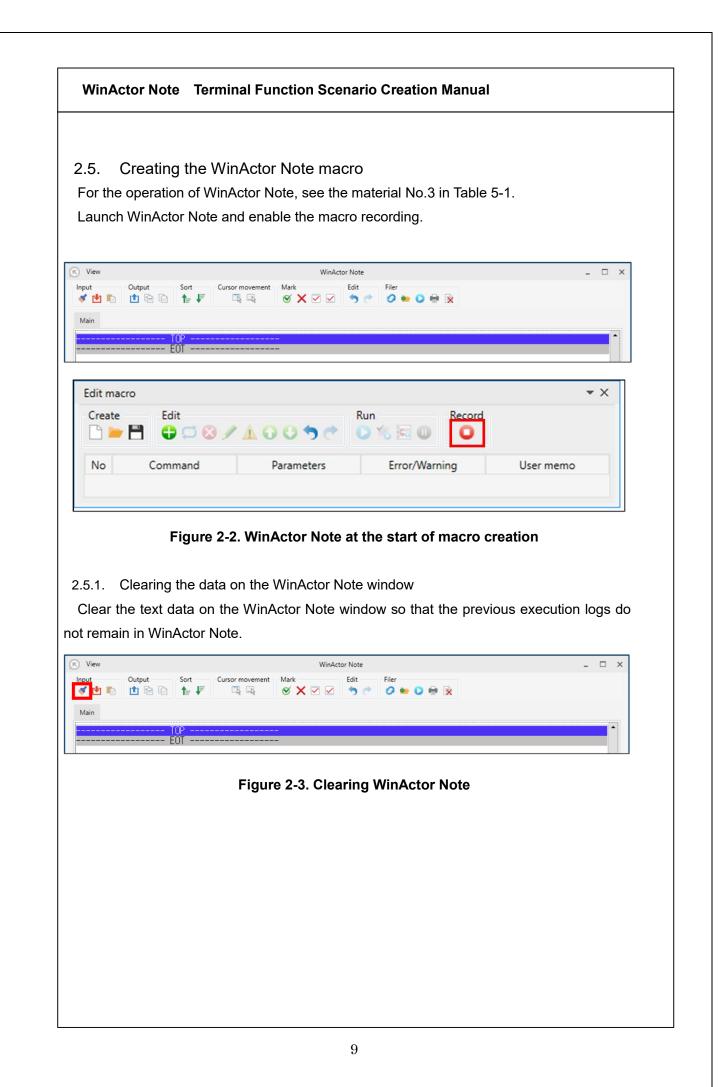
In the scenario to be created in this tutorial, you will use the nodes and user libraries of WinActor. For details of the nodes and user libraries in the scenario, see the materials No.1 and 2 in Table 5-1.

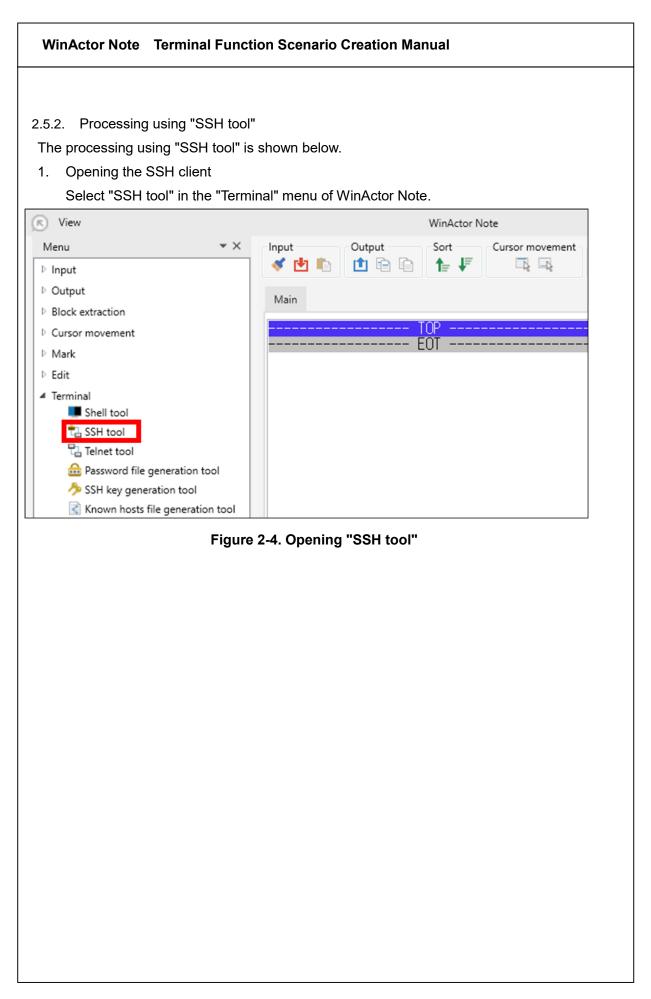
In addition, general Linux commands are used in the scenario to be created. For details on each command, see Linux-related documents and web sites.

#### 2.4. Scenario overview

The overview of how to create the scenario is as follows.

- 1. Creating the Creating macro (see Section 2.5)
  - A) Create the processing of getting a value that can determine whether the file with the extension "log" exists in the log folder on the SSH server and loading it into the WinActor variable. At that time, create the macro with WinActor Note launched and with the macro recorded.
- 2. Creating the Creating the WinActor scenario (see Section 2.6)
  - A) Add the processing of calling the created macro from the WinActor scenario.
  - B) Add the processing of judging whether the file with the extension "log" exists.
  - C) Add the processing of receiving the file with the extension "log" when the file with the extension "log" exists.
  - D) Add the processing of creating the folder to save logs and moving the log files.





Set the parameters in the area surrounded by the red frame ① in Figure 2-5, and click the "Open SSH session" button (the red frame ②). The input value of the password file is "C:\Terminal\_function\_scenario\cipherPassword.json."

-Connection s	ettings			
	haracter encoding	UTF-8		-
		LF (Linux, MacOS X)		-
	Host	192.168.56.2		
	Port	22		
	Username			
		_scenario\cipherPassword.json	Browse	
	Prompt string	ferrar and the second		
т	meout value [sec]			
	Known hosts file		Browse	
	rd authentication		browse	
		d the workFolder change into m	Browse	
SSH client		Receive file (SCP)		
		Open SSH session		2
Command		Close session Control code Ctrl+A 💌	Execute	
	Figur	e 2-5. Opening the SSH c	lient	

En	ecuting commands ter the commands written under [Command to be entered] in the following A) to D) to ① in Figure 2-6, and click the Execute button (② in Figure 2-6).
	Figure 2-6. Executing commands
A)	Move to the log folder
	[Command to be entered] cd log
B)	-
	[Command to be entered]
	date
C)	
	[Command to be entered] Is *.log
D)	-
	[Command to be entered] echo \$?
[No	ote]
CO	you execute the commands at an interval from "1. Opening the SSH client," the mmand execution may fail. In that case, click the "New" button in the "Create" menu
UI I	the "Edit macro" pane of WinActor Note and try again from Section 2.5.1.

Clic	
	k the "Close session" button in Figure 2-7 to close the SSH client.
	Close session 1
	Command Control code Ctrl+A   Execute
-	
	Figure 2-7. Closing the SSH client
n (7	) in Table 2-2 is server#1.log,server#2.log, the result of executing up to this point
be o	displayed in WinActor Note as shown in Figure 2-8.
[	S View WinActor Note
	Input Output Sort Cursor movement Mark Edit
	Main TOP
I	Last login: Thu Aug 29 14:07:12 2019 from 192.168.56.1
	user@wipactor:~\$ cd log
	user@winactor:~\$ cd log user@winactor:~/log\$ date
	user@winactor:~/log\$ date
	user@winactor:~/log\$ date Thu Aug 29 2019 14:35:54 JST
	user@winactor:~/log\$ date Thu Aug 29 2019 14:35:54 JST user@winactor:~/log\$ Is *.log
	user@winactor:~/log\$ date Thu Aug 29 2019 14:35:54 JST user@winactor:~/log\$ ls *.log server#1.log server#2.log
	user@winactor:~/log\$ date Thu Aug 29 2019 14:35:54 JST user@winactor:~/log\$ Is *.log
	user@winactor:~/log\$ date Thu Aug 29 2019 14:35:54 JST user@winactor:~/log\$ Is *.log server#1.log server#2.log user@winactor:~/log\$ echo \$?

#### 2.5.3. Text processing

Follow the steps below to select the result of the existence of files with the extension "log."

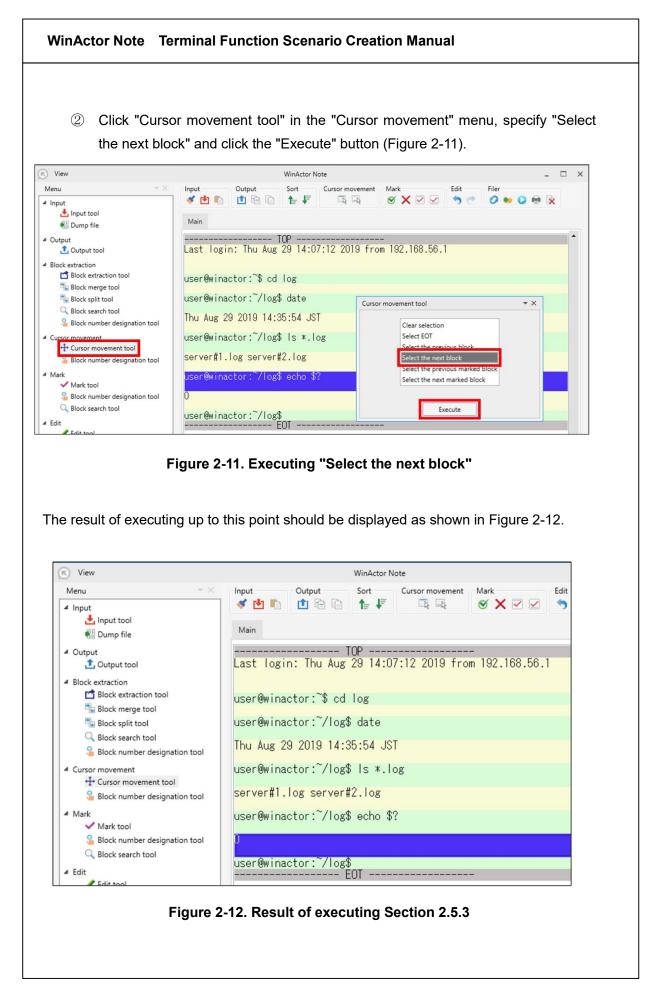
 Removing unnecessary spaces by using "White space removal tool" in the "Edit" menu Select "All blocks" for the target and "Remove line breaks" for the operation (① in Figure 2-9), and click the "Execute" button (② in Figure 2-9).

White space removal	tool	<b>▼</b> ×
Target	All blocks	
	Remove white spaces	
	Remove line breaks	1
	Remove the leading white spaces	
	Remove the trailing white spaces	
	Remove the leading and trailing white spaces	
	for the second sec	
	Execute	

Figure 2-9. Executing "White space removal tool"

- 2. Selecting the result of executing "echo \$?" by using "Block search tool" and "Select the next block"
  - Click "Block search tool" in the "Block extraction" menu, enter "\$?" for the keyword, select "Contain" in the drop-down list and "Forward search (cursor moves down)" for the operation (① in Figure 2-10), and click the "Execute" button (② in Figure 2-10).

Keyword	S? Contain	
i	Forward search (cursor moves down)	
	Backward search (cursor moves up) Mark the matched blocks (unmark others)	
	Mark the matched blocks	
	Jnmark the matched blocks	
	Delete the matched blocks	
	Delete the unmatched blocks	
	Mark up to the next matched block	
ļ		
	Execute	



2.5.4. Saving the text data on the WinActor Note window to a file

Save the text data on the WinActor Note window as a log.

Click "Output tool" in the "Output" menu, set the values according to Table 2-3 and click the "Execute" button.

Output tool		▼ ×
Target	All blocks	
	to File 🛛 🗸 🗸	
3	nction_scenario\SSH_client_execution_log.txt Browse	
Character encoding	MS932 (Shift JIS)	
Line-break code	CR+LF (Windows)	
6	Do not record the workFolder change into macro.	
0	Ensure line-break code at the end.	
	Execute	

Figure 2-13. Settings in the "Output tool" window

No. in Figure	Setting value
2-13	
1	Select "All blocks"
2	Select "to File"
3	Enter "C:\Terminal_function_scenario\SSH_client_execution_log.txt"
4	Select "MS932(Shift JIS)"
5	Select "CR+LF(Windows)"
6	Check the box
$\bigcirc$	Check the box

# Table 2-3. "Output tool" settings

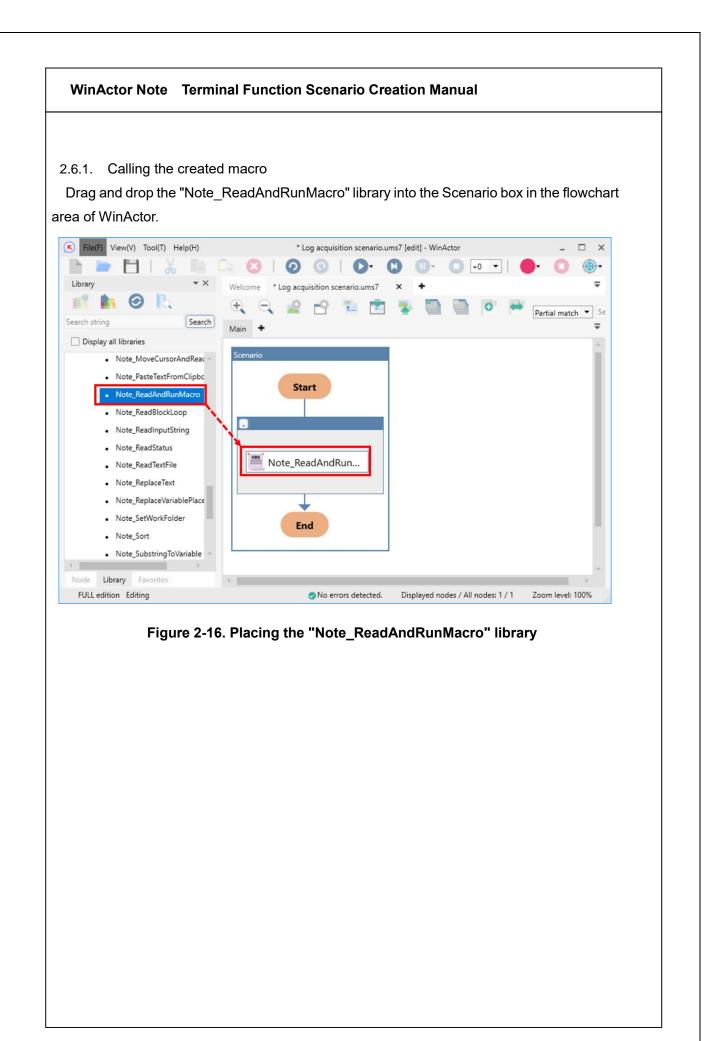
#### 2.5.5. Saving the macro

In the "Edit macro" pane of WinActor Note, (if the pane is not displayed, open it from the "View" menu), disable the macro recording (① in Figure 2-14), and click the "Save" button (② in Figure 2-14).

Save the macro currently being edited (④ in Figure 2-14) to the folder "C:\Terminal\_function\_scenario" with the filename "Check\_SSH\_server\_log.json" (③ in Figure 2-14).

		0050 B0	Record 1	
>	Command	Parameters	Error/Warning	User memo
	trim	[-target, ALL, CR]		
	jump	[top]		
	search	[-condition, CONTAINS, -action,		
	next			
	workFolder		Delete the "workFolder" comma	
	save	[-appendCr, -crCode, CRLF, -tar		
	Organize  New for Desktop Documents Downloads Music Pictures Videos Local Disk (C:)	<ul> <li>Name</li> <li>No its</li> <li>&lt;</li> <li>eck_SSH_server_log.json</li> </ul>	Date modified ems match your search.	I_function_sc
	<ul> <li>Hide Folders</li> </ul>		Save 4	Cancel
		Figure 2-14. Sa	ving the macro	

<ol> <li>Creating the Win ave the WinActor scenar</li> </ol>		time as a ne	w scenario file.	
	File(F)     View(Normalized       Lit     Open       Sea     Save       Save     Save       Import	/) Tool(T) I		
Save		, 		×
← → ~ ↑ 📙 « Loca	> Terminal_fun	م ا	Search Terminal_function	1_SC
Organize 👻 New folder			8== ▼	•
Desktop Documents Downloads Music Pictures		tems match your :	Date modified	Туре
File name: Log acc	uisition scenario			~
Save as type: Scenari	o files (*.ums7)			~
∧ Hide Folders			Save Cancel	:
Fig	ure 2-15. Saving the	e new scena	ario file	

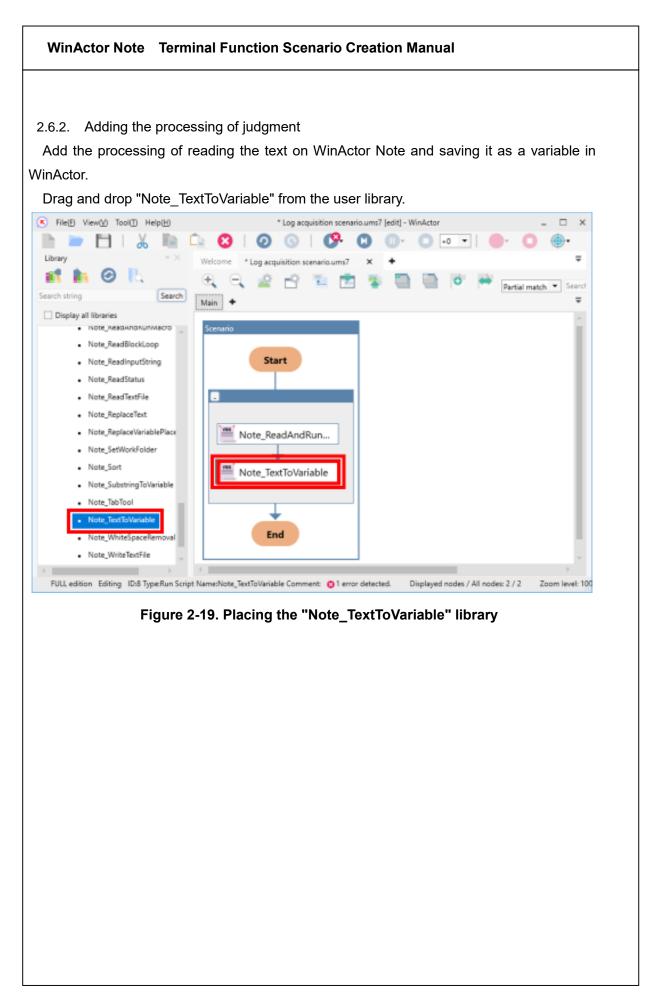


Name Note_ReadAndRunMacro   Comment   Strings Script Annotation Version Read amacro into WinActor Note and execute it. * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> CATerminal_function_scenario\Check_SSH_ • Update Restore Restore	Name       Note_ReadAndRunMacro         Comment	Name       Note_ReadAndRunMacro         Comment	Name       Note_ReadAndRunMacro         mment	Name       Note_ReadAndRunMacro         Comment	Comment		un Script		
Comment         Settings       Script         Annotation       Version         Read a macro into WinActor Note and execute it.         * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.         Macro_filename       Value=> C:\Terminal_function_scenario\Check_SSH_          Update       Restore	Comment          Settings       Script       Annotation       Version         Read a macro into WinActor Note and execute it.       *         * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.	Comment         Settings       Script       Annotation       Version         Read a macro into WinActor Note and execute it.         * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.         Macro_filename       Value=> C:\Terminal_function_scenario\Check_SSH_          Update       Restore	Ings Script   Annotation Version   a macro into WinActor Note and execute it. ro_filename is relative to the working folder or the folder where the ent scenario exists when the working folder is not set.   Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_ <   Update   Restore	Comment Settings Script Annotation Version ead a macro into WinActor Note and execute it. Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_	Comment	ReadAndRunMacro			
Settings       Script       Annotation       Version         Read a macro into WinActor Note and execute it.       *         * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.       Macro_filename       Value=> C:\Terminal_function_scenario\Check_SSH_           Update       Restore	Settings       Script       Annotation       Version         Read a macro into WinActor Note and execute it.       *         * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.       ////////////////////////////////////	Settings       Script       Annotation       Version         Read a macro into WinActor Note and execute it.       *       *       Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.         Macro_filename       Value=>       C:\Terminal_function_scenario\Check_SSH_           Update       Restore	ngs Script Annotation Version a macro into WinActor Note and execute it. ro_filename is relative to the working folder or the folder where the ent scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_ < Update Restore	Settings Script Annotation Version ead a macro into WinActor Note and execute it. Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_ < Update Restore					
Read a macro into WinActor Note and execute it. * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore	Read a macro into WinActor Note and execute it. * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.  Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore	Read a macro into WinActor Note and execute it.  * Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.  Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore	a macro into WinActor Note and execute it. ro_filename is relative to the working folder or the folder where the ent scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_ < Update Restore	ead a macro into WinActor Note and execute it. Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_ Update Restore	Settings Script				
* Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_ < Update Restore	* Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_ Update Restore	* Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_   Update Restore	ro_filename is relative to the working folder or the folder where the ent scenario exists when the working folder is not set. Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_   Update Restore	Macro_filename is relative to the working folder or the folder where the current scenario exists when the working folder is not set.          Macro_filename       Value=> C:\Terminal_function_scenario\Check_SSH_          Update       Restore					
current scenario exists when the working folder is not set.  Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_   Update Restore	current scenario exists when the working folder is not set.  Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_   Update Restore	current scenario exists when the working folder is not set.  Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_   Update Restore	ent scenario exists when the working folder is not set.  Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_   Update Restore	current scenario exists when the working folder is not set.  Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_   Update Restore					
Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore	Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore	Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore	Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore	Macro_filename Value=> C:\Terminal_function_scenario\Check_SSH_  Update Restore				ere the	
Update	Update	Update	Update	Update					<b></b> _
					Macro_filenar	me Value=> C:\Termina	al_function_scenario	\Check_SSH_ ▼	
Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename	U	pdate		Restore	
Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename	Figure 2-17. Specifying the macro filename					
						Figure 2-17. Speci	fying the mac	ro filename	

Check whether it works properly with the "Note\_ReadAndRunMacro" library alone. Click the "Run scenario" button on the toolbar of WinActor and check that the expected text contents are output on WinActor Note. After checking the operation, delete the "C:\Terminal\_function\_scenario\SSH\_client\_execution\_log.txt" file.

cenario.ums7 [edit] - WinActor	Imput       Output       Sort       Cursor movement       Mark       Edit         Imput       Imput       Imput       Imput       Imput       Imput         Imput       Imput       Imput       Imput       Imput       Imput       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Imput       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Imput       Edit         Imput
cenario.ums7 [edit] - WinActor	Input       Output       Sort       Cursor movement       Mark       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Imput       Imput       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Imput       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Edit       Imput       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Edit       Imput       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Edit       Imput       Edit         Imput       Imput       Imput       Imput       Imput       Imput       Imput       Edit       Imput       Imput

Figure 2-18. Checking the operation of the "Note\_ExecuteMacroLoading" library



After placing the library, double-click it to open the Property window.

To select the result of executing "echo \$?" after running the "Note\_ReadAndRunMacro" library and import the content of the selected result as a variable on WinActor, define the variable name as "echo\_execution\_result" and click the "Update" button.

A window for confirming whether to register the entered name as a new variable will appear. Click "Yes," and it will be registered as a new variable in the Variable list pane of WinActor.

Property			<b>▼</b> ×
		Run Script	
Nan	ne echo_exe	cution_result	
Comme	nt		
0.00		and the second second	
Settings		notation Version	
Get text fro	om WinActor	Note and store it to a variable.	
	Targe	Selected_block	
Lir	ne-break_code	Not_appended 💌	
Line-bre	eak_code_type	CR+LF (Windows)	
Va	riable_for_tex	echo_execution_result	
	Update	Restore	

Figure 2-20. Entering the variable name

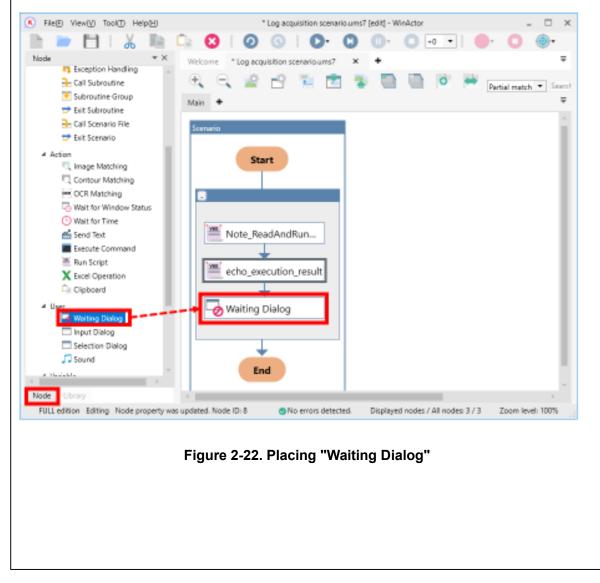
riable	list					<b>▼</b> ×
(xy)	🔍 💀 🗞	$\mathbf{O}$ $\mathbf{O}$ $\mathbf{O}$		8 🖌 '	h 😨 🗄	► <b>Ç</b>
		N	Current value	No initializat	Initial value	Mask
	Group name	Variable name	Current value	NO minializat	initial value	IVIdSK

#### Figure 2-21. Confirming the update of the Variable list pane

Check the operation that the string resulted from executing "echo \$?" will be imported into the variable "echo\_execution\_result."

If the "Run scenario" button is clicked in the current scenario state, the variable value will be initialized after all the processing is completed. To prevent this from happening, place "Waiting Dialog" at the end of the scenario.

Drag and drop "Waiting Dialog" from the Node tab.



Run the current scenario to check the operation.

Click the "Run scenario" button on the toolbar of WinActor, and check whether the current value for the "echo\_execution\_result" in the Variable list pane is updated with the text resulted from executing "echo \$?" ("0" in this case) entered.

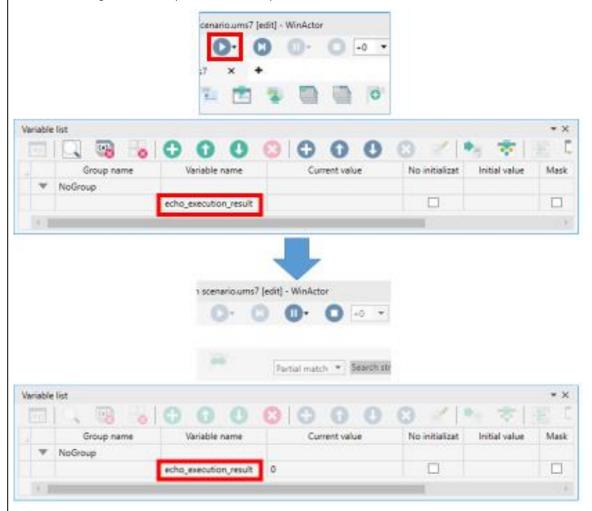
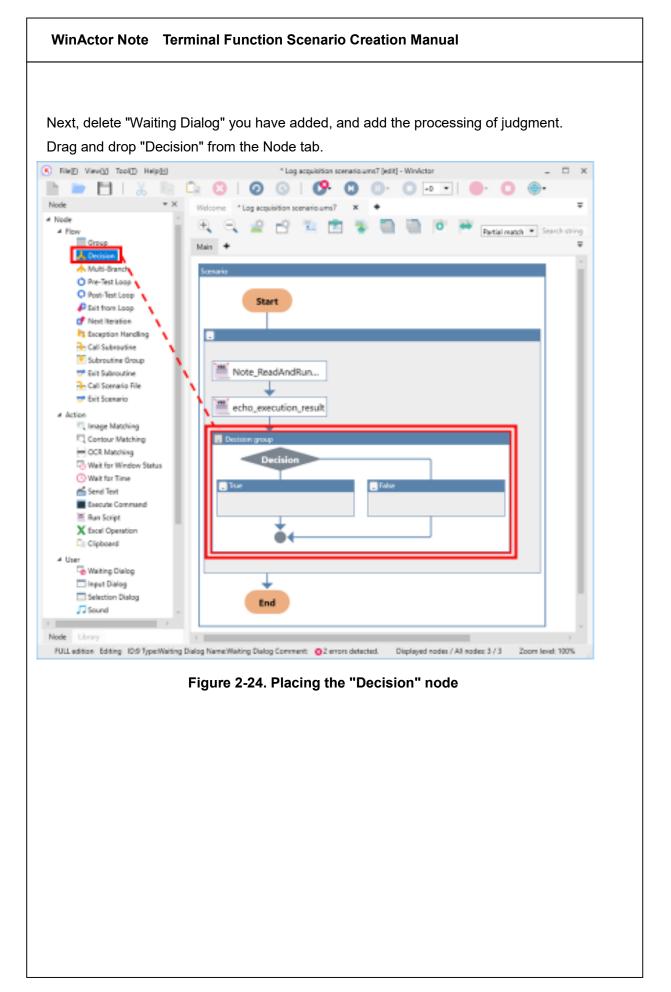


Figure 2-23. Checking the operations of importing the value into the variable



After placing the node, double-click it to open the Property window.

After changing the name and the branch name of the "Decision" property window, click "Edit," and in the "Conditional expression" property window, select "echo\_execution\_result" for the left side of the conditional expression and enter 0 on the right side of the conditional expression. As a result of these settings, if the result of executing "echo \$?" is 0, that means if the file with the extension "log" exists, it will move to the operation with the branch named "true."

Lastly, click the "Update" button on the property windows of "Conditional expression" and "Decision" to complete the property settings.

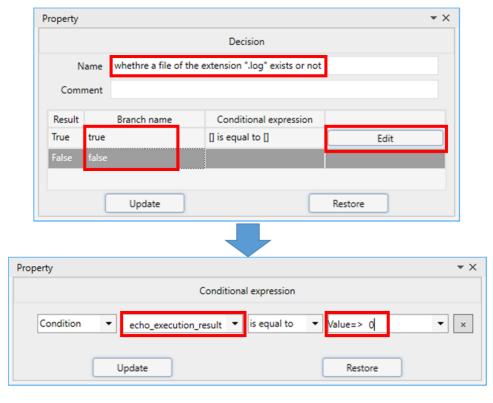
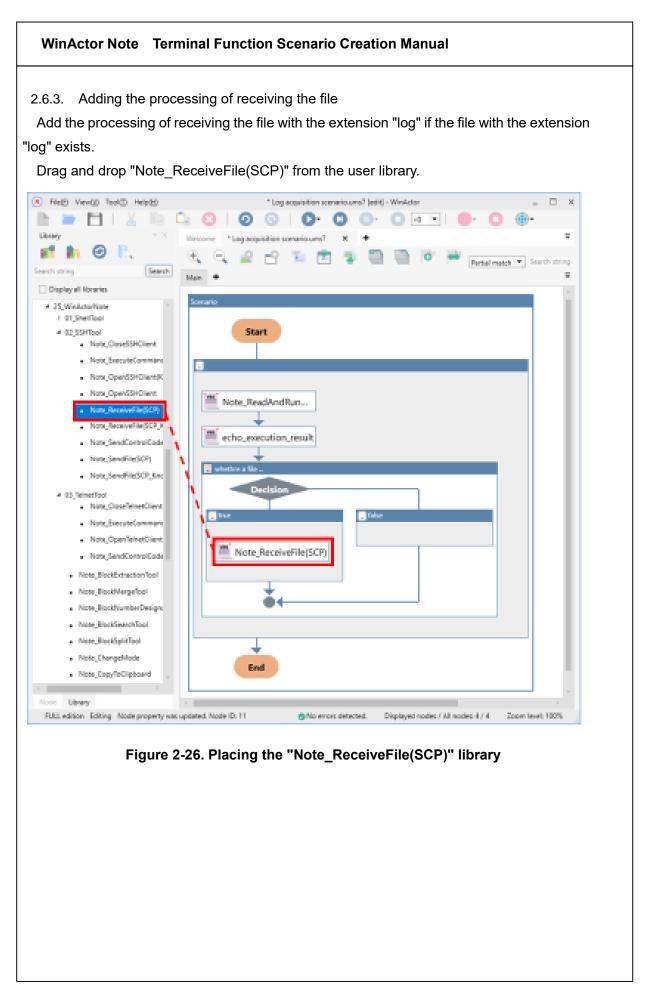


Figure 2-25. Conditional expression settings



Double-click the placed library to open the Property window.

For each item, enter the value according to Table 2-4. After entering the values, click the "Update" button. A window for confirming whether to register these values as new variables will appear. Click "Yes," and the values will be registered as new variables in the Variable list pane of WinActor.

Item	Setting value	Remarks
Authentication_method	Select "Password_authentication"	-
Host	Enter "Host"	*1
Port	Enter "Port_number"	*1
User_name	Enter "Username"	*1
Password_file	Enter "Password_file"	*1
Private_key_file		Leave "Value=>" as it is
Timeout_value[sec]	Enter "Timeout_value"	*1
Source_file	Enter "Source_file"	*1
Destination_path	Enter "Date"	*1

#### Table 2-4. "Note\_ReceiveFile(SCP)" library property settings

\*1 Select "\*" once and enter the value.

operty						*	×
			Run Script				
Name Note_ReceiveFile(SCP)							
Comment							
	nere l'anne						
Settings So Receive a file b		otation Versi	on				
	here the cu	rrent scenario e				•	
	Host	Host				<b>-</b>	
	Port	Port_numbe	r			•	
l.	User_name	Username				•	
Pas	ssword_file	Password_file				•	
Private_key_file Value=>							
Timeout	_value[sec]	Timeout_val	ue			•	
:	Source_file	Source_file				•	
Destin	ation_path	Date				•	
ſ	Update			Re	estore		
	F	igure 2-27.	After settin	g propert	ies		

Enter the initial value for each variable in the Variable list pane according to Table 2-5.

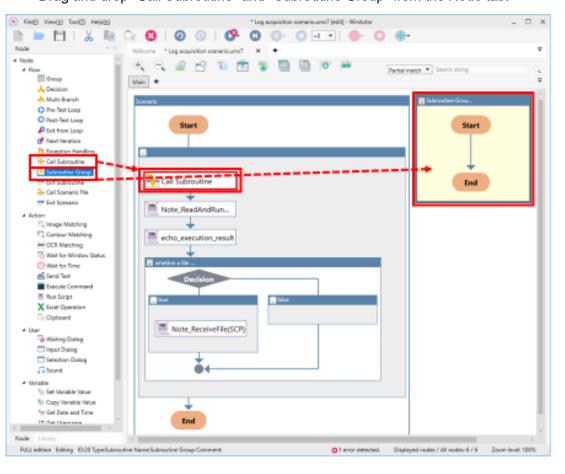
Table 2-5. Initial value settings in the Variable list pane

Variable name	Initial value	Remarks
Host	192.168.56.2	-
Port_number	22	-
Username	user	-
Password_file	cipherPassword.json	-
Timeout_value	10	Adjust with an appropriate value
		according to your environment.
Source_file	/home/user/log/*.log	-
Date		Leave the initial value blank. The value
		will be automatically generated by the
		processing of Section 2.6.4.

(xy)	🔍 😨 😽	$\bigcirc \bigcirc $	3 0 0 0	8	1 🍬 😨 🗄	<b> </b> →
	Group name	Variable name	Current value	No initializat	Initial value	Mas
$\mathbf{v}$	NoGroup					
		echo_execution_result				
		Host			192.168.56.2	
		Port_number			22	
		Username			user	
		Password_file			cipherPassword.json	
		Timeout_value			10	
		Source_file			/home/user/log/".log	
		Date				

Figure 2-28. After setting the initial values

- 2.6.4. Adding the processing of creating the folder for saving logs
- Creating a subroutine and the processing of calling the subroutine Add the processing of creating the folder according to the date and time for storing the logs on the SSH server. The log folder creation is processed in a subroutine. Drag and drop "Call Subroutine" and "Subroutine Group" from the Node tab.



# Figure 2-29. Placing the nodes ("Call Subroutine" and "Subroutine Group")

After placing these nodes, open the Property window of each node.

For the "Subroutine Group" node property window, change the name to "Create log folder." For the "Call Subroutine" node property window, change the name to "Call Create log folder" and select "Create log folder" for the subroutine name.

Property 💌 👻	Property 👻 🗙
Subroutine	Call Subroutine
Name Create log folder	Name Call Create log folder
Comment	Comment
Local variable list (variables to restore the initial value at the end)	Subroutine name Create log folder    Synchronize
Variable Select variable name Add Delete	Return value Select variable name (optional)
Allow empty string	
Update Restore	Update Restore

# Figure 2-30. "Subroutine Group" and "Call Subroutine" property settings

- 2 Creating the processing in the subroutine
   Create the processing in the "Create log folder" subroutine.
   The overall steps are as follows.
  - 1. Creating the folder name
    - 1 Getting the date and time.

The format of the date and time to be acquired is as follows.

<Date and time format> yyyy-mm-dd HH:mm:ss

However, it may not be in the above format depending on the regional settings of Windows. In that case, refer to this manual and change the processing contents of the scenario.

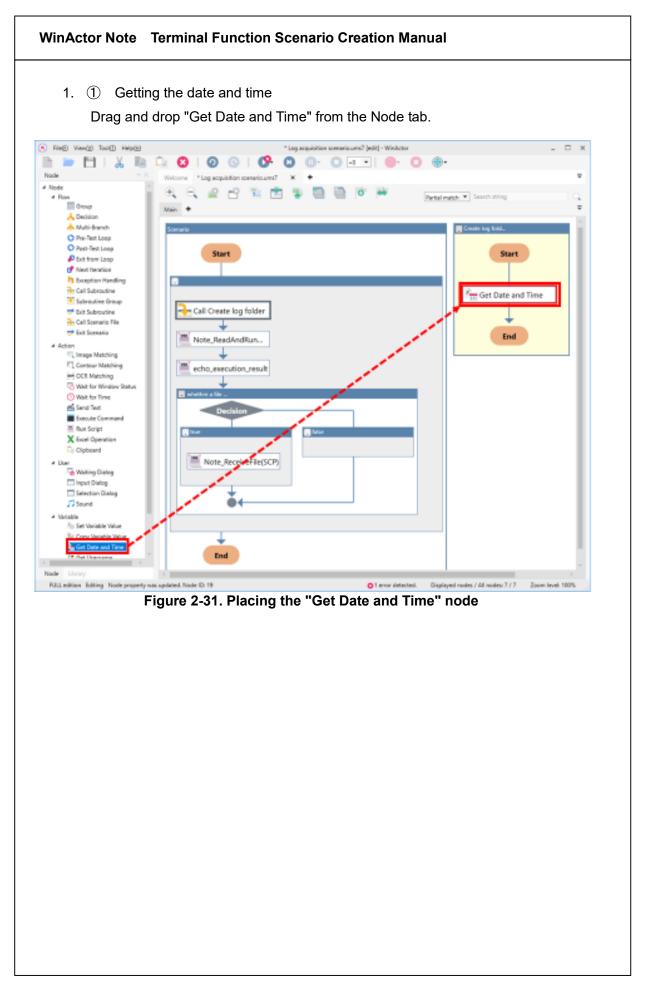
② Generating the folder name.

Convert the format of ① into the format "yyyy-mm-dd\_HHmmss" of ④ in Table 2-2 by removing ":" and replace the space with an underscore.

#### 2. Creating the folder

Create the folder with the folder name generated in the step ② of 1 above.

The details of each processing are described below.



After placi	ina the n	ode double	e-click it to one	n the Property v	vindow
					ne date format, "O
			lick the "Upda		
Property					<del>~</del> ×
		Get	t Date and Time		
Name	Get Date	and Time			
Comment					
Outpu	ut variable	Date			<b>~</b>
Fo	ormat type	Date and time	2		-
Da	ate format	yyyy-mm-dd 🗸			-
т	ïme zone	OS default			<b>•</b>
				Partara	
	Updat	ie		Restore	
Figu			and Time" no	de property se	ttings
Figu			and Time" no		ttings
Figu			and Time" no		ttings
Figu			and Time" no		ttings
Figu			and Time" no		ttings
Figu			and Time" no		ttings
Figu			and Time" no		ttings
Figu			and Time" no		ttings

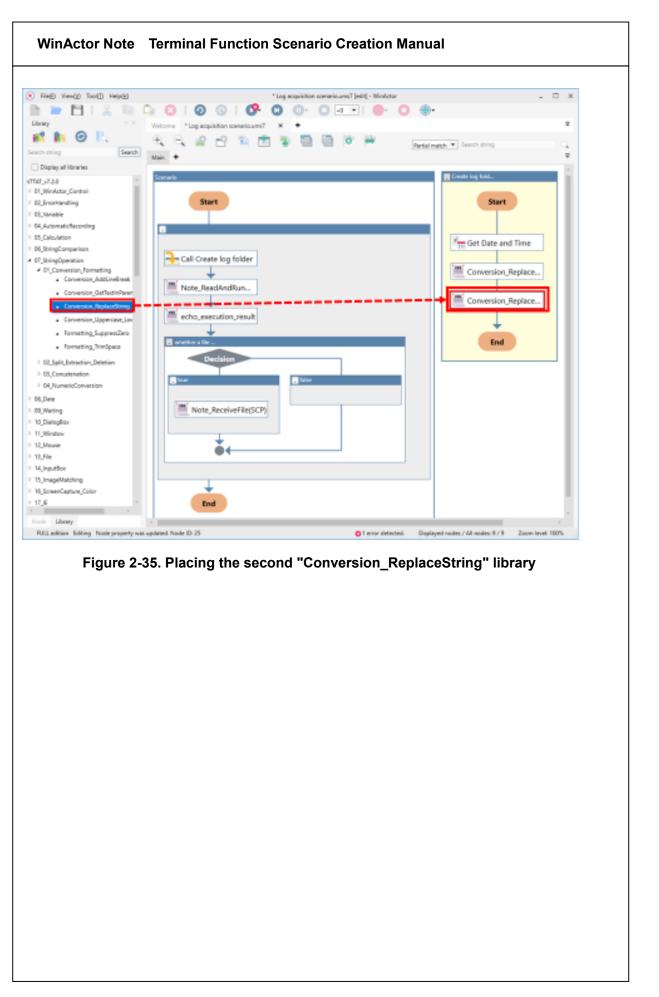
1. ② Generating the folder name

First, add two "Conversion\_ReplaceString" libraries in a row. For Converted\_string, select "Date" for both libraries. For Before\_replacement and After\_replacement, select ":" and blank for the first "Conversion\_ReplaceString" library, and a single space and "\_" for the second "Conversion\_ReplaceString" library.

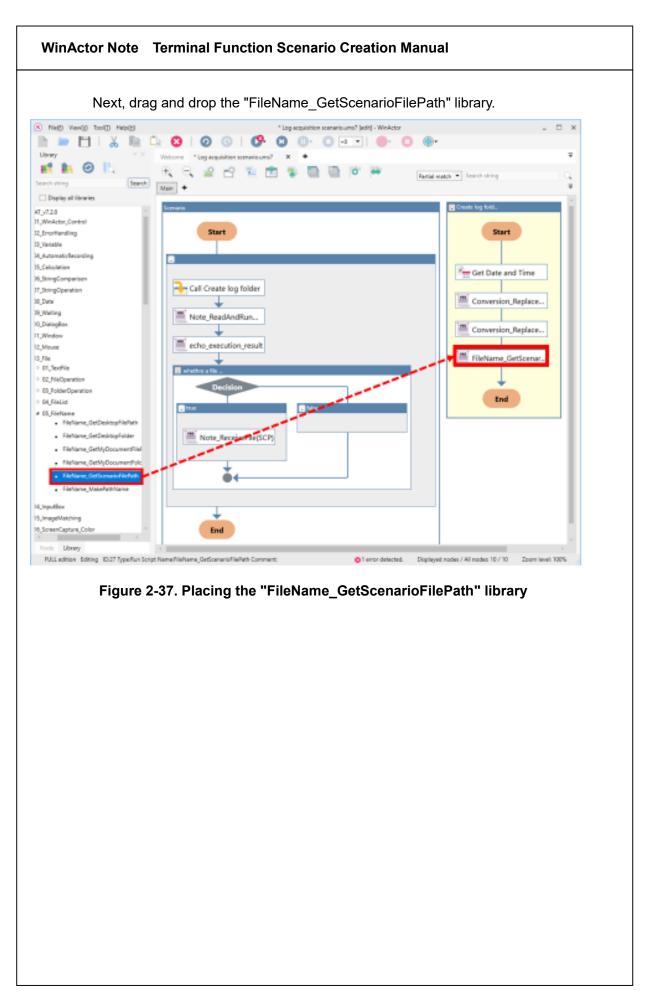
Library Librar	Image: Image
Linny Witcom * Log acquisition scenario.um? * Ment coing Search Display all linearies Tut you have the coing of the coi	Constant long total.
Conversion, Jornarting     Note, ReadAndRun     Conversion, State Conversion     Bi, Dree     Dy, Market     Conversion, Conversion     Display Leftware     Conversion, Conversion     Conversion, Conversio	Constant long total.
Auch ming Search Wain + Display all libraries Tall , A 20 (Wain Question Control Of Libraries Tall , A 20 (Wain Question Control Of Libraries Conversion formarting Californiania Conversion formarting Californianiania Conversion formarting Californianianiania Conversion formarting Californianianianianianianianianianianianianian	Constant long total.
Vain	Start Fm Get Date and Time Conversion_Replace
Tut y 3,3,8 Starting Stylestarting Starting Starting Starting Starting Starting Call Create log folder Note_ReadAndBun. Call Create log folder Note_ReceiveFile(SCP) Note_ReceiveFile(SCP) Starting Note_ReceiveFile(SCP) Julianaw Starte	Start Fm Get Date and Time Conversion_Replace
DI JWoldzar, Cantrol B2, Frankfanding B3, Skiniskie D4, Automatoficeording B3, SkinigCompanion B5, SkinigC	Start Fm Get Date and Time Conversion_Replace
B2_Encertanating B3_Encertanating B4_Encertanation B5_EnceContent B5_EnceContention B5_EnceContention B5_EnceContention B5_EnceContention B5_EnceContention B5_EnceContention B5_EnceContention B5_Ence B5_EnceContention B5_EnceCon	Conversion_Replace
<ul> <li>B), Variable</li> <li>B), Sing Comparison</li> <li>B), Sing Comparison</li> <li>B), Sing Comparison</li> <li>B), Sing Comparison</li> <li>B), Sing Comparison, Tomarzing</li> <li>Conversion, Udditionalisesk</li> <li>Conversion, Udditionalisesk</li> <li>Conversion, Uppervase, Law</li> <li>Romanzing, Suppress/Earo</li> <li>Romanzing, Supp</li></ul>	Conversion_Replace
B4, AutomaticResording B5, Calculation B6, StringComparison B7, DeinsgComparison P Of Conversion, Formatting Call Create log folder Call Create	Conversion_Replace
B5_Calculation B6_StringComparison B7_DisingDparsion B7_DisingDparsion Conversion_Host Host Conversion_Host Host Conversion_Host Host B7_DisingDparsion_Determin Conversion_Host Host Conversion_Host Host B7_DisingDparsion_Determin B7_DisingDisc B7_Disc	Conversion_Replace
B. ShingComparison EF_SkingOperation # Of Conversion_Formating • Conversion_Vidit LineBask • Operating_VinitGask • Opera	Conversion_Replace
20. StringOperation	Conversion_Replace
OU_Conversion_Formatting     Conversion_SetEstin@lask     Conversion_GetEstin@lask     Conversion_	
Conversion_distinglinest     Conversion_GefExctinPare     Conversion_GefExctinPare     Conversion_UpgenessExe     Conversion_UpgenessExe     Conversion_UpgenessExe     Conversion_Deterion     OU_Spirl_Entrestion     OU_Constitution_Deterion     OU_Constitution_Deterion_Deterion     OU_Constit     OU_Constitution_Deterion     OU_	
Convention_GefSectinGuer      Conversion_MapleceStrep      Conversion_MapleceStrep      Conversion_MapleceStrep      Conversion_Version      Col_Spir_Ensection_Detertion      Col_Spir_Ensection_Detertion_Detertion_Detertion      Col_Spir_Ensection_Deterti	End
Conversion_Uppervase_tav     Conversion_Uppervase_tav     Formatting_SuppressEuro     Pormatting_TrinSpace     OO_Spirit_Breastion_Deletion     OO_Spirit_Breastion_Delet	
Convestion_Uppervase_law     Formatting_SuppressEuro     Formatting_TrinSpace     CO_SpirLEnsexten_Deletion     CO_SpirLE	End
Formatting_SuppressDare     Formatting_NimSpace     O0_Spit_Entrestion_Deletion     Od_Spit_Entrestion     Od_Spite	
Formatting_NimSpace     OL_Split_Entextion_Deletion     OL_Split_Entextion_Deletion     OL_Split_Entextion     OL_VisionG     O_Waring     O_UNaring	
Tomating_impace     Outpic Entertion     Outpi	
O Concelenation     O Concelenatio     O Concelenation     O Concelenation     O Concelenation     O	Calue
OS,Constantion     OL/NumericConvenian     OL/NumericConvenian     OL/NumericConvenian     OL/NumericConvenian     OL/NumericConvenian     Note_ReceiveFile(SCP)     Note_ReceiveFile(SCP)     OL/NumericConvenian	Calue
OL/NumericConversion	Calue
08,Dvie 00,Weing 10,Duiog8ax 11,Windaw 12,Mouve	
00,Weing 10,Dalog8xx 11,Window 12,Meave	
10,Dalog8xx 11,Window 12,Mexee	
11,Window 12,Mexee	
12 Mouse	
14, InputSox	
15 ImageMatching	
16_SomerrCapture_Color	
17,6 T End	
Node Library *	
RJILledition Bolting ID-25 Type Run Script Name Conversion, ReplaceString Comment:	O1 error detected. Displayed nodes / All nodes: 8 / 8 Zoom level 100
Figure 2-33. Placing the first '	

roperty	· · · · · · · · · · · · · · · · · · ·	×
	Run Script	
Name	Conversion_ReplaceString	
Comment		
Settings S	Script Annotation Version	
replaces it witl < <caution>&gt;</caution>		
replaces it wit < <caution>&gt; - If the string l - When deletin deleted before</caution>	th "After_replacement" string. before replacement cannot be found, it remains as the original string. ing a specific string from the conversion string, specifies the character to be re substitution as an empty after replacement.	
replaces it wit < <caution>&gt; - If the string l - When deletin deleted before</caution>	th "After_replacement" string. before replacement cannot be found, it remains as the original string. ing a specific string from the conversion string, specifies the character to be	
replaces it witl < <caution>&gt; - If the string I - When deletin deleted before Conve</caution>	th "After_replacement" string. before replacement cannot be found, it remains as the original string. ing a specific string from the conversion string, specifies the character to be re substitution as an empty after replacement.	

# Figure 2-34. The first "Conversion\_ReplaceString" library property settings (":" and blank)



Run Script         Name       Conversion_ReplaceString         Comment       Settings       Script       Annotation       Version         searches for "Before_replacement" string.       Settings       Script       Annotation       Version         (Caution>>       If the string before replacement as the original string.       Settings       Setting Top Setting from the conversion string, specifies the character to be beleted before substitution as an empty after replacement.         (Converted_string)       Date       Image: Converted_string)       Image: Converted_string)	roperty				<del>~</del> ×
Comment         Settings       Script       Annotation       Version         Gearches for "Before_replacement" string from the string specified in "Converted_string" and eplaces it with "After_replacement" string.         < <caution>&gt;         If the string before replacement cannot be found, it remains as the original string.         When deleting a specific string from the conversion string, specifies the character to be beleted before substitution as an empty after replacement.         Converted_string       Date         Before_replacement       Value=&gt;         Update       Restore</caution>			Run Script		
Settings       Script       Annotation       Version         Searches for "Before_replacement" string from the string specified in "Converted_string" and eplaces it with "After_replacement" string.         < <caution>&gt;         If the string before replacement cannot be found, it remains as the original string.         When deleting a specific string from the conversion string, specifies the character to be beleted before substitution as an empty after replacement.         Converted_string       Date         Before_replacement       Value=         Update       Restore</caution>	Name	Conversion_ReplaceSt	tring		
Searches for "Before_replacement" string from the string specified in "Converted_string" and eplaces it with "After_replacement" string. Searches for "Before replacement" string. If the string before replacement cannot be found, it remains as the original string. When deleting a specific string from the conversion string, specifies the character to be beleted before substitution as an empty after replacement. Converted_string Date Before_replacement Value=> Update Restore re 2-36. The second "Conversion_ReplaceString" library property settings	Comment				
Searches for "Before_replacement" string from the string specified in "Converted_string" and eplaces it with "After_replacement" string. Searches for "Before replacement" string. If the string before replacement cannot be found, it remains as the original string. When deleting a specific string from the conversion string, specifies the character to be beleted before substitution as an empty after replacement. Converted_string Date Before_replacement Value=> Update Restore re 2-36. The second "Conversion_ReplaceString" library property settings	Catting C	tet Assession X	(aution)		
re 2-36. The second "Conversion_ReplaceString" library property settings	eplaces it wit < <caution>&gt; If the string When deleti leleted befor Conve Before_re</caution>	n "After_replacement" s before replacement can bg a specific string from substitution as an emp rted_string Date placement Value=>	string. nnot be found, it remains n the conversion string, s	as the original string	g.
re 2-36. The second "Conversion_ReplaceString" library property settings	1	Undate		Restore	
		opone			



Double-click the placed library to open the Property window.

Select "Date" for the filename, enter "Result\_folder" for the file path generation result, and click the "Update" button.

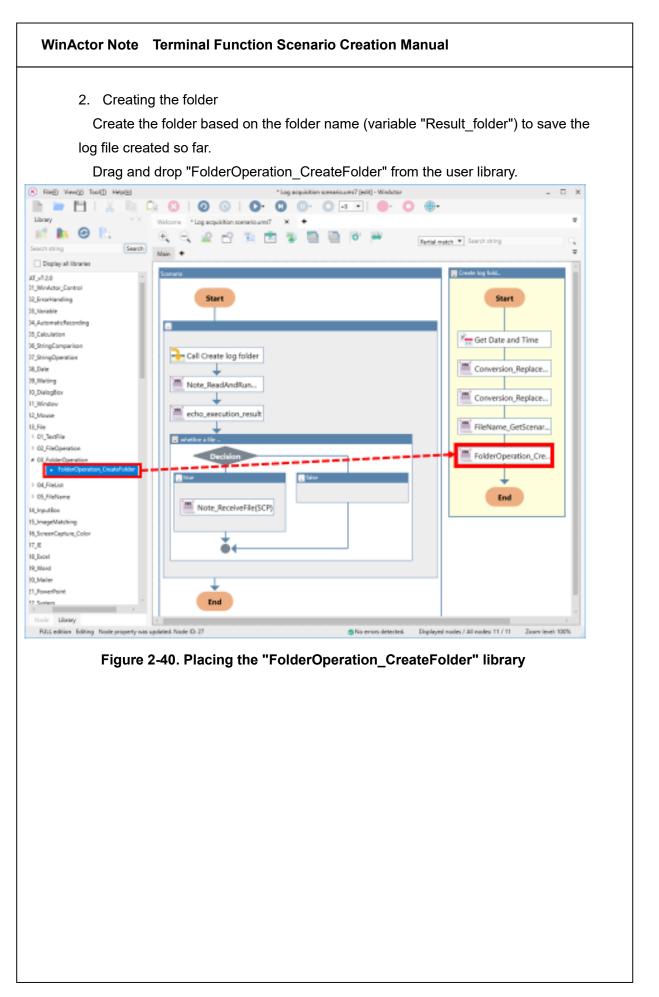
A window for confirming whether to register the entered value as a new variable will appear. Click "Yes," and it will be registered as a new variable in the Variable list pane of WinActor.

				Run Scrip	ot		
Nar	ne Fil	eName_	GetScen	arioFilePat	h		
Comme	ent						
Settings	Script	Anno	otation	Version			
						 ified as the	file
	emp\a.t	kt is stor	ed as file	:\temp fold e path gen		 ified as the	file
name, c:\te	emp\a.t	kt is stor	ed as file Date			 ified as the	e file
name, c:\te	emp\a.t File	kt is stor	ed as file Date	e path gen		 ified as the	e file

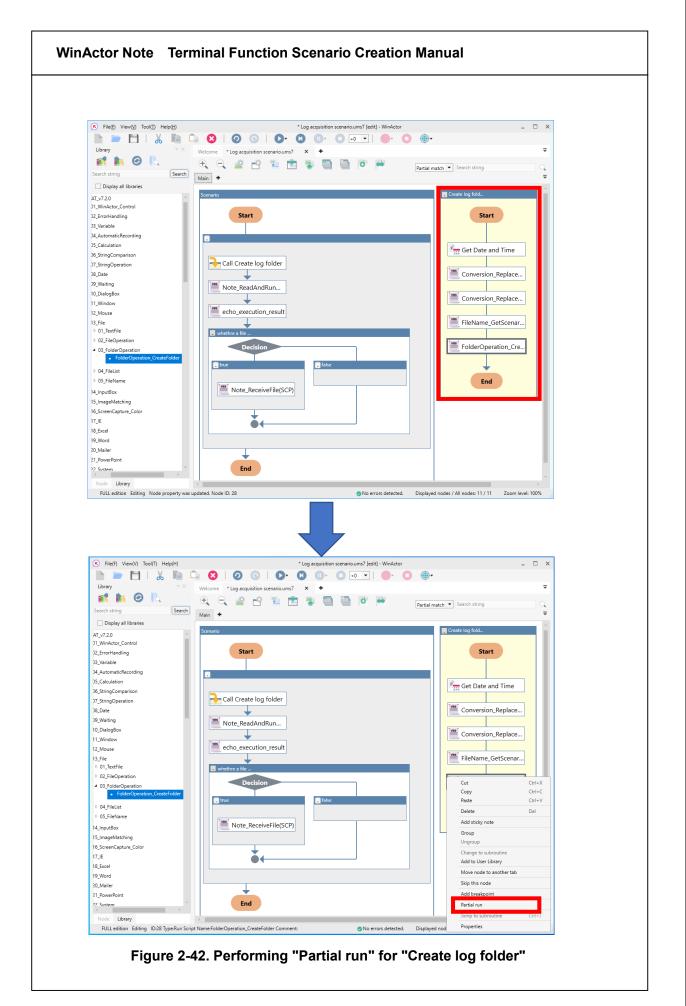
#### Figure 2-38. "FileName\_GetScenarioFilePath" library property settings

					ion Manua		
Varia	ble lis	st					• ×
(X)	2	् 礘 🐻	000	3000	) 🖸 🖻	1 🍬 😨 🗄	ь С
4		Group name	Variable name	Current value	No initializat	Initial value	Mask
	•	VoGroup					
			echo_execution_result				
			Host			192.168.56.2	
			Port_number			22	
			Username			user	
			Password_file			cipherPassword.json	
			Timeout_value			10	
			Source_file			/home/user/log/".log	
			Date				
			Result_folder				

Figure 2-39. Confirming the update of the Variable list pane



Run Script         Name       FolderOperation_CreateFolder         Comment	Property				<del>~</del> ×
Name       FolderOperation_CreateFolder         Comment			Run Script		
Comment         Settings Script Annotation Version         Create a folder.         The folder where the current scenario exists is the base for relative path.         Create_folder_name Result_folder         Update Restore         Figure 2-41. "FolderOperation_CreateFolder" library property setting         eck the operations, perform "Partial run" for the current scenario.         licking and selecting "Create log folder" in the flowchart area, right-click it and selecting	Name	FolderOperation_			
Create a folder.         * The folder where the current scenario exists is the base for relative path.         Create_folder_name         Result_folder         Update         Restore    Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and selection	Comment				
Create a folder.         * The folder where the current scenario exists is the base for relative path.         Create_folder_name         Result_folder         Update         Restore    Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and selection					
The folder where the current scenario exists is the base for relative path.          Create_folder_name       Result_folder         Update       Restore         Figure 2-41. "FolderOperation_CreateFolder" library property setting         exk the operations, perform "Partial run" for the current scenario.         licking and selecting "Create log folder" in the flowchart area, right-click it and selection	[	-	Version		
Create_folder_name       Result_folder         Update       Restore         Figure 2-41. "FolderOperation_CreateFolder" library property setting         eck the operations, perform "Partial run" for the current scenario.         licking and selecting "Create log folder" in the flowchart area, right-click it and selection					
Update       Restore         Figure 2-41. "FolderOperation_CreateFolder" library property setting         ck the operations, perform "Partial run" for the current scenario.         licking and selecting "Create log folder" in the flowchart area, right-click it and selection	* The folder	where the current sco	enario exists is the ba	se for relative path.	
Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and select	Create_f	folder_name Resu	ult_folder		-
Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and select					
Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and select					
Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and select					
Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and select					
Figure 2-41. "FolderOperation_CreateFolder" library property setting eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and select					
Figure 2-41. "FolderOperation_CreateFolder" library property setting ck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and select					
eck the operations, perform "Partial run" for the current scenario. licking and selecting "Create log folder" in the flowchart area, right-click it and selec					
	Figure 2-		eration_CreateF		perty setting
un."		41. "FolderOpe		older" library prop	
	eck the ope	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	ck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope licking and	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).
	eck the ope	<b>41. "FolderOpe</b> rations, perform	"Partial run" for t	older" library prop	).



After performing "Partial run," open C:\Terminal\_function\_scenario in Explorer and confirm that the empty folder is created in the format of ④ in Table 2-2. Delete the created folder after confirmation.

2.6.5. Adding the processing of moving the log file

① Creating a subroutine and the processing of calling the subroutine

Add the processing of moving the file (SSH\_client\_execution\_log.txt) where the display contents of WinActor Note are saved to the log folder.

Drag and drop "Call Subroutine" and "Subroutine Group" from the Node tab.

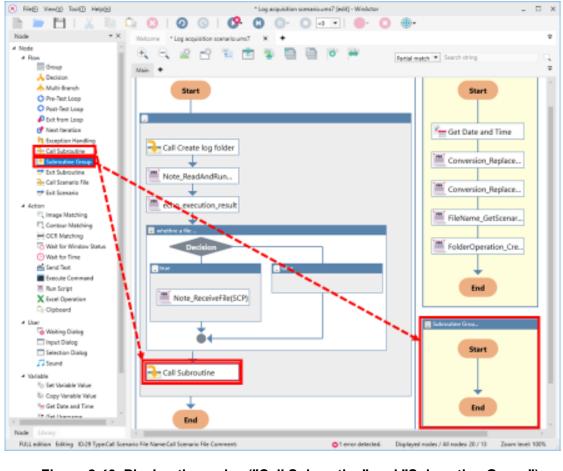


Figure 2-43. Placing the nodes ("Call Subroutine" and "Subroutine Group")

After placing these nodes, open the Property window of each node.

For the "Subroutine Group" node property window, change the name to "Move log file." For the "Call Subroutine" node property window, change the name to "Call Move log file" and select "Move log file" for the subroutine name.

roperty	<b>▼</b> ×	Property		•
Subroutine			Call Subroutine	
Name Move log file		Name Call Move	e log file	
Comment		Comment		
Local variable list (variables to restore the initial value	e at the end)	Subroutine name	Move log file	- Synchronize
Variable Select variable	name  Add Delete	Return value	Select variable name	<ul> <li>(optional)</li> </ul>
Update	estore	Update	R	lestore

Figure 2-44. "Subroutine Group" and "Call Subroutine" property settings

<ol> <li>Creating the p</li> </ol>	processing in the subroutine	
•	-	brouting
-	ocessing in the "Move log file" su	
1. Adding th	e "FileName_GetScenarioFilePa	th" library
Rive View Tool Health	* Log acquisition scenario.um	
lin 📂 🛄 I 🔏 📖		• • • • • •
📫 🖍 Θ 🖪		
Search string Search	Main. +	Tactial match * Search tring
Display all libraries		
α_y/λ≥8 1_WinActor_Control	8	
2, EnorHandling		Find Get Date and Time
8, Variable 4, Automatic Recording	Call Create log folder	
5_Calculation		Conversion_Replace
6_StringComparison 7_StringOperation	Note_ReadAndRun	Comming Sectors
8,Date	echo_execution_result	Conversion_Replace
9,Welting 0_Dialog8ox	CHO_ERECOON_IPSUR	FileName_GetScenar
1_Window	🖬 whethre a file _	
2_Mouse	Decision	FolderOperation_Cre
k,File + 01_TextFile	E fore	
02_FileOperation		
<ul> <li>03_FolderOperation</li> <li>04_FileList</li> </ul>	Mote_ReceiveFile(SCP)	End
4 05_NeNeme		
<ul> <li>FileName_SetDecktopFilePath</li> <li>FileName_SetDecktopFolder</li> </ul>	±.	Mave log He
<ul> <li>Fiel/ame,SetMyDocumentFile</li> </ul>		Start
<ul> <li>Fieldame_SetMyDocumentFold</li> </ul>	+	
FileName_SetScenarioFilePath     HicName_MakePathName	Call Move log file	
		FileName_GetScenar
4_inputBox 5_imageMatching	End	↓ I II
6_SoreenCapture_Color		End
Node Library		
RJILedition Editing Node property was	spdated. Node ID: 84	terror detected. Doplayed nodes / All nodes 14 / 14 Zoans level 102%
i igure z	2-45. Placing the "FileName_G	

Double-click the placed library to open the Property window.

Enter "SSH\_client\_execution\_log.txt" for the filename and "SSH\_log\_filename\_absolute\_path" for the file path generation result, and click the "Update" button.

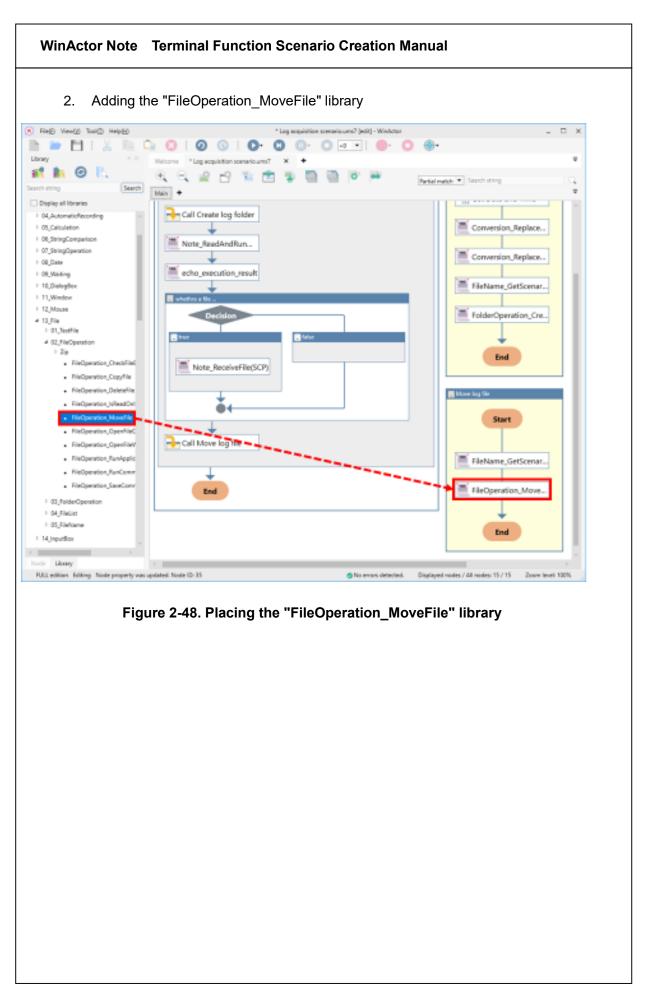
A window for confirming whether to register the entered value as a new variable will appear. Click "Yes," and it will be registered as a new variable in the Variable list pane of WinActor.

operty			
			Run Script
Name	e FileNa	ame_GetScen	narioFilePath
Commen	t		
Cattings	Corint	Annotation	Version
		-	ENARIO folder. If SCENARIO is not saved,
mormation	will be e	mpty.	
e.g.) When S	SCENARIO	D is saved in	c:\temp folder and a txt is specified as the file
e.g.) When S	SCENARIO	D is saved in	c:\temp folder and a.txt is specified as the file le path generation result.
e.g.) When S	SCENARIO np\a.txt i	O is saved in a stored as fil	le path generation result.
e.g.) When S	SCENARIO np\a.txt i	O is saved in a stored as fil	
e.g.) When S name, c:\ten	SCENARIO np\a.txt i	D is saved in a stored as fil ame Value=	le path generation result.
e.g.) When S name, c:\ten	SCENARIO np\a.txt is File_na	D is saved in a stored as fil ame Value=	<pre>le path generation result. &gt; SSH_client_execution_log.txt </pre>
e.g.) When S name, c:\ten	SCENARIO np\a.txt is File_na	D is saved in a stored as fil ame Value=	<pre>le path generation result. &gt; SSH_client_execution_log.txt </pre>
e.g.) When S name, c:\ten	SCENARIO np\a.txt is File_na	D is saved in 6 s stored as fil ame Value= esult SSH_	<pre>le path generation result. &gt; SSH_client_execution_log.txt </pre>

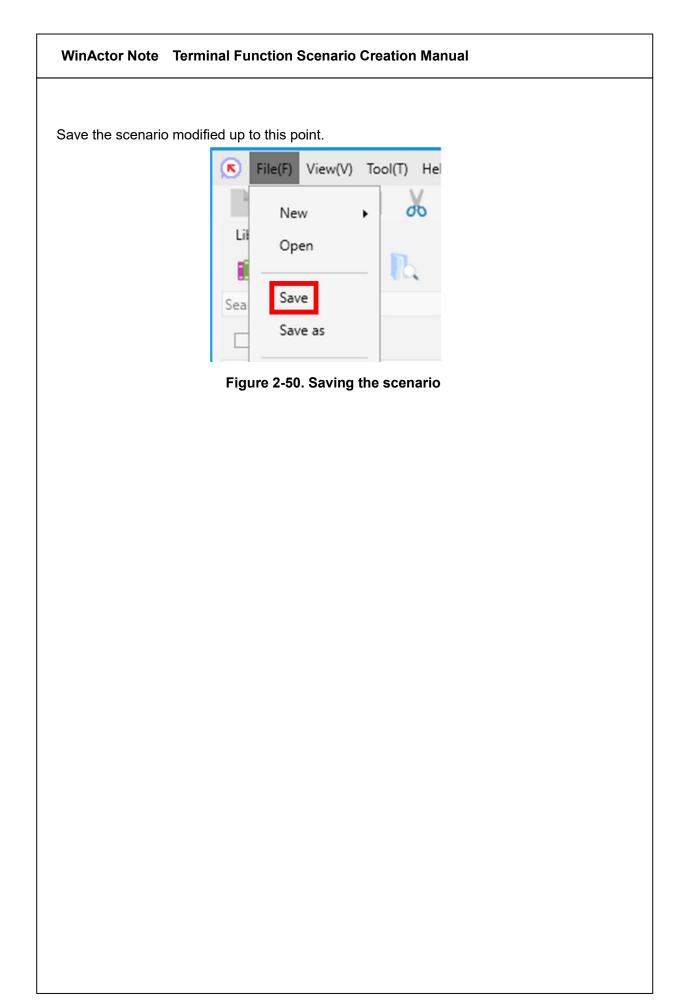
Figure 2-46. "FileName\_GetScenarioFilePath" library property settings

ariable	list					• X					
(xy)	🔍 👒 🐻	0000	000	3 🗹 🕨	s 😎 💿 🗣						
	Group name	Variable name	Current value	No initializat	Initial value	Ma					
	NoGroup										
		echo_execution_result									
		Host			192.168.56.2						
		Port_number			22						
		Username			user						
		Password_file			cipherPassword.json						
		Timeout_value			10						
		Source_file			/home/user/log/".log						
		Date									
		Result_folder									
		SSH_log_filename_absolute_path									

Figure 2-47. Confirming the update of the Variable list pane



_	g_filename_absolute_pa	aut tor wove_iton		
	lick the "Update" button.		, enter Result	
Property			•	· ×
	Run	Script		
Name	FileOperation_MoveFile			
Comment				
Settings S	cript Annotation Versio	on		
"Move_to": Specify a des If a folder is without char If you want t include the r * The folder w	irce file to move as an absol stination folder or file as abs specified, the file is moved to iging the filename. o change the filename in the new filename. here the current scenario ex	olute path or relative p o the specified folder e destination folder, ists is the base for rela name_absolute_path	oath.	
	Update	Rest	ore	



#### 2.7. Checking the operations

Check the final operations of the scenario.

Click the "Run scenario" button on the toolbar of WinActor.



Figure 2-51. Running the scenario

After running the scenario, the folder at the date and time when the scenario was run will be created in the "C:\Terminal\_function\_scenario" folder, and three files of "server#1.log," "server#2.log," and "SSH\_client\_execution\_log.txt" will be acquired.

Figure 2-52 shows the example of the acquired "SSH\_client\_execution\_log.txt."

```
Last login: Thu Aug 29 14:07:12 2019 from 192.168.56.1
[user@demoserver ~]$ cd log
[user@demoserver log]$ date
Thursday, August 29 2019 14:35:54 JST
[user@demoserver log]$ ls *.log
server#1.log server#2.log
[user@demoserver log]$ echo $?
O
[user@demoserver log]$
```

#### Figure 2-52. Example of the acquired execution log

# 3. Library and property list

# 3.1. Shell tool

This section describes the libraries for using PowerShell and Command Prompt and the procedure to create a basic scenario using those libraries. The property of each library related to "Shell tool" is described in 3.1.1 to 3.1.4.

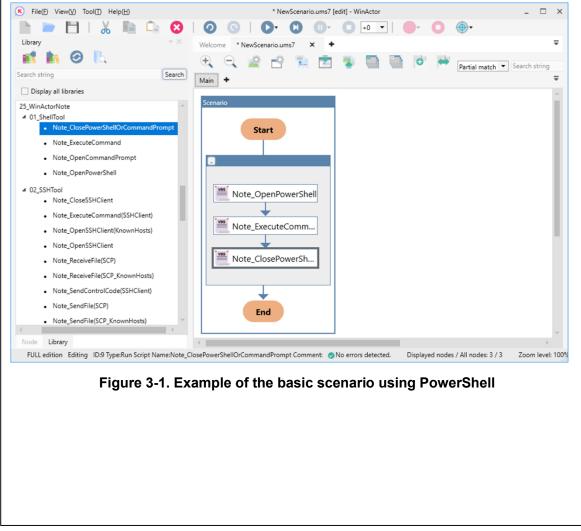
[Procedure to execute PowerShell]

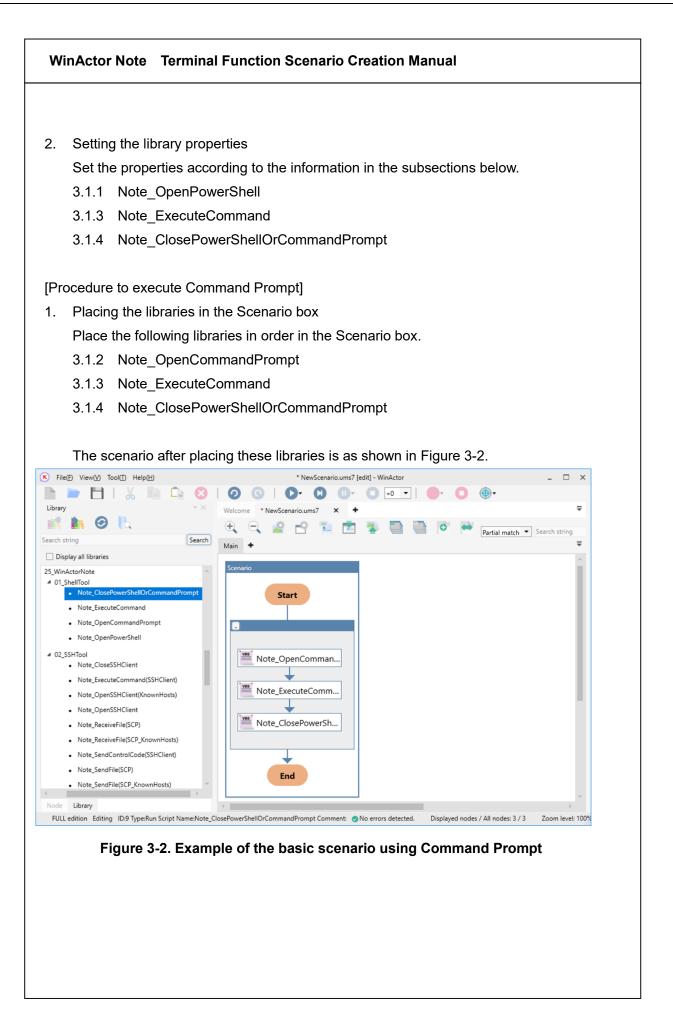
1. Placing the libraries in the Scenario box

Place the following libraries in order in the Scenario box.

- 3.1.1 Note\_OpenPowerShell
- 3.1.3 Note\_ExecuteCommand
- 3.1.4 Note\_ClosePowerShellOrCommandPrompt

The scenario after placing these libraries is as shown in Figure 3-1.





2. Setting the library properties

Set the properties according to the information in the subsections below.

- 3.1.2 Note\_OpenCommandPrompt
- 3.1.3 Note\_ExecuteCommand
- 3.1.4 Note\_ClosePowerShellOrCommandPrompt

#### 3.1.1. Note\_OpenPowerShell

This library is to open PowerShell.

#### Table 3-1. "Note\_OpenPowerShell" library property setting

No.	Item	Description	
1	Encoding	Specify a character encoding for importing into	
		PowerShell and exporting to WinActor Note.	

#### 3.1.2. Note\_OpenCommandPrompt

This library is to open Command Prompt.

#### Table 3-2. "Note\_OpenCommandPrompt" library property setting

No.	Item	Description	
$\bigcirc$	Encoding	Specify a character encoding for importing into	
		Command Prompt and exporting to WinActor Note.	

#### 3.1.3. Note\_ExecuteCommand

This library is to execute a command on PowerShell or Command Prompt.

#### Table 3-3. "Note\_ExecuteCommand" library property setting

No.	ltem	Description
1	Command	Specify a command you want to execute on PowerShell or
		Command Prompt. Only text can be entered. Control
		characters cannot be sent.

3.1.4. Note\_ClosePowerShellOrCommandPrompt

This library is to close a session of PowerShell or Command Prompt.

# 3.2. SSH tool

This section describes the libraries for using the SSH client function and the file transfer function (SCP) provided in "SSH tool" and the procedure to create a basic scenario using those libraries. The setting methods to connect to a common server with the SSH client file transfer function (SCP) are described in 3.2.1. The property of each library related to "SSH tool" is described in 3.2.2 to 3.2.9.

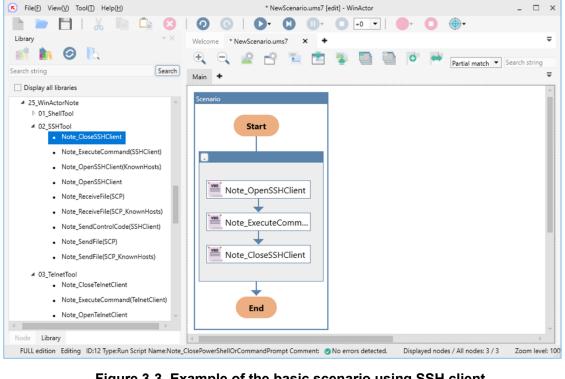
[Procedure to execute the SSH client function]

1. Placing the libraries in the Scenario box

Place the following libraries in order in the Scenario box.

- 3.2.2 Note\_OpenSSHClient
- 3.2.4 Note\_ExecuteCommand(SSHClient)
- 3.2.5 Note\_CloseSSHClient

The scenario after placing these libraries is as shown in Figure 3-3.



- Figure 3-3. Example of the basic scenario using SSH client
- Setting the library properties
   Set the properties according to the information in the subsections below.

- 3.2.2 Note\_OpenSSHClient
- 3.2.4 Note\_ExecuteCommand(SSHClient)
- 3.2.5 Note\_CloseSSHClient

[Procedure to execute the function to send a file (SCP)]

- Placing the library in the Scenario box Place the following library in the Scenario box.
   3.2.6 Note\_SendFile(SCP)
- Setting the library property
   Set the property according to the information in the subsection below.
   3.2.6 Note\_SendFile(SCP)

[Procedure to execute the function to receive a file (SCP)]

- Placing the library in the Scenario box Place the following library in the Scenario box.
   3.2.8 Note\_ReceiveFile(SCP)
- Setting the library property
   Set the property according to the information in the subsection below.
   3.2.8 Note\_ReceiveFile(SCP)

#### 3.2.1. Connection settings

This subsection describes the setting methods to connect to a common server with the SSH client file transfer function (SCP).

In "SSH tool," two authentication methods, Password authentication and Public key authentication, are available to connect to an SSH server. The following describes the setting items required to connect to an SSH server for each of the above two authentication methods.

#### [Password authentication]

- Generate a password file that contains the password information for connecting to an SSH server in advance. For details on how to generate a password file, see the section of "Password file generation tool" in the material No.3 in Table 5-1.
- 2. Set the values according to Table 3-4. The setting values described here follow the example in Table 2-1. Environment used in this tutorial. Set the values according to your environment.

Item	Description	Setting values in the example in this document
Authentication_method	Select the authentication method.	Password authentication
Character_encoding	Specify a character encoding for importing into the server and exporting to WinActor Note.	UTF-8
Line-break_code_for_sending	Specify a line-break code for sending to the server.	LF(Linux, MacOS X)
Host	Specify an IPv4 address of the SSH server to be connected.	192.168.56.2
Port	Specify a port number of the SSH server to be connected.	22
User_name	Specify a login name when logging in to the SSH server.	user
Password_file	Specify a password file	secret\sshLogin.json

# Table 3-4. Settings for password authentication

	that contains the password required to log in to the SSH server. Specify a relative path from the folder where the scenario file is located.	* This is an example of the case where the folder containing the scenario file has the secret folder and the password file (sshLogin.json) is located in that folder.
Private_key_file	It is not necessary to set when using password authentication.	
Prompt_string	Specify a string containing the end of the prompt that will be displayed when the login process is completed. If you want to specify more than one, enter with comma-separated values. (Example) Enter "\$,#" for specifying "\$" and "#."	"\$ " " is not required. There is a space after \$.
Timeout_value[sec]	Specify a maximum wait time in each step of the login process to the SSH server in seconds. Adjust with an appropriate value according to your environment.	10 * This setting value is just an example. Note that it may not operate depending on the environment.
Known_hosts_file	Specify a known hosts file for the hosts whose connections have been confirmed with the "Known hosts file generation tool." Specify a relative path from the folder where the scenario file is located.	[Only when setting a known hosts file] destination_hosts * This is an example of the case where the known hosts file (destination_hosts) is located directly under the folder where the scenario

	file is located.

[Public key authentication]

- 1. Generate a private key. For details on how to generate a private key, see the section of "SSH key generation tool" in the material No.3 in Table 5-1.
- 2. If you set a passphrase when generating a private key, generate a password file. For details on how to generate a password file, see the section of "Password file generation tool" in the material No.3 in Table 5-1.
- 3. Set the values according to Table 3-5. The setting values described here follow the example in Table 2-1. Environment used in this tutorial. Set the values according to your environment.

Item	Description	Setting values in the example in this document		
Authentication_method	Select the authentication method.	Public key authentication		
Character_encoding	Specify a character encoding for importing into the server and exporting to WinActor Note.	UTF-8		
Line-break_code_for_sending	Specify a line-break code for sending to the server.	LF(Linux, MacOS X)		
Host	Specify an IPv4 address of the SSH server to be connected.	192.168.56.2		
Port	Specify a port number of the SSH server to be connected.	22		
User_name	Specify a login name when logging in to the SSH server.	user		
Password_file	Specify a password file for the private key with a passphrase. Specify a	[Only if a passphrase for the private key is set] secret\sshLoginPassphras		

# Table 3-5. Settings for public key authentication

WinActor Note Terminal Function Scenario Creation Manual					
	relative path from the	e.json			
	folder where the scenario	* This is an example of the			
	file is located.	case where the folder			
		containing the scenario			
	It is not necessary to set	file has the secret folder,			
	if you do not set a	and the password file			
	passphrase for the	(sshLoginPassphrase.json			
	private key.	) is located in that folder.			
Private_key_file	Specify a private key file.	secret\sshLoginKey			
	Specify a relative path	* This is an example of the			
	from the folder where the	case where the folder			
	scenario file is located.	containing the scenario			
		file has the secret folder,			
		and the private key file			
		(sshLoginKey) is located			
		in that folder.			
Prompt_string	Specify a string	"\$ "			
	containing the end of the	" is not required. There is a			
	prompt that will be	space after \$.			
	displayed when the login				
	process is completed.				
	If you want to specify				
	more than one, enter with				
	comma-separated				
	values. (Example) Enter				
	"\$,#" for specifying "\$"				
	and "#."				
Timeout_value[sec]	Specify a maximum wait	10			
	time in each step of the	* This setting value is just			
	login process to the SSH	an example. Note that it			
	server in seconds. Adjust				
	with an appropriate value	depending on the			
	according to your	environment.			
	environment.				
Known hosts file	Specify a known hosts	[Only when setting a			
	file for the hosts whose	known hosts file]			

WinActor Note Terminal Function Scenario Creation Manua	on Manual
---	-----------

connections	s have	been	destir	nation_hosts	;
confirmed	with	the	* This	is an exam	ple of the
"Known	hosts	file	case	where the	e known
generation	tool." S	pecify	hosts		file
a relative p	oath fro	m the	(desti	nation_host	s) is
folder where	e the sc	enario	locate	ed directly ι	under the
file is locate	ed.		folder	where the	scenario
			file is	located.	

3.2.2. Note\_OpenSSHClient

This library is to open an SSH client.

# Table 3-6. "Note\_OpenSSHClient" library property settings

No.	Item	Description
1	Authentication_method	See 3.2.1.
2	Character_encoding	
3	Line-break_code_for_sending	
4	Host	
5	Port	
6	User_name	
$\overline{O}$	Password_file	
8	Private_key_file	
9	Prompt_string	
10	Timeout_value[sec]	

# 3.2.3. Note\_OpenSSHClient(KnownHosts)

This library is to open an SSH client by specifying a known hosts file.

For details on how to generate a known hosts file, see the section of "Known hosts file generation tool" in the material No.3 in Table 5-1.

#### Table 3-7. "Note\_OpenSSHClient(KnownHosts)" library property settings

No.	ltem	Description
1)	Authentication_method	See 3.2.1.
2	Character_encoding	
3	Line-break_code_for_sending	
4	Host	
5	Port	
6	User_name	
$\bigcirc$	Password_file	
8	Private_key_file	
9	Prompt_string	
10	Timeout_value[sec]	
(1)	Known_hosts_file	

# 3.2.4. Note\_ExecuteCommand(SSHClient)

This library is to execute a command on an SSH client.

# Table 3-8. "Note\_ExecuteCommand(SSHClient)" library property settings

No.	Item	Description
1	Command	Specify a command you want to execute on the SSH
		client. Only text can be entered. Control characters
		cannot be sent.
2	Prompt_string	Specify a string containing the end of the message
		indicating that the command processing has been
		completed.
		It will usually be the same as the "Prompt_string" in
		3.2.1.
3	Timeout_value[sec]	Specify a maximum wait time until the string of $\textcircled{2}$ is
		displayed.

3.2.5. Note\_CloseSSHClient

This library is to close a session of SSH client.

3.2.6. Note\_SendFile(SCP)

This library is to send a file with SCP.

# Table 3-9. "Note\_SendFile(SCP)" library property settings

No.	ltem	Description
1	Authentication_method	See 3.2.1.
2	Host	
3	Port	
4	User_name	
5	Password_file	
6	Private_key_file	
$\overline{\mathcal{O}}$	Timeout_value[sec]	
8	Source_file	Specify a file to be sent with a relative path from the
		folder where the scenario file is located. Only a single
		file can be specified. Multiple files or a folder cannot
		be specified.
9	Destination_path	Enter a storage destination path for the source file $(8)$
		on the SSH server.

# 3.2.7. Note\_SendFile(SCP\_KnownHosts)

This library is to send a file with SCP by specifying a known hosts file.

For details on how to generate a known hosts file, see the section of "Known hosts file generation tool" in the material No.3 in Table 5-1.

No.	Item	Description
1	Authentication_method	See 3.2.1.
2	Host	
3	Port	
4	User_name	
5	Password_file	
6	Private_key_file	
$\bigcirc$	Timeout_value[sec]	
8	Known_hosts_file	
9	Source_file	Specify a file to be sent with a relative path from the
		folder where the scenario file is located. Only a
		single file can be specified. Multiple files or a folder
		cannot be specified.
10	Destination_path	Enter a storage destination path for the source file
		(9) on the SSH server.

# Table 3-10. "Note\_SendFile(SCP\_KnownHosts)" library property settings

3.2.8. Note\_ReceiveFile(SCP)

This library is to receive a file with SCP.

# Table 3-11. "Note\_ReceiveFile(SCP)" library property settings

No.	ltem	Description
1	Authentication_method	See 3.2.1.
2	Host	
3	Port	
4	User_name	
5	Password_file	
6	Private_key_file	
$\overline{O}$	Timeout_value[sec]	
8	Source_file	Specify a file to be received on the SSH server. A
		folder cannot be specified. In the Environment used
		in this tutorial, wildcards (* and ?) can be used for a
		file.
9	Destination_path	Enter a storage destination path for the source file $(8)$
		on the SSH server.

#### 3.2.9. Note\_ReceiveFile(SCP\_KnownHosts)

This library is to receive a file with SCP by specifying a known hosts file.

For details on how to generate a known hosts file, see the section of "Known hosts file generation tool" in the material No.3 in Table 5-1.

No.	Item	Description
1	Authentication_method	See 3.2.1.
2	Host	
3	Port	
4	User_name	
5	Password_file	
6	Private_key_file	
$\overline{\mathcal{O}}$	Timeout_value[sec]	
8	Known_hosts_file	
9	Source_file	Specify a file to be received on the SSH server. A
		folder cannot be specified. In the Environment used
		in this tutorial, wildcards (* and ?) can be used for a
		file.
10	Destination_path	Enter a storage destination path for the source file $(9)$
		on the SSH server.

#### Table 3-12. "Note\_ReceiveFile(SCP\_KnownHosts)" library property settings

# 3.2.10. Note\_SendControlCode(SSHClient)

This library is to send a control code on an SSH client.

# Table 3-13. "Note\_SendControlCode(SSHClient)" library property settings

No.	Item	Description
1	Control_code	Specify a control code to be sent by an SSH client.
		Ctrl+A to Ctrl+Z can be specified.
2	Prompt_string	Specify a string containing the end of the message
		indicating that the command processing has been
		completed.
		It will usually be the same as the "Prompt_string" in
		3.2.1.
3	Timeout_value[sec]	Specify a maximum wait time until the string of $\textcircled{2}$ is
		displayed.

# 3.3. Telnet tool

This section describes the libraries for using the Telnet client function and the procedure to create a basic scenario using those libraries. The property of each library related to "Telnet " is described in 3.3.1 to 3.3.3.

[Procedure to execute the Telnet client function]

1. Placing the libraries in the Scenario box

Place the following libraries in order in the Scenario box.

- 3.3.1 Note\_OpenTelnetClient
- 3.3.2 Note\_ExecuteCommand(TelnetClient)
- 3.3.3 Note\_CloseTelnetClient

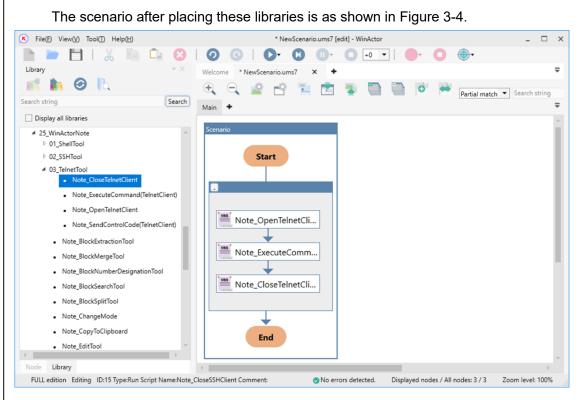


Figure 3-4. Example of the basic scenario using the Telnet client function

2. Setting the library properties

Set the properties according to the information in the subsections below.

- 3.3.1 Note\_OpenTelnetClient
- 3.3.2 Note\_ExecuteCommand(TelnetClient)
- 3.3.3 Note\_CloseTelnetClient

3.3.1. Note\_OpenTelnetClient

This library is to open a Telnet client.

Table 3-14. "No	ote OpenTelnetClient'	' library property settings
		instally property cottainge

No.	Item	Description
1	Character_encoding	Specify a character encoding for importing into the
		server and exporting to WinActor Note.
2	Host	Specify an IPv4 address of the Telnet server to be
		connected.
3	Port	Specify a port number of the Telnet server to be
		connected.
4	User_name	Specify a login name when logging in to the Telnet
		server.
5	Password_file	Specify a password file that contains the password
		required to log in to the Telnet server. Specify a
		relative path from the folder where the scenario file
		is located.
6	Login_prompt_string	Specify a string containing the end of the message
		that the Telnet server prompts for a username.
$\bigcirc$	Password_prompt_string	Specify a string containing the end of the message
		that the Telnet server prompts for a password.
8	Command_prompt_string	Specify a string containing the end of the prompt that
		will be displayed when the login process is
		completed.
		If you want to specify more than one, enter with
		comma-separated values. (Example) Enter "\$,#" for
		specifying "\$" and "#."
9	Timeout_value[sec]	Specify a maximum wait time in each step of the
		login process to the Telnet server in seconds.

	Adjust with an appropriate value according to your
	environment.

\*Line-break code for sending is fixed to CR+LF.

3.3.2. Note\_ExecuteCommand(TelnetClient)

This library is to execute a command on a Telnet client.

# Table 3-15. "Note\_ExecuteCommand(TelnetClient)" library property settings

No.	Item	Description
1	Command	Specify a command you want to execute on a Telnet
		client. Only text can be entered. Control characters
		cannot be sent.
2	Prompt_string	Specify a string containing the end of the message
		indicating that the command processing has been
		completed.
		It will usually be the same as the
		"Command_prompt_string" in 3.3.1.
3	Timeout_value[sec]	Specify a maximum wait time until the string of $②$ is
		displayed.

# 3.3.3. Note\_CloseTelnetClient

This library is to close a session of Telnet client.

# 3.3.4. Note\_SendControlCode(TelnetClient)

This library is to send a control code on a Telnet client.

No.	Item	Description
1	Control_code	Specify a control code to be sent by a Telnet client.
		Ctrl+A to Ctrl+Z can be specified.
2	Prompt_string	Specify a string containing the end of the message
		indicating that the command processing has been
		completed.
		It will usually be the same as

		"Command_prompt_string" in 3.3.1.
3	Timeout_value[sec]	Specify a maximum wait time until the string of $\textcircled{2}$ is
		displayed.

# 4. Docking window

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

# 5. Reference materials

Table 5-1 shows the materials referenced in this manual.

#### Table 5-1. Reference materials

No.	Material name	
1	WinActor Operation Manual	
2	WinActor User Library Sample Manual	
3	WinActor Note Operation ManualWinActor Note Text Processing Scenario Creation Manual	
4		



#### NTT ADVANCED TECHNOLOGY CORPORATION

Copyright © 2013-2025 NTT, Inc. & NTT ADVANCED TECHNOLOGY CORPORATION

This document is protected under copyright law. It is forbidden to duplicate or copy any part or all of this document without prior consent.

The contents of this document are subject to change without notice.

WA7-P-20250603