



WinActor ノート

端末機能シナリオ
作成マニュアル

NTTアドバンステクノロジー株式会社

目次

1. はじめに	1
1.1. 本書の位置づけと概要	1
1.2. ライセンス	2
1.3. 本書に関する注意	2
2. シナリオ作成チュートリアル	3
2.1. 事前準備	3
2.2. シナリオの作成方法	4
2.3. シナリオ作成手順	5
2.4. シナリオの全体像	7
2.5. WinActor ノートのマクロ作成	8
2.5.1. WinActor ノートの画面クリア	8
2.5.2. SSH ツールでの処理	9
2.5.3. テキスト処理	13
2.5.4. 画面上のテキストデータのファイル保存	15
2.5.5. マクロの保存	16
2.6. WinActor のシナリオ作成	17
2.6.1. 作成したマクロの呼び出し	18
2.6.2. 判定処理の追加	21
2.6.3. ファイル受信処理の追加	27
2.6.4. ログ保存用フォルダの作成処理の追加	31
2.6.5. ログファイルの移動処理の追加	46
2.7. 動作確認	54
3. ライブラリ・プロパティの一覧	55
3.1. シェルツール	55
3.1.1. PowerShell を開く	58
3.1.2. コマンドプロンプトを開く	58
3.1.3. コマンドを実行する(PowerShell, コマンドプロンプト)	58
3.1.4. PowerShell・コマンドプロンプトを閉じる	58
3.2. SSH ツール	59
3.2.1. 接続設定	61
3.2.2. SSH クライアントを開く	65
3.2.3. SSH クライアントを開く(接続先一覧設定)	66
3.2.4. コマンドを実行する(SSH クライアント)	67
3.2.5. SSH クライアントを閉じる	67

3.2.6.	ファイル送信(SCP).....	68
3.2.7.	ファイル送信(SCP・接続先一覧設定).....	69
3.2.8.	ファイル受信(SCP).....	70
3.2.9.	ファイル受信(SCP・接続先一覧設定).....	71
3.2.10.	制御コードを送信する(SSH クライアント).....	72
3.3.	Telnet ツール	73
3.3.1.	Telnet クライアントを開く	74
3.3.2.	コマンドを実行する(Telnet クライアント).....	75
3.3.3.	Telnet クライアントを閉じる.....	75
3.3.4.	制御コードを送信する(Telnet クライアント).....	75
4.	ドッキングウィンドウ.....	76
5.	参照資料	77

1. はじめに

1.1. 本書の位置づけと概要

本書は WinActor ノートと連携して動作する端末機能のシナリオを作成するためのマニュアルです。

端末機能では以下の機能が提供されます。

- ・ Windows シェル機能
Windows PowerShell、コマンドプロンプト上でのコマンド実行機能
- ・ SSH/SCP クライアント機能
SSH サーバへ接続・切断、コマンド実行機能
SSH サーバへのファイル転送機能(SCP)
- ・ Telnet クライアント機能
Telnet サーバへの接続・切断、コマンド実行機能

端末機能は WinActor ノートのマクロ機能と組み合わせて使用することで、コマンドの実行結果に応じたより細かい制御が可能になります。

1.3 章では、CentOS 上に構築された SSH サーバに接続してコマンドを実行し、コマンドの出力結果に応じて、ログファイルを取得するシナリオを作成します。本チュートリアルを通じて WinActor ノートの端末機能を体験することができます。

なお WinActor ノートの使用方法については、表 5-1 参照資料の No.3,4 の資料をご参照ください。

1.2. ライセンス

本書において以下に記載された名称、およびその他記載されている会社名、製品名は、各社の登録商標または商標です。TM、®、©マークは省略しています。

- ・ Windows、Windows PowerShell は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。
 - * Windows の正式名称は、Microsoft Windows Operating System です。
- ・ CentOS の名称は、CentOS ltd.の商標または登録商標です。
- ・ Mac OS、OS X は、Apple Inc.の商標です。
- ・ Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

1.3. 本書に関する注意

- ・ 本書および提供するソフトウェア類に付された著作権表示「Copyright © 2013-2025 NTT, Inc. & NTT ADVANCED TECHNOLOGY CORPORATION」の変更、削除をすることはできません。
本書の著作権は NTT 株式会社及び NTT アドバンステクノロジー株式会社に帰属します。
- ・ 本書では、Windows の操作方法や機能を理解されていることを前提として説明しています。本書に記載されていないことについては、Microsoft が提供しているドキュメントなどをご覧ください。

2. シナリオ作成チュートリアル

2.1. 事前準備

本チュートリアルで使用する環境を表 2-1 に示します。WinActor のシナリオを実行する PC と SSH サーバとの間で、あらかじめ SSH プロトコルの導通確認が完了している状況を前提とします。

表 2-1 本チュートリアルで使用する環境

対象	項目	内容	本チュートリアルでの設定値
サーバ	サーバ OS	CentOS Linux release 7.6.1810	-
	SSH サーバ	OpenSSH 7.4p1	IP アドレス:192.168.56.2 ポート番号: 22
	動作検証用 アカウント	任意	ユーザ名:user パスワード:ax12bc=9
WinActor を動作さ せる PC	IP アドレス	任意	IP アドレス:192.168.56.1
	シナリオ格納 フォルダ	任意	C:\¥端末機能シナリオ
	パスワード ファイル	任意。ただし、シナリオ 格納フォルダに格納し てください。	パスワード:ax12bc=9、ファイ ル名:cihperPassword.json と してシナリオ格納フォルダに 格納してください。詳細は 2.3 をご参照ください。
	Windows	エディション: Windows 10 Pro 日本語版	

2.2. シナリオの作成方法

端末機能を使用したシナリオの作成方法には 2 種類あります。

- ① ユーザライブラリのみで作成する方法
- ② WinActor ノートのマクロ機能を使用する方法

2.3 章で作成するシナリオのように WinActor ノート上で連続して処理をさせる場合、WinActor ノートのマクロ記録を行った上で動作確認をし、作成したマクロをマクロ読み込み実行ライブラリで動作させることで、シナリオを効率的に作成できます。

2.3. シナリオ作成手順

作成するシナリオの内容と動作を以下に示します。

[作成するシナリオ(図 2-1 ①②)の内容]

- ・ SSH クライアントを開き、/home/user/log フォルダ下に拡張子が log のファイルが存在する場合(図 2-1 ⑦)、拡張子が log のファイルを取得します(図 2-1 ⑥)
- ・ SSH クライアントの操作をログとして保存します(図 2-1 ⑤)
- ・ WinActor が動作する PC 上に日付と時刻を含む名称のフォルダ(図 2-1 ④)を作成し、ログを格納します

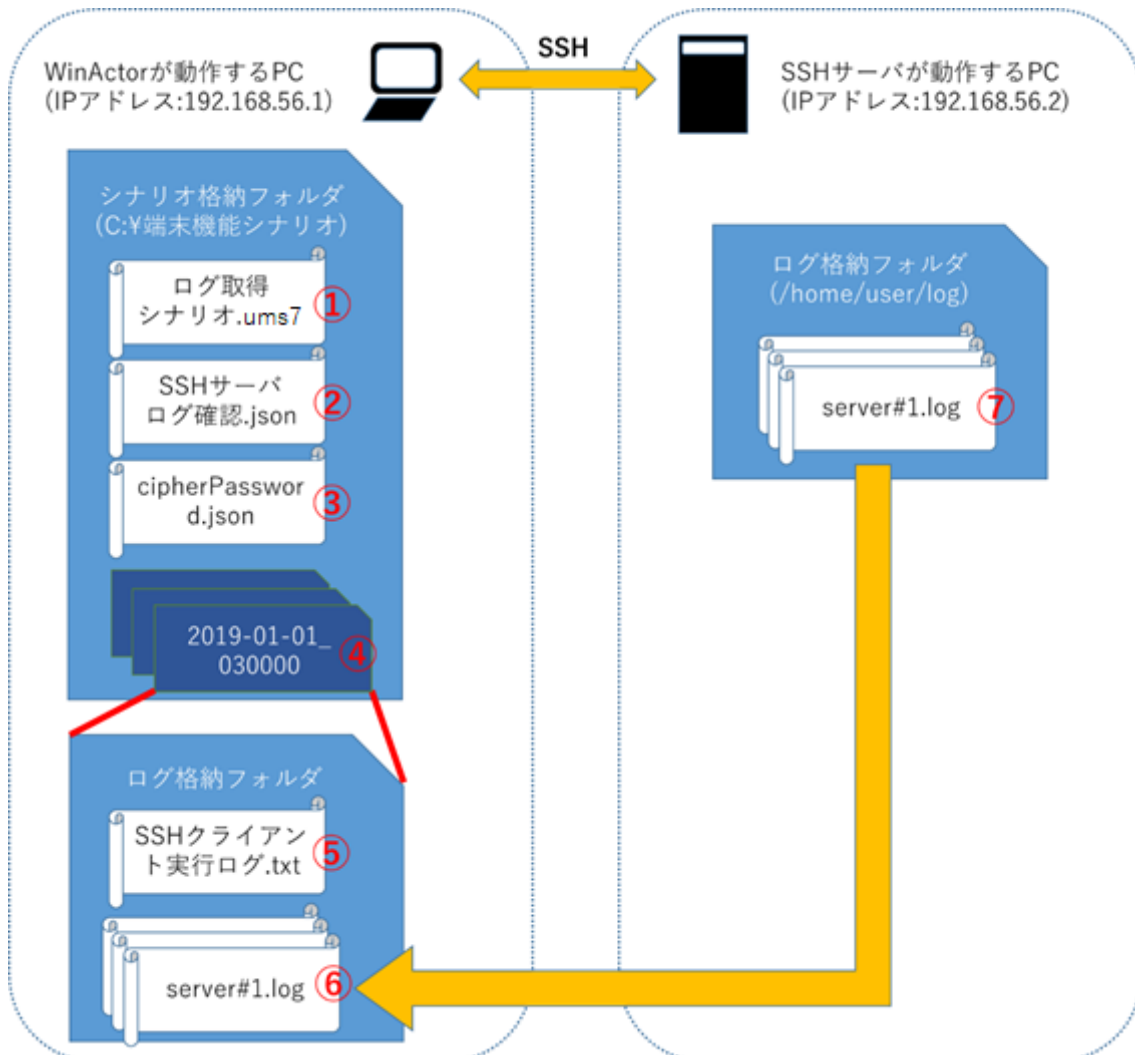


図 2-1 シナリオ動作

表 2-2 各種ファイル、フォルダの説明

図 2-1 の番号	ファイル、フォルダ名	説明
①	ログ取得シナリオ.ums7	本チュートリアルで作成する WinActor のシナリオです。ここでは WinActor Ver.7 でのシナリオを例とします。
②	SSH サーバログ確認.json	本チュートリアルで作成する WinActor ノートのマクロです。
③	cipherPassword.json	SSH サーバにログインする際のパスワードファイルです。表 5-1 の No.3 の資料の「 <u>パスワードファイル生成ツール</u> 」を参照して、 <u>事前に作成してください。</u>
④	2019-01-01_030000 など	①のシナリオ実行により作成されるログを格納するためのフォルダで、WinActor が動作する PC 上に作成されます。フォルダが作成された日時となり、左記の場合は 2019 年 1 月 1 日 3 時 00 分 00 秒を意味します。
⑤	SSH クライアント実行ログ.txt	②のマクロ実行中に WinActor ノートに表示された内容が本ファイルに保存されます。本チュートリアルで作成するシナリオでは SSH サーバへのログインメッセージ、コマンドの実行と結果が情報として残ります。
⑥	server#1.log など	①のシナリオ実行により SSH サーバが動作する PC から取得した拡張子 log のファイルです。
⑦	server#1.log など	SSH サーバが動作する PC 上に存在する拡張子 log のファイルです。 <u>本チュートリアルで作成するシナリオでは本ファイルを取得することを目的としています。SSH サーバ上にファイルがない場合は事前にファイルを作成してください。</u> なお、ファイルの中身としては 10 文字未満のテキストデータを使用しています。

作成するシナリオでは、WinActor のノード、ユーザライブラリを使用しています。シナリオ中のノード、ユーザライブラリの内容については、表 5-1 の No.1,2 の資料をご参照ください。

また作成するシナリオでは、一般的な Linux のコマンドを使用しています。各コマンドの内容については Linux 関連の書籍、Web ページ等でご確認ください。

2.4. シナリオの全体像

シナリオを作成するための全体像を以下に示します。

1. WinActor ノートのマクロ作成(2.5 章参照)

- A) SSH サーバ上のログ格納フォルダ内に拡張子 `log` のファイルがあるかを判断可能な値を取得し、WinActor 変数に取り込むまでの処理を作成。その際、WinActor ノートを起動してマクロ記録を行いながらマクロを作成。

2. WinActor のシナリオ作成(2.6 章参照)

- A) WinActor のシナリオから作成したマクロの呼び出す処理を追加。
- B) 拡張子 `log` のファイルの有無の判定処理を追加。
- C) 拡張子 `log` のファイルが存在する場合は拡張子 `log` のファイルを受信する処理を追加。
- D) その他、ログを保存するフォルダの作成、ログの移動等の処理を追加。

2.5. WinActor ノートのマクロ作成

WinActor ノートの操作については、最新の表 5-1 の No.3 の資料をご参照ください。

WinActor ノートを起動しておき、かつマクロ記録を有効にします。

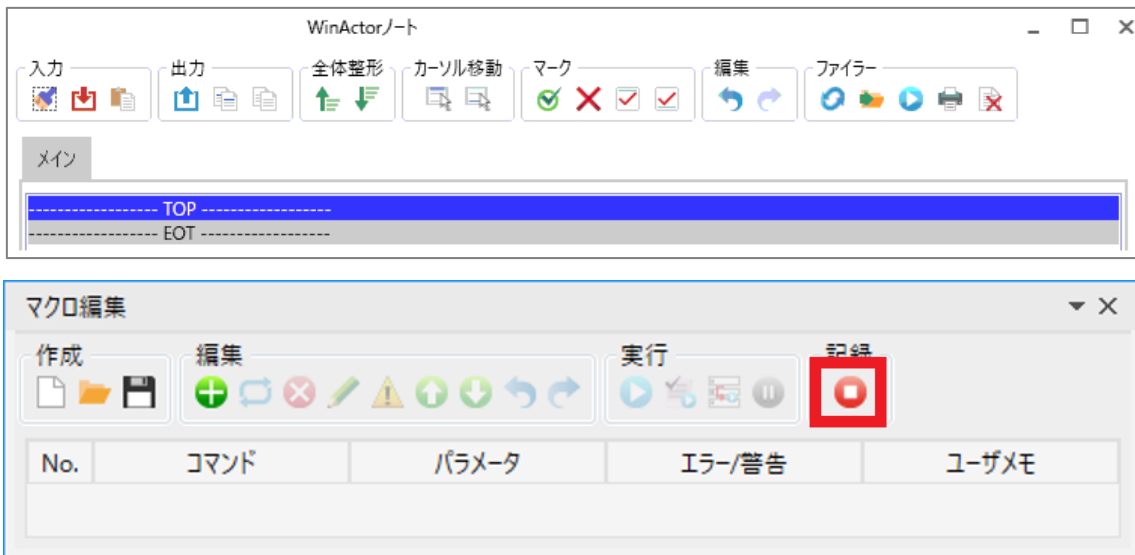


図 2-2 マクロ作成開始時点の WinActor ノート

2.5.1. WinActor ノートの画面クリア

前回の実行時のログが WinActor ノートに残らないように WinActor ノートの画面クリアを行います。

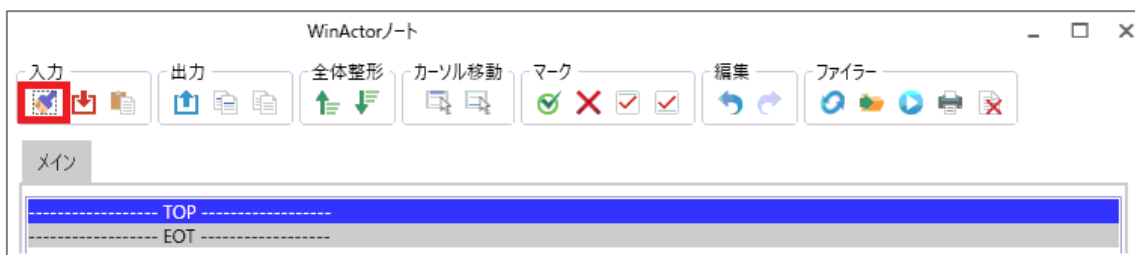


図 2-3 画面クリア

2.5.2. SSH ツールでの処理

SSH ツールでの処理を以下に示します。

1. SSH クライアントを開く

WinActor ノートの「端末」メニューから「SSH ツール」を選択します。

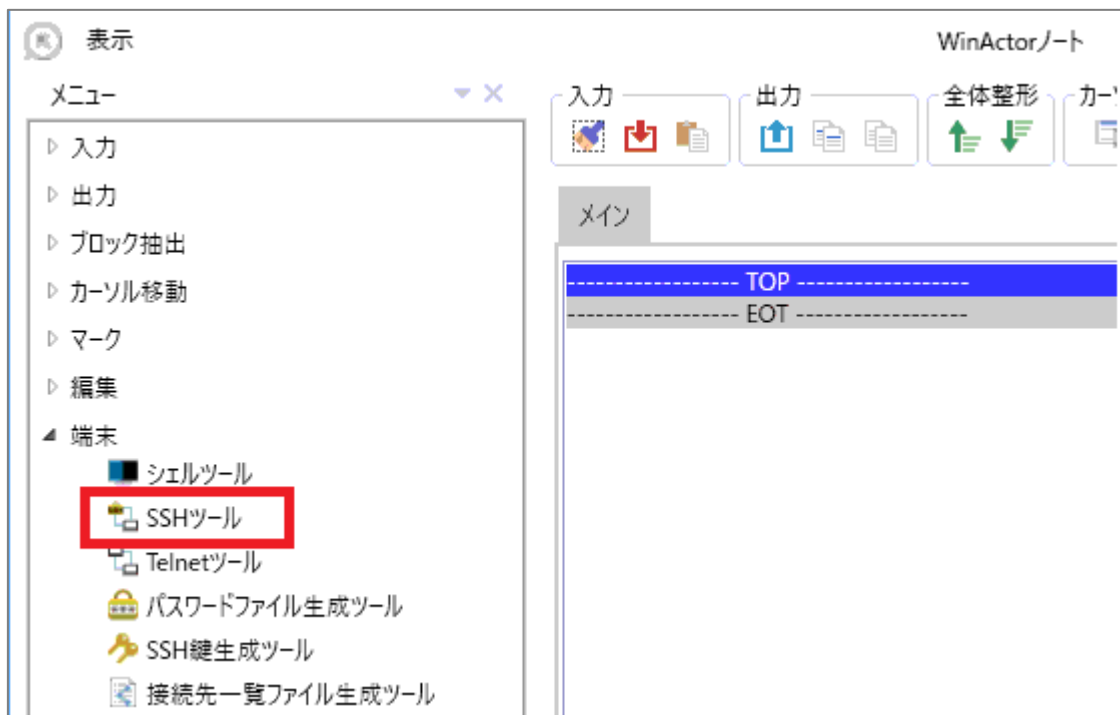


図 2-4 SSH ツールを開く

図 2-5 の赤枠①で囲われた部分のパラメータを設定し、「SSH クライアントを開く」ボタンをクリックします(赤枠②)。パスワードファイルの入力値は「C:¥端末機能シナリオ¥cipherPasswordr.json」となります。

SSHツール

接続設定

文字コードセット UTF-8

送信用改行コード LF (Linux, MacOS X)

ホスト 192.168.56.2

ポート番号 22

ユーザ名 user

パスワードファイル C:¥端末機能シナリオ¥cipherPasswordr.json 参照

コマンド待ち文字列 \$

タイムアウト値[秒] 10

接続先一覧ファイル 参照

パスワード認証

公開鍵認証 参照

ワークフォルダ変更をマクロに記録しない

SSHクライアント ファイル送信(SCP) ファイル受信(SCP)

SSHセッションを開く

セッションを閉じる

コマンド 制御コード Ctrl+A 実行

図 2-5 SSH クライアントを開く

2. コマンドを実行する

図 2-6 の①に下記 A)~D)の[入力するコマンド]に記載されているコマンドを入力したのち、実行ボタン(図 2-6 の②)をクリックします。

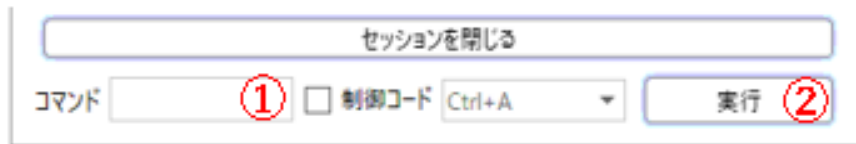


図 2-6 コマンドを実行する

- A) ログフォルダへ移動する
[入力するコマンド]
`cd log`
- B) 日付を取得する
[入力するコマンド]
`date`
- C) 拡張子 log のファイルを表示
[入力するコマンド]
`ls *.log`
- D) C)のコマンドの実行結果を表示
[入力するコマンド]
`echo $?`

[注意事項]

「1. SSH クライアントを開く」から間隔を開けてコマンドを実行した場合には、コマンドの実行が失敗することがあります。その際には、WinActor ノート マクロ編集ウィンドウにて、「作成」にある「新規作成」ボタンをクリックし、再度 2.5.1 章からやり直してください。

3. SSH クライアントを閉じる

図 2-7 の「セッションを閉じる」ボタンをクリックし、SSH クライアントを閉じます。



図 2-7 SSH クライアントを閉じる

表 2-2 の⑦が server#1.log,server#2.log の場合、ここまでの実行結果は、WinActor ノートに図 2-8 のように表示されます。

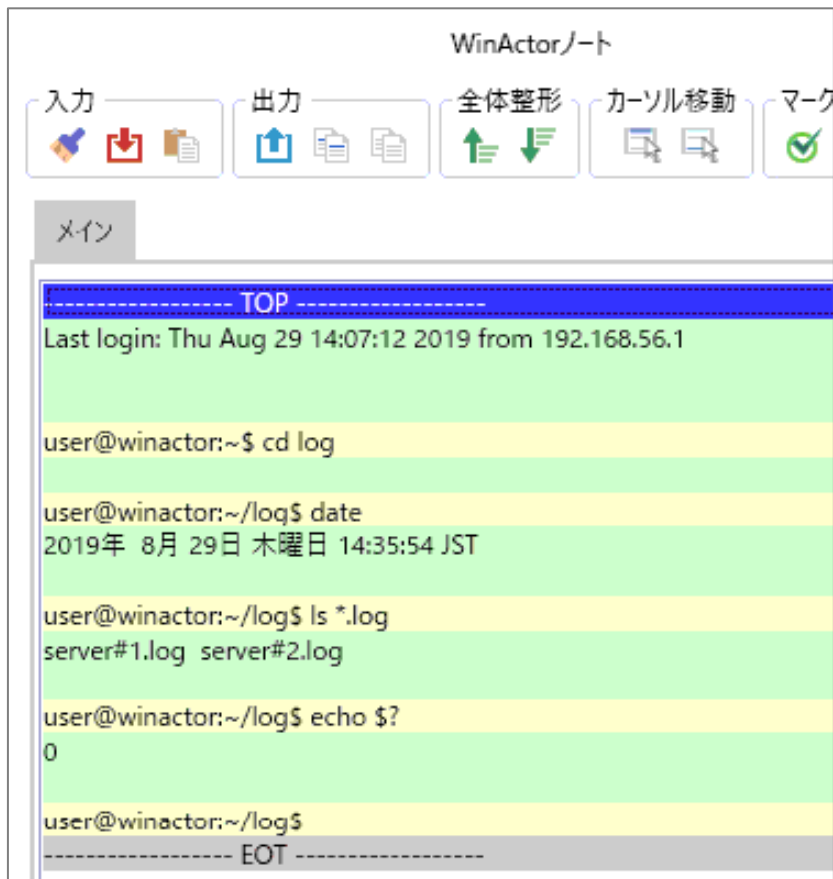


図 2-8 2.5.2 章までの実行結果

2.5.3. テキスト処理

以下の手順にしたがって、拡張子が log のファイル有無の結果を選択します。

1. 「空白除去ツール」(「編集」→「空白除去ツール」)で余分な空白を除去
「対象」として「全ブロック」を選択、「改行を除去」を選択し(図 2-9 ①)、「実行」ボタンをクリック(図 2-9 ②)

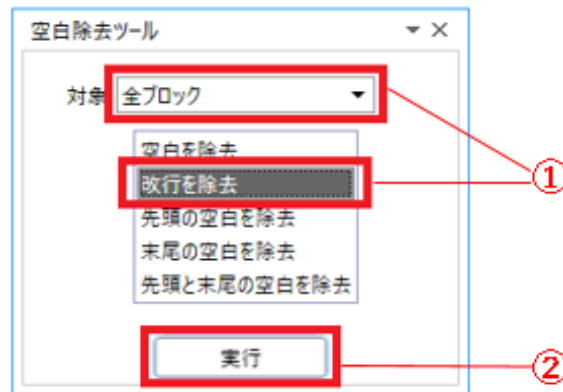


図 2-9 「空白除去ツール」の実行

2. 「ブロック検索ツール」と「次のブロックを選択」で、echo \$?の実行結果を選択
① ブロック検索ツール(「ブロック抽出」→「ブロック検索ツール」)で、「キーワード」に「\$?」を入力、「を含む」を選択、「前方検索(カーソル移動)」を選択し(図 2-10 ①)、「実行」ボタンをクリック(図 2-10 ②)

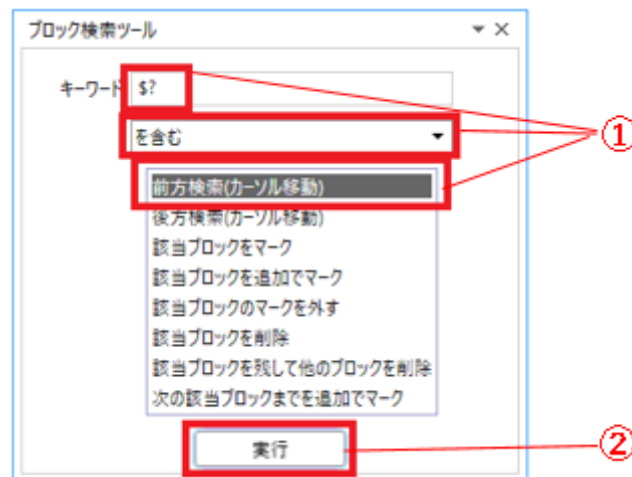


図 2-10 「ブロック検索ツール」による検索

- ② カーソル移動ツール（「カーソル移動」→「カーソル移動ツール」）で、「次のブロックを選択」を選択し、「実行」ボタンをクリック(図 2-11)

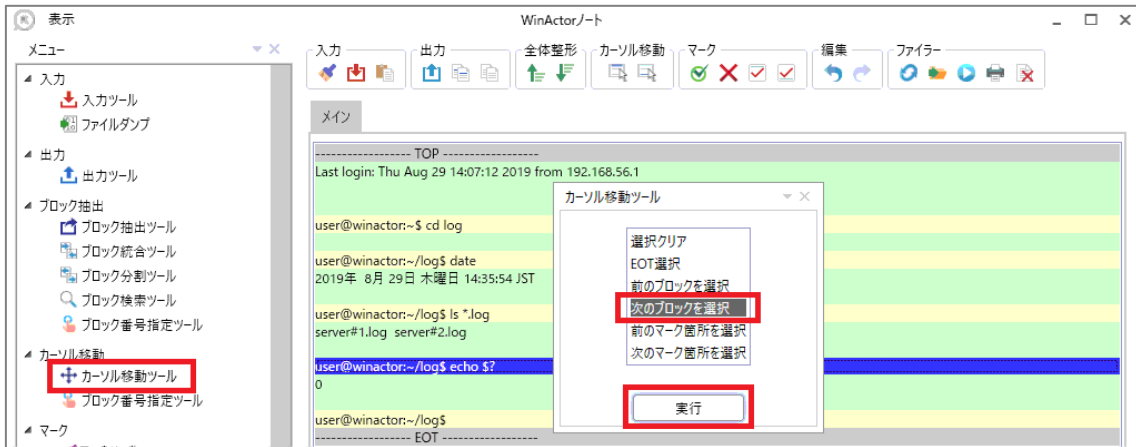


図 2-11 「次のブロックを選択」の実行

ここまでの実行結果は、図 2-12 のように表示されます。

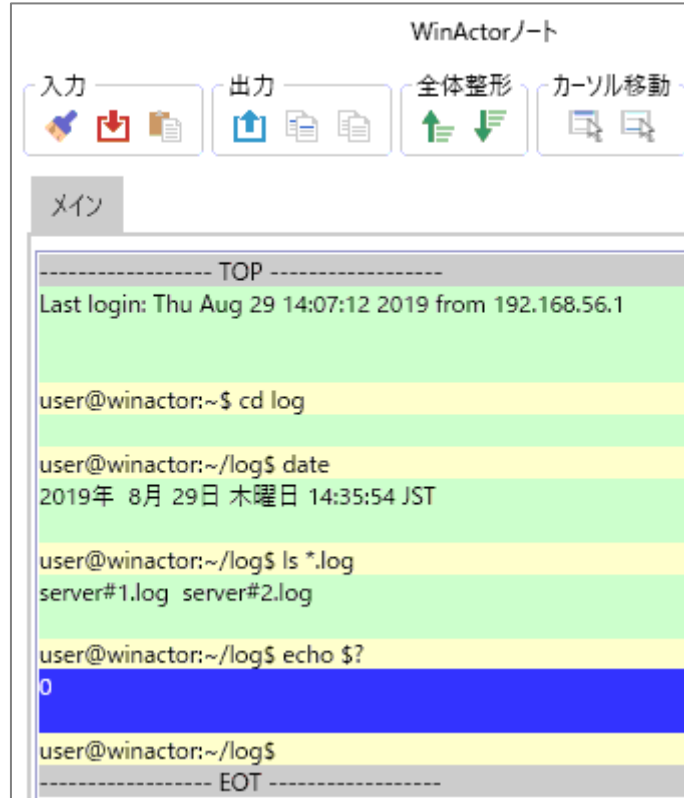


図 2-12 2.5.3 章の実行結果

2.5.4. 画面上のテキストデータのファイル保存

WinActor ノートの画面上のテキストデータをログとして保存します。

「出力ツール」(「出力」→「出力ツール」)を使用し、表 2-3 にしたがって値を設定し、「実行」ボタンをクリックします。

図 2-13 出力ツールの設定画面

表 2-3 出力ツールの設定値

図 2-13 の番号	設定値
①	「全ブロック」を選択
②	「ファイルへ」を選択
③	「C:\¥端末機能シナリオ¥SSH クライアント実行ログ.txt」を入力
④	「MS932(Shift JIS)」を選択
⑤	「CR+LF(Windows)」を選択
⑥	チェックを入れる
⑦	チェックを入れる

2.5.5. マクロの保存

WinActor ノートマクロ編集画面(「表示」→「マクロ編集」で開く)で、マクロ記録を無効にし(図 2-14 ①)、「保存」ボタンをクリックします(図 2-14 ②)。

ファイル名を SSH サーバログ確認.json(図 2-14③)として、C:¥端末機能シナリオのフォルダ(図 2-14 ③)へ現在編集中のマクロを保存します(図 2-14 ④)。

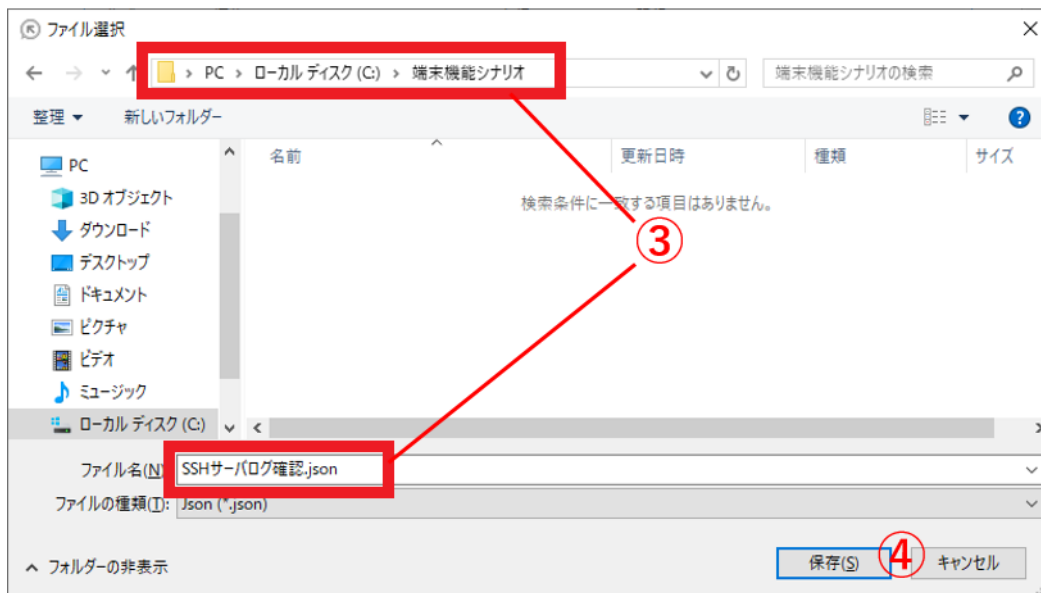
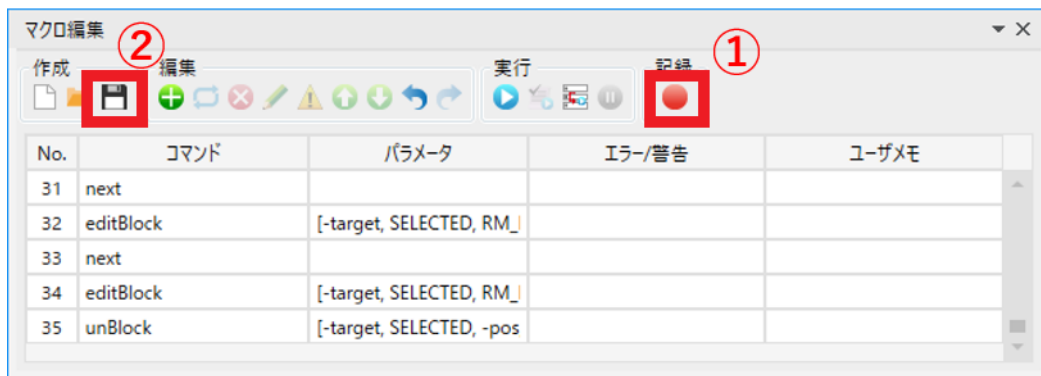


図 2-14 マクロの保存

2.6. WinActor のシナリオ作成

今回作成する WinActor のシナリオを新規シナリオファイルとして保存します。

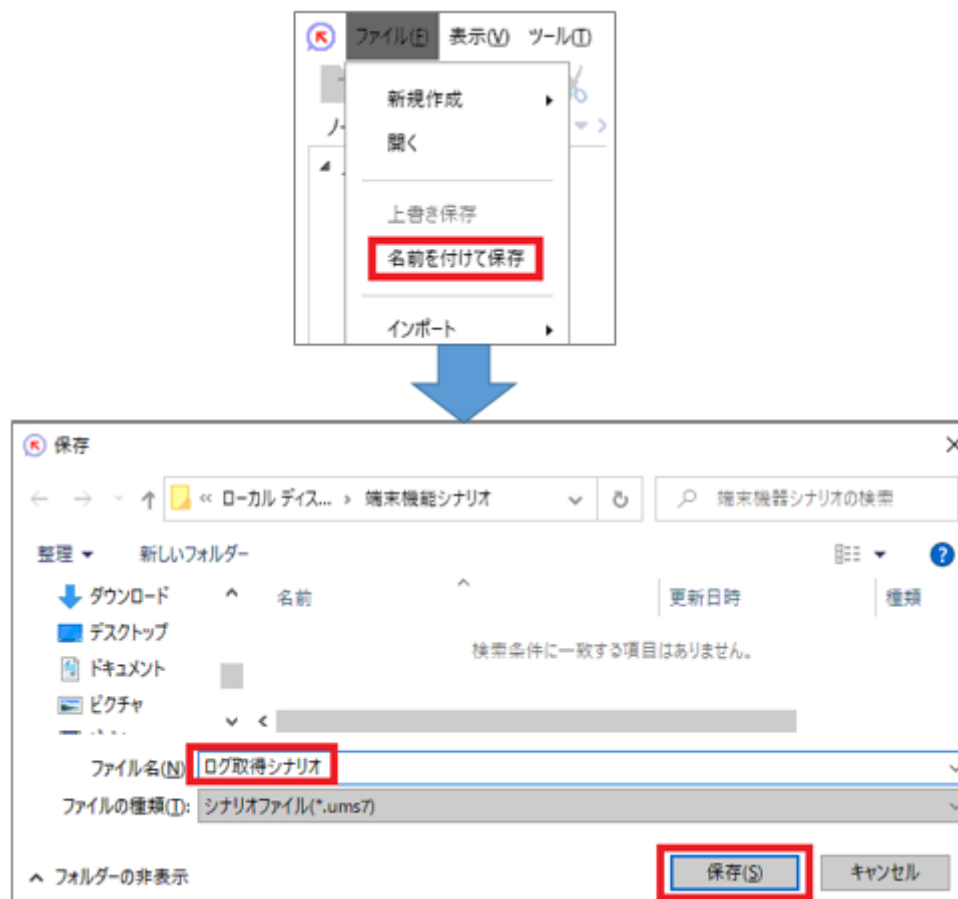


図 2-15 新規シナリオファイルの保存

2.6.1. 作成したマクロの呼び出し

フローチャート画面のシナリオに、ライブラリ「マクロ読み込み実行」をドラッグ&ドロップで配置します。

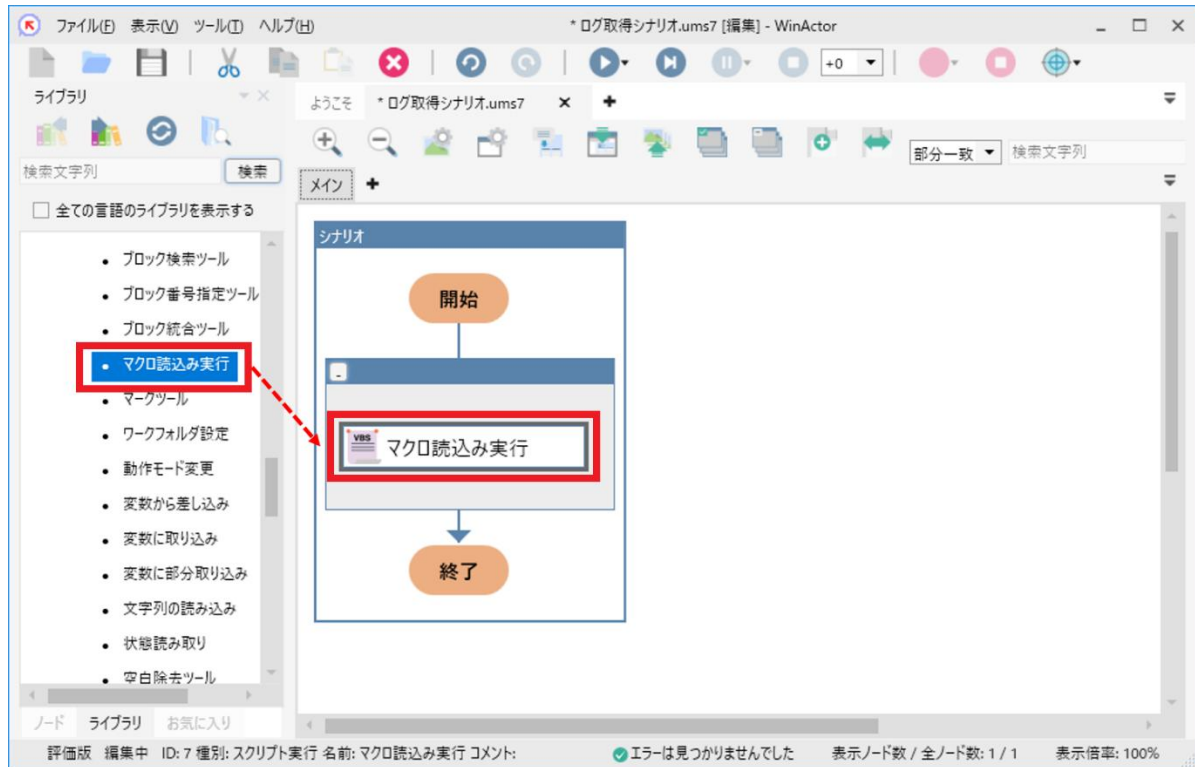


図 2-16 ライブラリ(マクロ読み込み実行)の挿入

配置したライブラリをダブルクリックしてプロパティを開きます。

ファイル名の欄に 2.5.5 章で作成したマクロと同じファイル名 (SSH サーバログ確認.json) を入力し [更新] ボタンをクリックします。

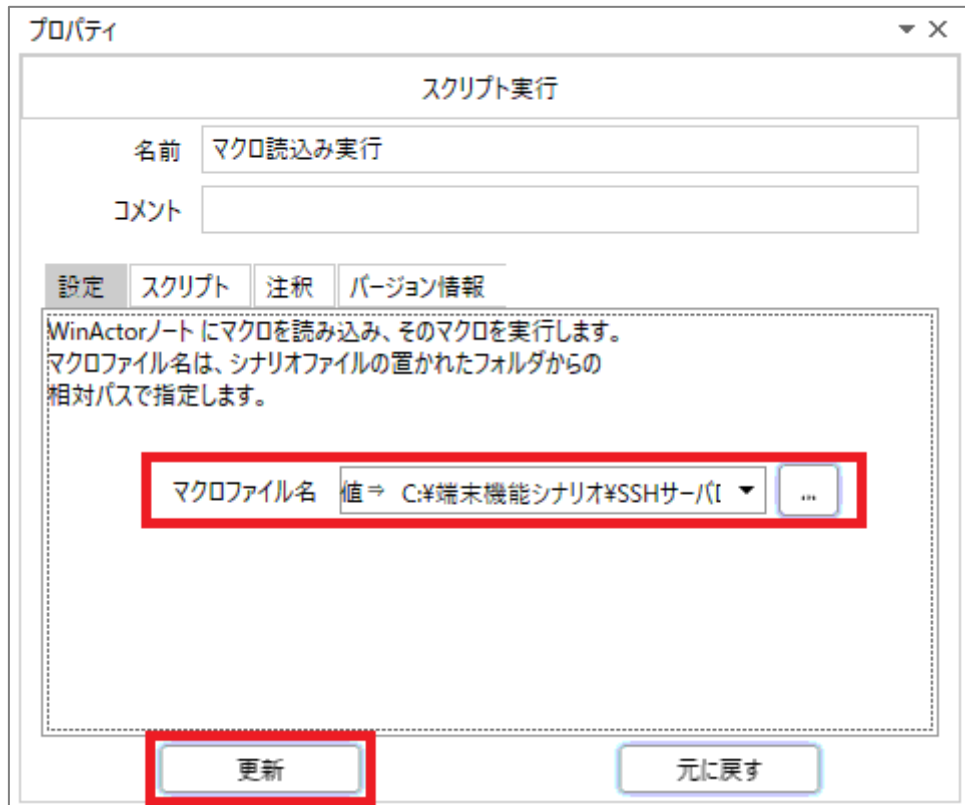


図 2-17 マクロファイル名の指定

マクロ読み込み実行ライブラリのみで正常に動作するかを確認します。

WinActor のツールバーの実行ボタンをクリックすると、WinActor ノート上に想定の内容が出力されることを確認できます。確認後、「C:\¥端末機能シナリオ¥SSH クライアント実行ログ.txt」ファイルを削除してください。

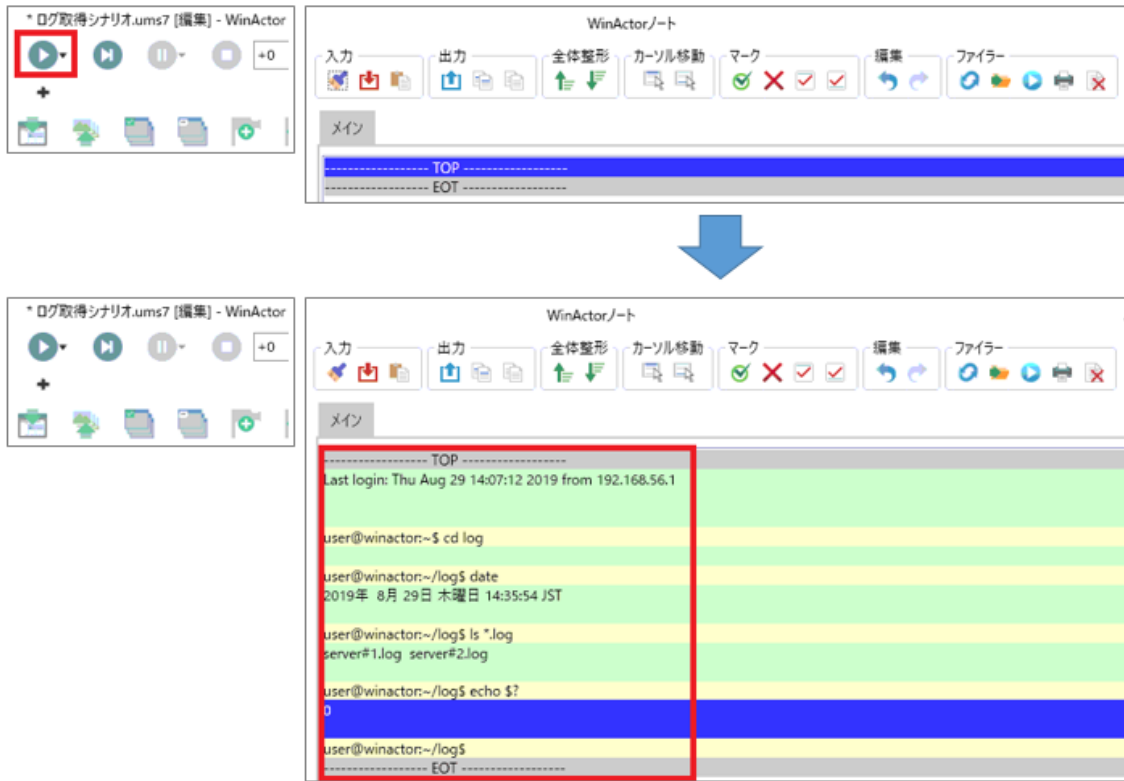


図 2-18 マクロ読み込み実行ライブラリの動作確認

2.6.2. 判定処理の追加

WinActor ノート上のテキストを読み取り、WinActor で変数として保存するための動作を追加します。

ユーザライブラリから「変数に取り込み」をドラッグ&ドロップで配置します

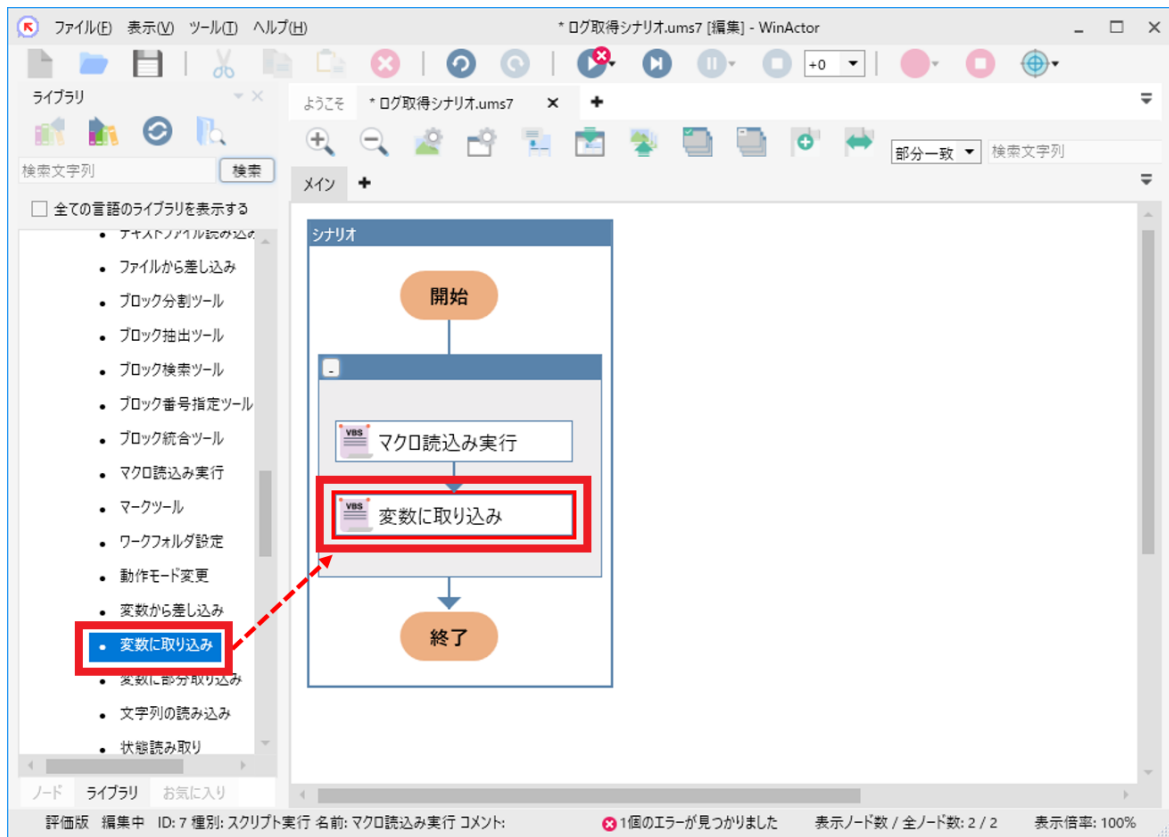


図 2-19 ライブラリ (変数に取り込み) の挿入

ライブラリの配置後、プロパティ画面を開きます。

マクロ読み込み実行ライブラリの実行後「echo \$?」の実行結果が選択され、選択内容を変数として WinActor 上に読み込むため、変数名を「echo 実行結果」と定義し [更新] ボタンをクリックします。

新しい変数として登録するか否かを確認する画面が表示され、[はい] をクリックすると変数一覧画面に新しい変数として登録されます。

The screenshot shows a 'プロパティ' (Properties) dialog box for a 'スクリプト実行' (Script Execution) block. The '名前' (Name) field contains 'echo実行結果'. Below it is a 'コメント' (Comment) field. There are four tabs: '設定' (Settings), 'スクリプト' (Script), '注釈' (Annotations), and 'バージョン情報' (Version Information). The '設定' tab is active, showing instructions: 'WinActorノート 上のテキストを変数に取り込みます。' (Load text from WinActor Note as a variable). Below are four dropdown menus: '対象' (Target) set to '選択されているブロック' (Selected block), '改行' (Line ending) set to '含めない' (Do not include), '改行コード' (Line ending code) set to 'CR+LF (Windows)', and '取得結果' (Result) set to 'echo実行結果'. At the bottom, there are two buttons: '更新' (Update) and '元に戻す' (Reset). The '更新' button is highlighted with a red box.

図 2-20 変数名の入力

変数一覧						
グループ名	変数名	現在値	初期化しない	初期値	マスク	
グループなし	echo実行結果		<input type="checkbox"/>		<input type="checkbox"/>	

図 2-21 変数一覧画面の更新確認

変数「echo 実行結果」に「echo \$?」の実行結果の文字列が読み込まれる動作を確認します。

現在のシナリオの状態ですべての処理が完了した後に変数の値が初期化されてしまうため、末尾に「待機ボックス」を配置します。

ノードから「待機ボックス」をドラッグ&ドロップで配置します

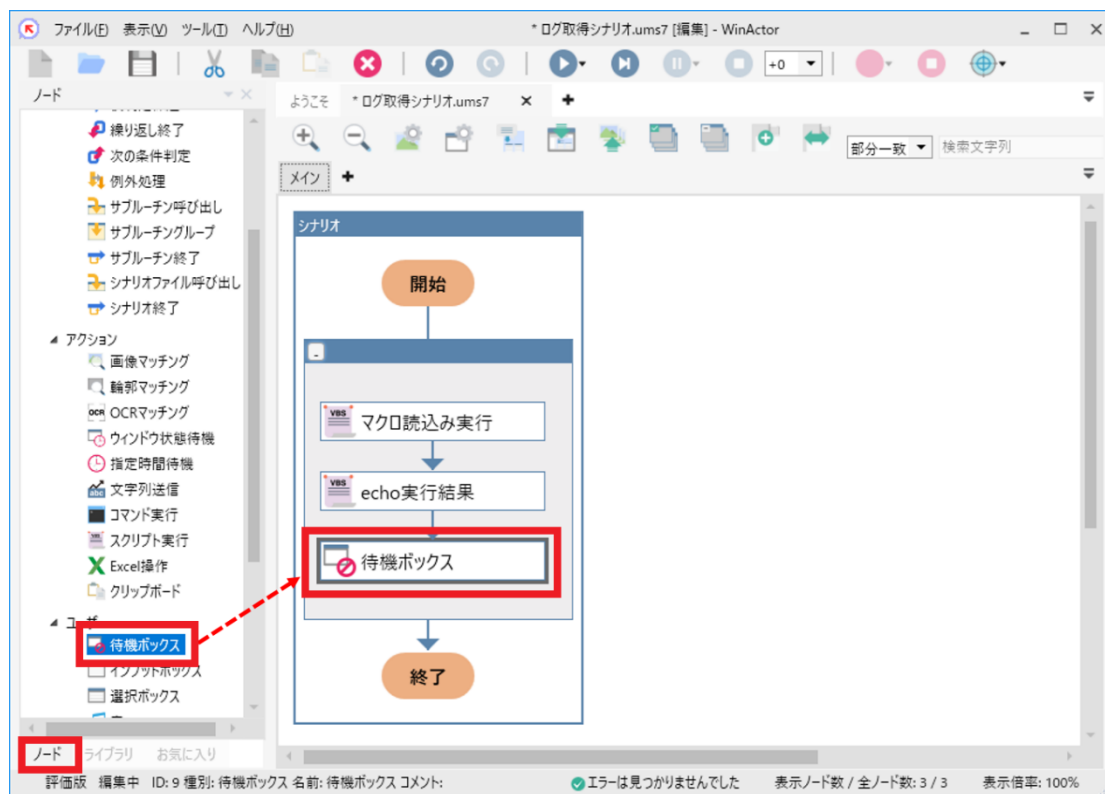


図 2-22 待機ボックスの挿入

現状態のシナリオを実行し、動作を確認します。

WinActor のツールバーの実行ボタンをクリックし、変数一覧画面の「echo 実行結果」に「echo \$?」の実行結果のテキスト（今回の場合は「0」）が入力され現在値が更新されるかを確認します。

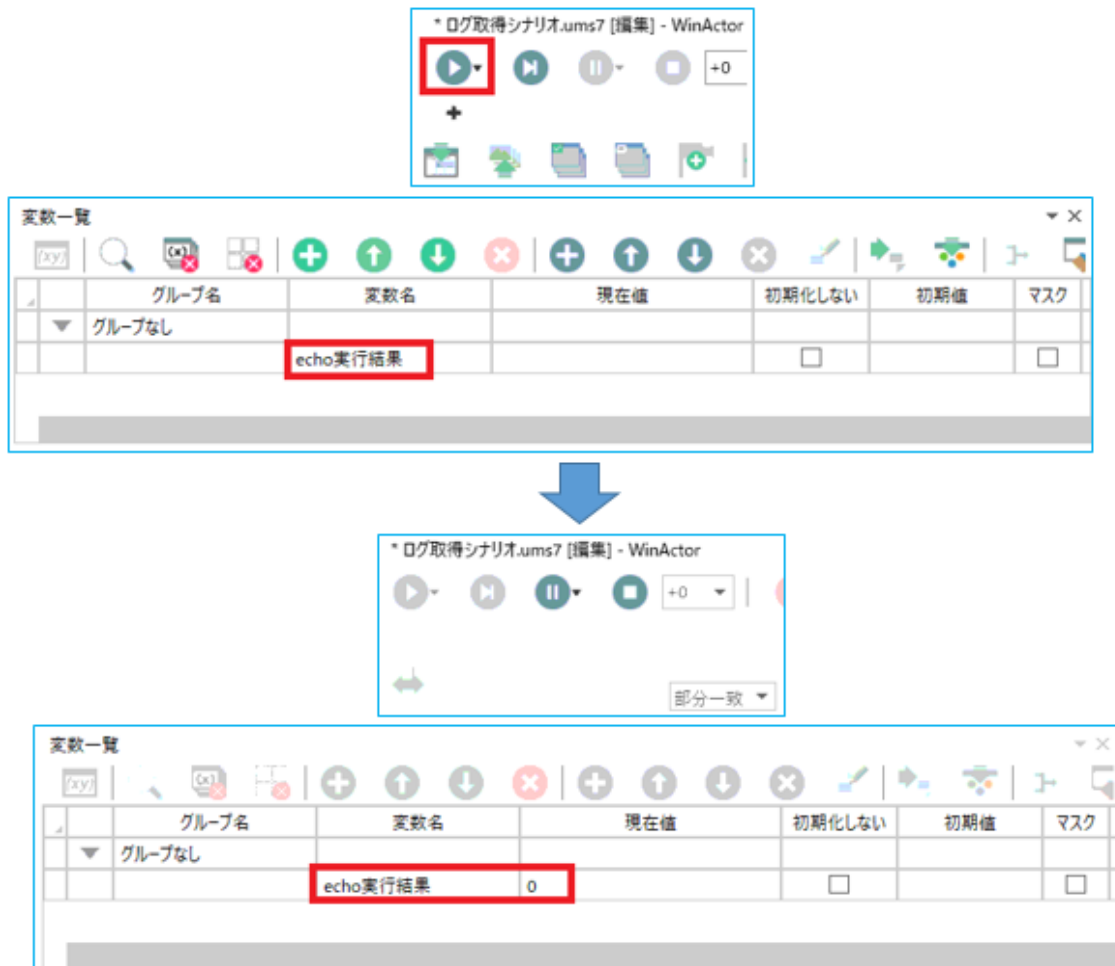


図 2-23 変数に取り込みの動作確認

次に追加した「待機ボックス」を削除し、判定処理を追加します。
ノードから「分岐」をドラッグ&ドロップで配置します。

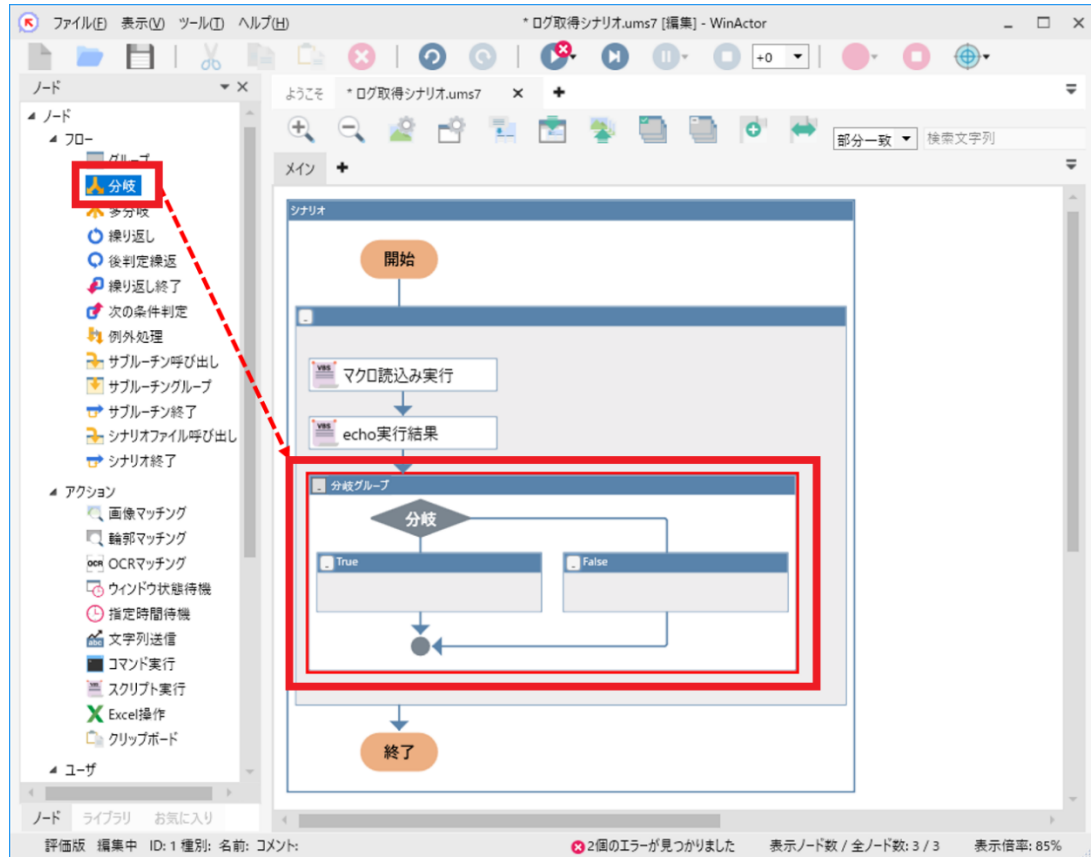


図 2-24 ノード(分岐)の挿入

ノードの配置後、プロパティ画面を開きます。

プロパティの「名前」、「分岐名」を変更したのち、「条件式設定」をクリックし、「条件式」ウィンドウで条件式の左辺は「echo 実行結果」を選択、右辺には 0 を入力します。これにより echo \$? の実行結果が 0 の場合、すなわち拡張子 log のファイルが存在する場合、分岐名「あり」の処理に移動します。

最後に条件式画面、プロパティ画面の [更新] ボタンを順にクリックし、プロパティ設定を完了します。



図 2-25 条件式の設定

2.6.3. ファイル受信処理の追加

拡張子 log のファイルが存在する場合は拡張子 log のファイルを受信する処理を追加します。

ユーザライブラリから「ファイル受信(SCP)」をドラッグ&ドロップで配置します。

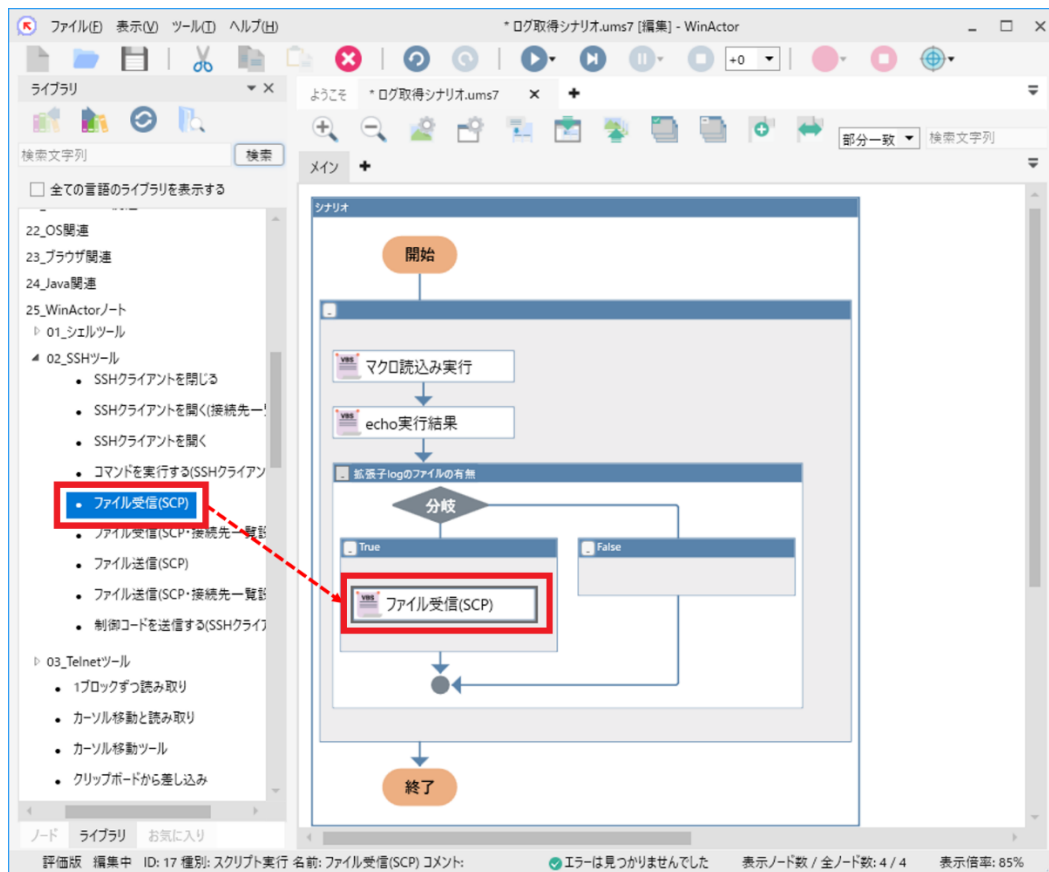


図 2-26 ライブラリ(ファイル受信(SCP))の挿入

配置したライブラリをダブルクリックしてプロパティを開きます。

各項目について、表 2-4 にしたがって入力してください。入力後 [更新] ボタンをクリックします。新しい変数として登録するか否かを確認する画面が表示され、[はい] をクリックすると変数一覧画面に新しい変数として登録されます。

表 2-4 ファイル受信(SCP)のプロパティ設定値

項目	設定値	備考
認証方式	「パスワード認証」を選択	-
ホスト	「ホスト」を入力	*1
ポート番号	「ポート番号」を入力	*1
ユーザ名	「ユーザ名」を入力	*1
パスワードファイル	「パスワードファイル」を入力	*1
秘密鍵ファイル		「値⇒」のままにします
タイムアウト値[秒]	「タイムアウト値」を入力	*1
送信元ファイル	「送信元ファイル」を入力	*1
送信先パス	「日付」を入力	*1

1 一度「」を選択してから値を入力します。

プロパティ

スクリプト実行

名前

コメント

設定 スクリプト 注釈 バージョン情報

SCPによるファイル受信を行います。
パスワードファイル、秘密鍵ファイル、送信先パスは、
シナリオファイルの置かれたフォルダからの相対パスで指定します。

認証方式	<input type="text" value="パスワード認証"/>
ホスト	<input type="text" value="ホスト"/>
ポート番号	<input type="text" value="ポート番号"/>
ユーザ名	<input type="text" value="ユーザ名"/>
パスワードファイル	<input type="text" value="パスワードファイル"/>
秘密鍵ファイル	<input type="text" value="値"/>
タイムアウト値[秒]	<input type="text" value="タイムアウト値"/>
送信元ファイル	<input type="text" value="送信元ファイル"/>
送信先パス	<input type="text" value="日付"/>

図 2-27 プロパティ設定後

変数一覧画面の各変数に対する初期値について、表 2-5 にしたがって入力してください。

表 2-5 変数一覧画面の初期値設定

変数名	初期値	備考
ホスト	192.168.56.2	-
ポート番号	22	-
ユーザ名	user	-
パスワードファイル	cipherPassword.json	-
タイムアウト値	10	ご使用になる環境に応じて適切な値を調整してください。
送信元ファイル	/home/user/log/*.log	-
日付		初期値は空欄にしてください。値は 2.6.4 章の処理で自動的に生成されます。



変数一覧

グループ名	変数名	現在値	初期化しない	初期値
グループなし	echo実行結果		<input type="checkbox"/>	
	ホスト		<input type="checkbox"/>	192.168.56.2
	ポート番号		<input type="checkbox"/>	22
	ユーザ名		<input type="checkbox"/>	user
	パスワードファイル		<input type="checkbox"/>	cipherPassword.json
	タイムアウト値		<input type="checkbox"/>	10
	送信元ファイル		<input type="checkbox"/>	/home/user/log/*.log
	日付		<input type="checkbox"/>	

図 2-28 初期値設定後

2.6.4. ログ保存用フォルダの作成処理の追加

① サブルーチン、サブルーチン呼び出し処理の作成

SSH サーバ上のログを格納するための日時に応じたフォルダを作成する処理を追加します。ログフォルダ作成については、サブルーチン内で処理を行います。

ノードからサブルーチン呼び出し、サブルーチングループをドラッグ&ドロップで配置します。

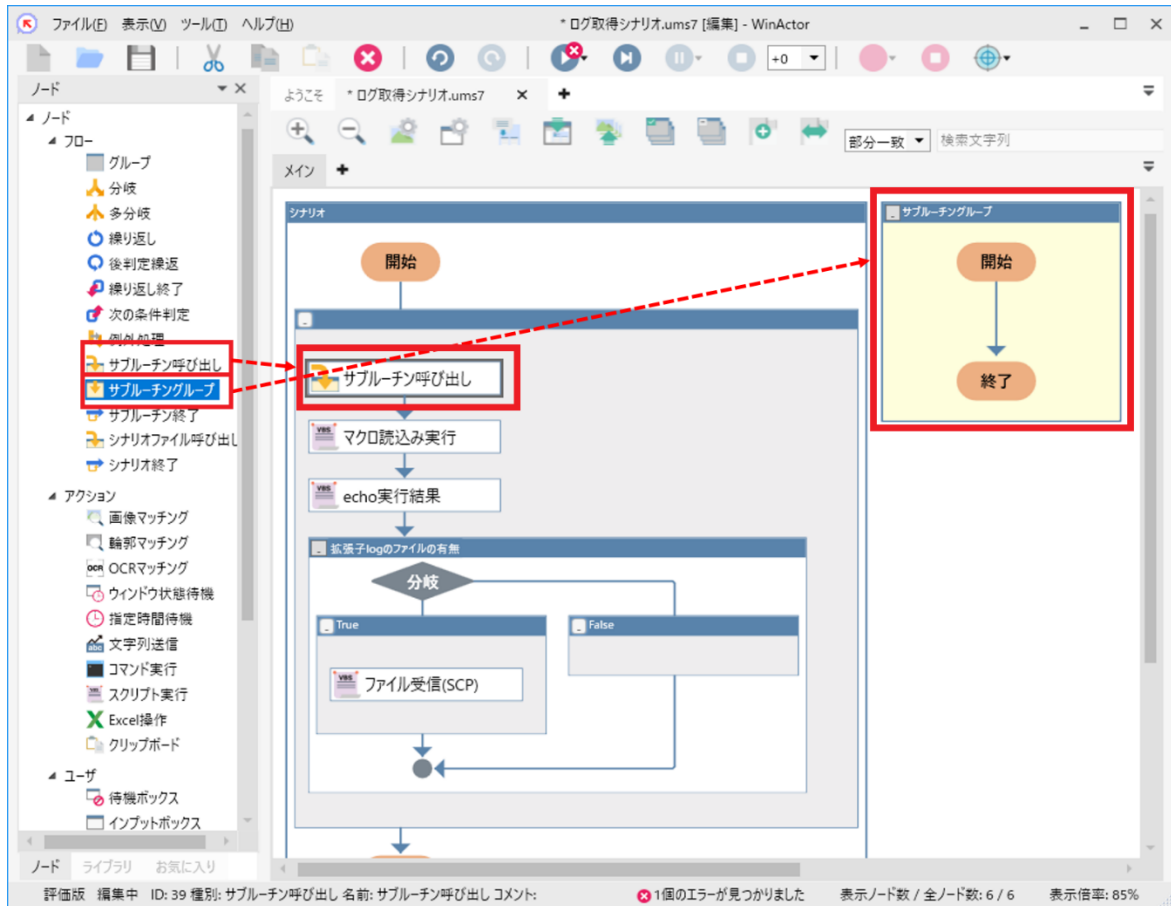


図 2-29 ノード(サブルーチン呼び出し、サブルーチングループ)の挿入

ノードを配置後、それぞれのプロパティ画面を開きます。

サブルーチングループのプロパティ画面については、名前を「ログフォルダ作成」に変更します。サブルーチン呼び出しについては名前を「ログフォルダ作成呼び出し」に変更し、サブルーチン名は「ログフォルダ作成」を選択します。



図 2-30 サブルーチングループ、サブルーチン呼び出しの設定

② サブルーチン内の処理の作成

「ログフォルダ作成」サブルーチン内の処理を作成します。
全体像を以下に示します。

1. フォルダ名の作成

① 日時の取得

取得される日時の形式としては、以下のようになります。

<取得される日時の形式>

yyyy-mm-dd HH:mm:ss

ただし Windows の地域の設定によっては、上記のような形式ではない場合があります。その場合は本マニュアルを参考にシナリオの処理内容を変更してください。

② フォルダ名の生成

①の形式から半角コロン(:)をなくし、スペースをアンダーラインに置き換えることで表 2-2 の④の形式(yyyy-mm-dd_HHmms)に変換します

2. フォルダの作成

上記 1. ②で生成されたフォルダ名のフォルダを作成します。

個々の処理について以降に示します。

1. ① 日時の取得

ノードから日時取得をドラッグ&ドロップで配置します。

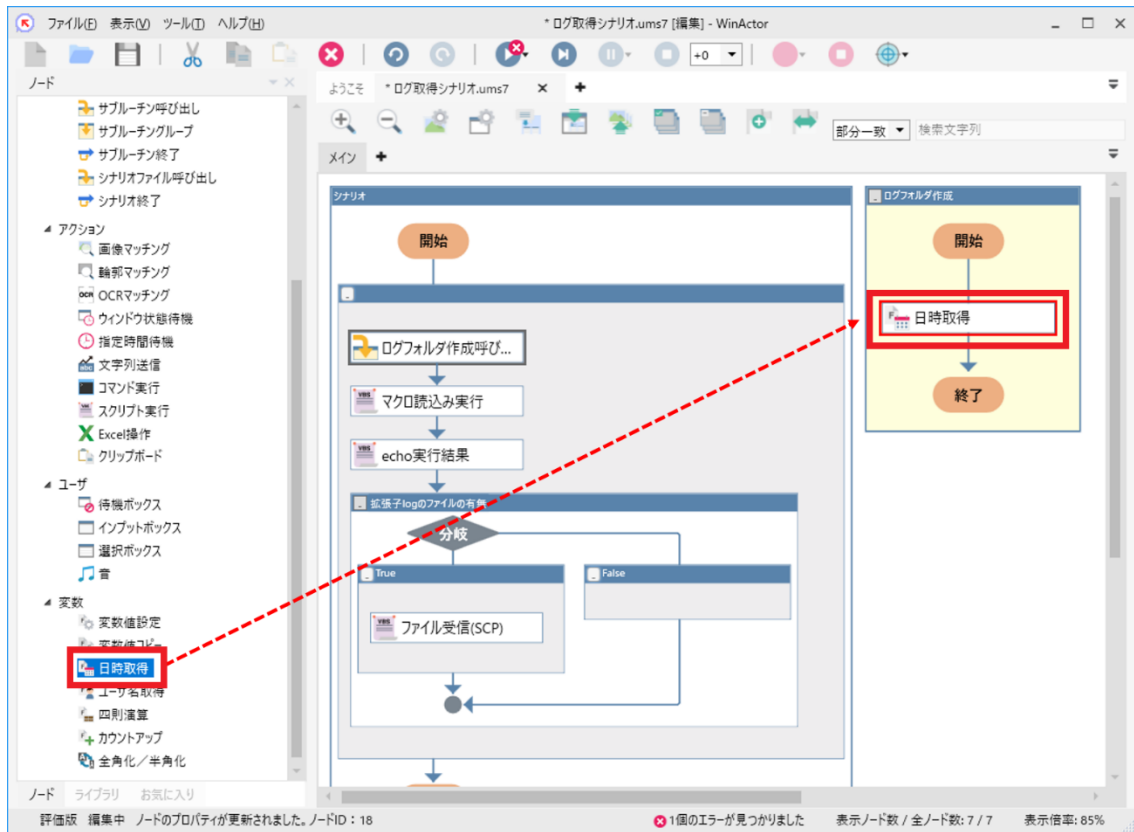


図 2-31 ノード(日時取得)の挿入

配置したノードをダブルクリックしてプロパティを開きます。

取得結果に「日付」を、日付形式に「yyyy-mm-dd」を、タイムゾーンに「OS デフォルト」を選択し [更新] ボタンをクリックします。

プロパティ

日時取得

名前 日時取得

コメント

取得結果 日付

フォーマットタイプ 日付と時間

日付形式 yyyy-mm-dd

タイムゾーン OSデフォルト

更新 元に戻す

図 2-32 プロパティ(日時取得)の設定

1. ② フォルダ名の生成

まず 2 つの文字列置換ライブラリを続けて追加します。変換文字列は双方とも「日付」を選択し、置換前はそれぞれ、半角コロン(:)と空欄、半角スペース()とアンダーライン(_)を選択します。

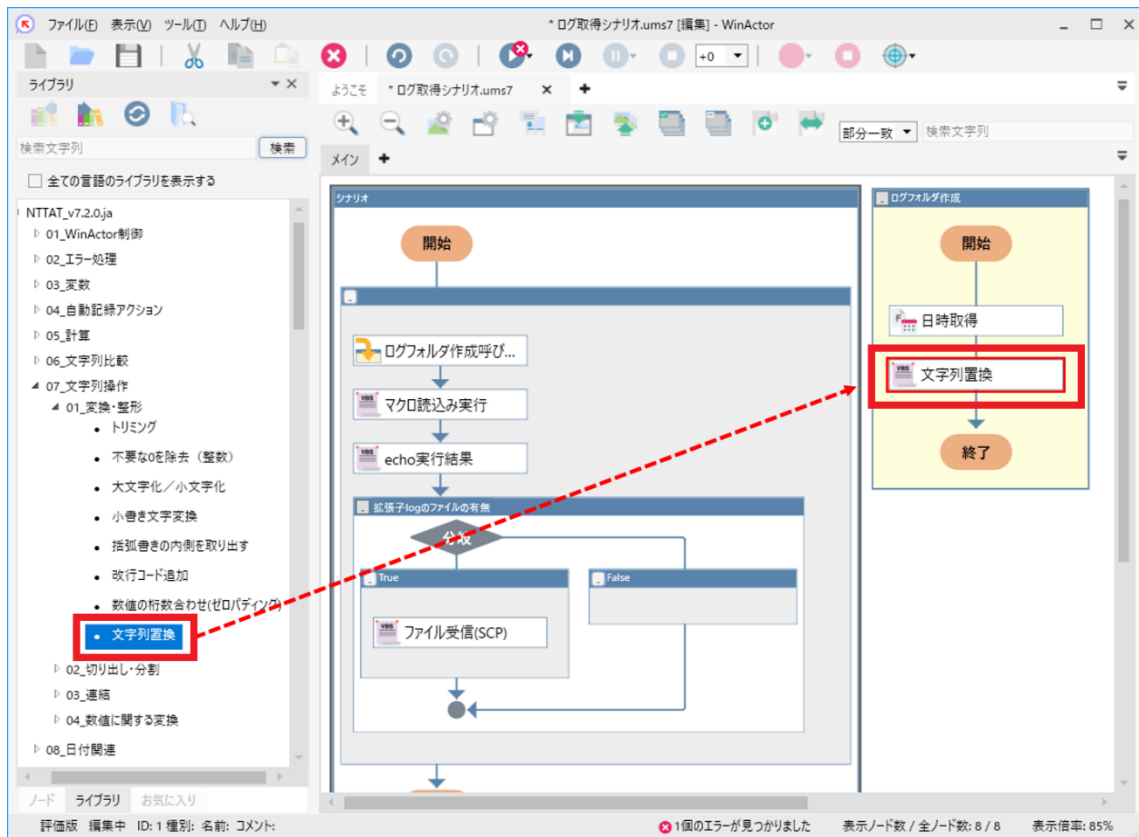


図 2-33 一つ目のライブラリ(文字列置換)の挿入

プロパティ

スクリプト実行

名前 文字列置換

コメント

設定 スクリプト 注釈 バージョン情報

「変換文字列」に指定した文字列から「置換前」の文字列を検索して、「置換後」の文字列に置き換えます。

注意：
・置換前の文字列が見つからない場合、元の文字列のままとなります。
・変換文字列から特定の文字列を消去していきたい場合は、置換前に消去したい文字、置換後は空で指定します。

変換文字列 日付

置換前 値⇒:

置換後 値⇒

更新 元に戻す

図 2-34 プロパティ(一つ目の文字列置換: 半角コロン(:)と空欄)の設定

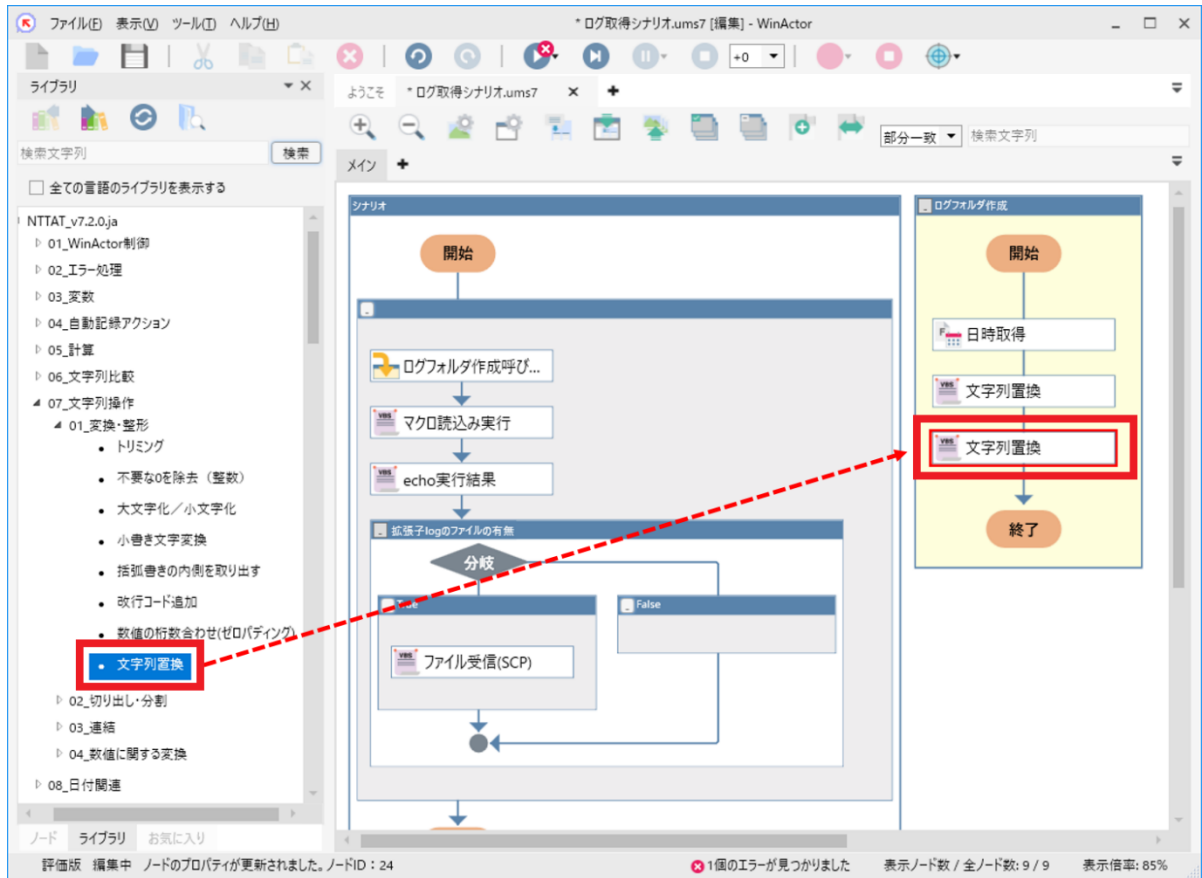


図 2-35 二つ目のライブラリ(文字列置換)の挿入

図 2-36 プロパティ(二つ目の文字列置換:半角スペース()とアンダーライン(_))の設定

次にシナリオフォルダのファイルパスライブラリを追加します。

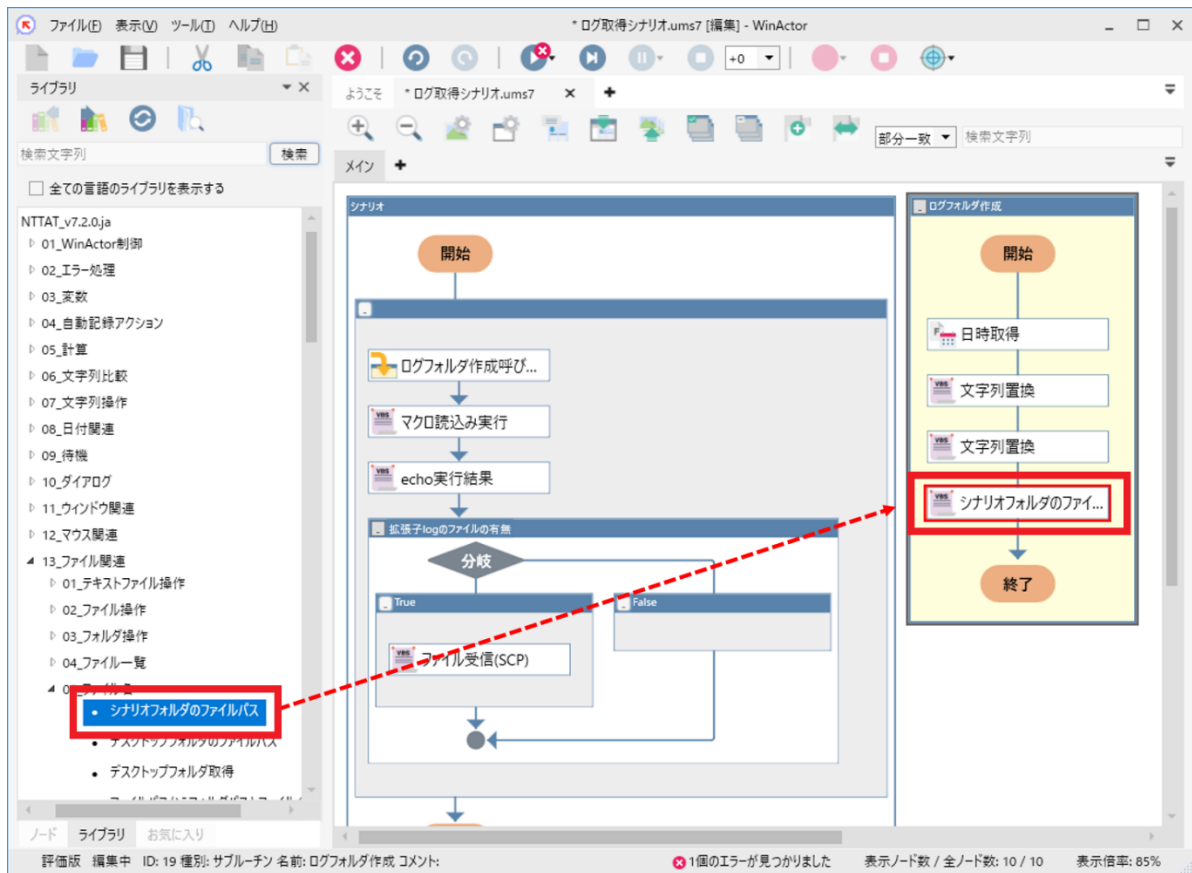


図 2-37 ライブラリ(シナリオフォルダのファイルパス)の挿入

配置したライブラリをダブルクリックしてプロパティを開きます。

ファイル名は「日付」を選択し、ファイルパス生成結果は「結果フォルダ」を入力し [更新] ボタンをクリックします。

新しい変数として登録するか否かを確認する画面が表示され、[はい] をクリックすると変数一覧画面に新しい変数として登録されます。

プロパティ

スクリプト実行

名前 シナリオフォルダのファイルパス

コメント

設定 スクリプト 注釈 バージョン情報

シナリオフォルダを起点としたファイルパスを作成します
シナリオが保存されていない場合は、空の情報となります。

例) シナリオが c:%temp フォルダに保存されており、
ファイル名に a.txt を指定した場合、
ファイルパス生成結果には c:%temp%a.txt が格納されます。

ファイル名 日付

ファイルパス生成結果 結果フォルダ

更新 元に戻す

図 2-38 プロパティ(シナリオフォルダのファイルパス)の設定

変数一覧

グループ名	変数名	現在値	初期化しない	初期値
グループなし	echo実行結果		<input type="checkbox"/>	
	ホスト		<input type="checkbox"/>	192.168.56.2
	ポート番号		<input type="checkbox"/>	22
	ユーザ名		<input type="checkbox"/>	user
	パスワードファイル		<input type="checkbox"/>	cipherPassword.json
	タイムアウト値		<input type="checkbox"/>	10
	送信元ファイル		<input type="checkbox"/>	/home/user/log/*.log
	日付		<input type="checkbox"/>	
	結果フォルダ		<input type="checkbox"/>	

図 2-39 変数一覧画面の更新確認

2. フォルダの作成

ここまでに作成したログファイルを保存するフォルダ名(変数「結果フォルダ」)をもとにフォルダを作成します。

ライブラリからフォルダ作成をドラッグ&ドロップで配置します。

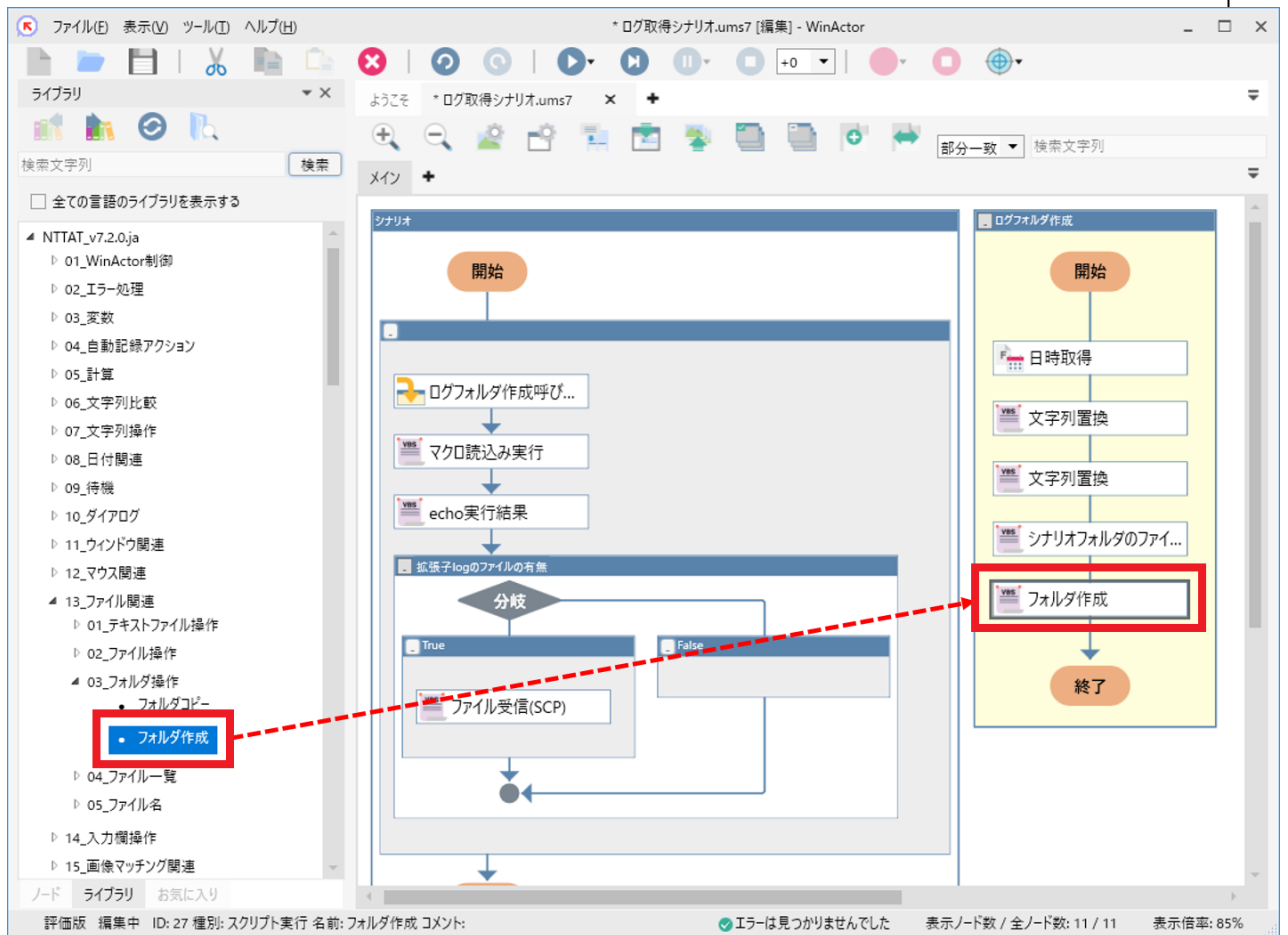


図 2-40 ライブラリ(フォルダ作成)の挿入

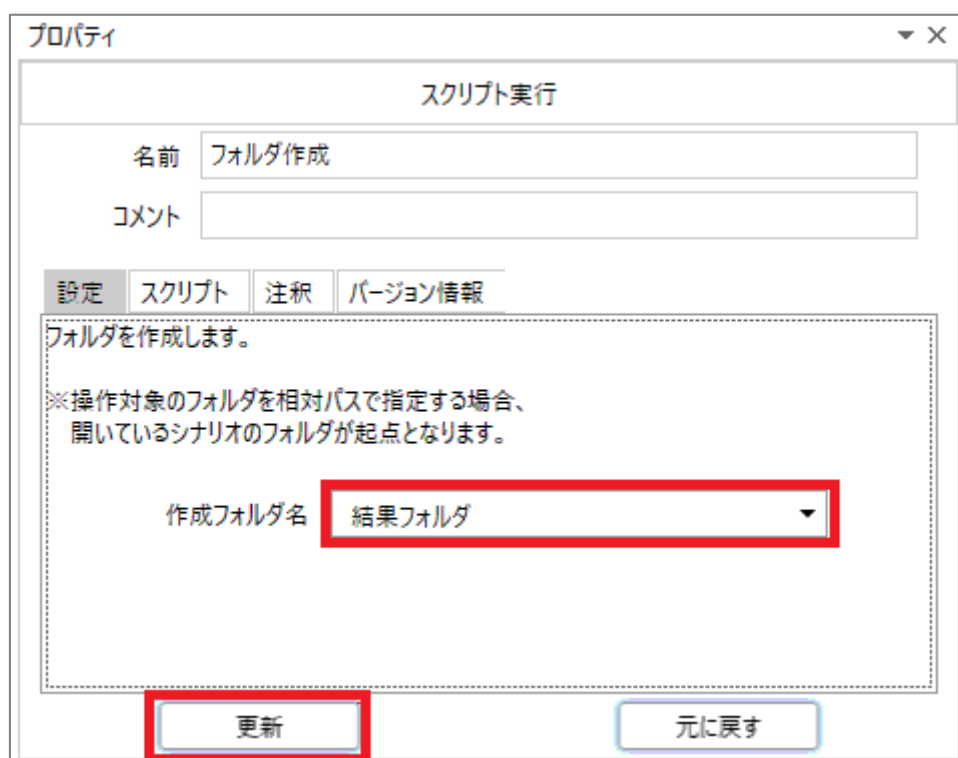


図 2-41 プロパティ(フォルダ作成)の設定

現状のシナリオの部分実行を行い、動作を確認します。

ログフォルダ作成を左クリックで選択後、右クリックし「部分実行」を選択します。

WinActor ノート 端末機能 シナリオ作成マニュアル

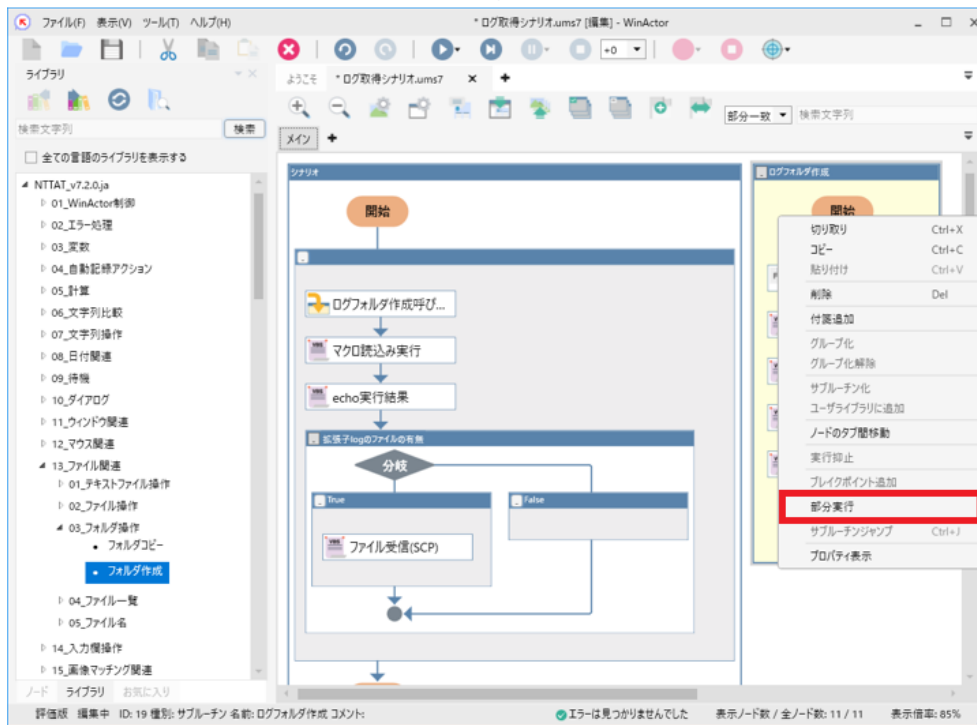
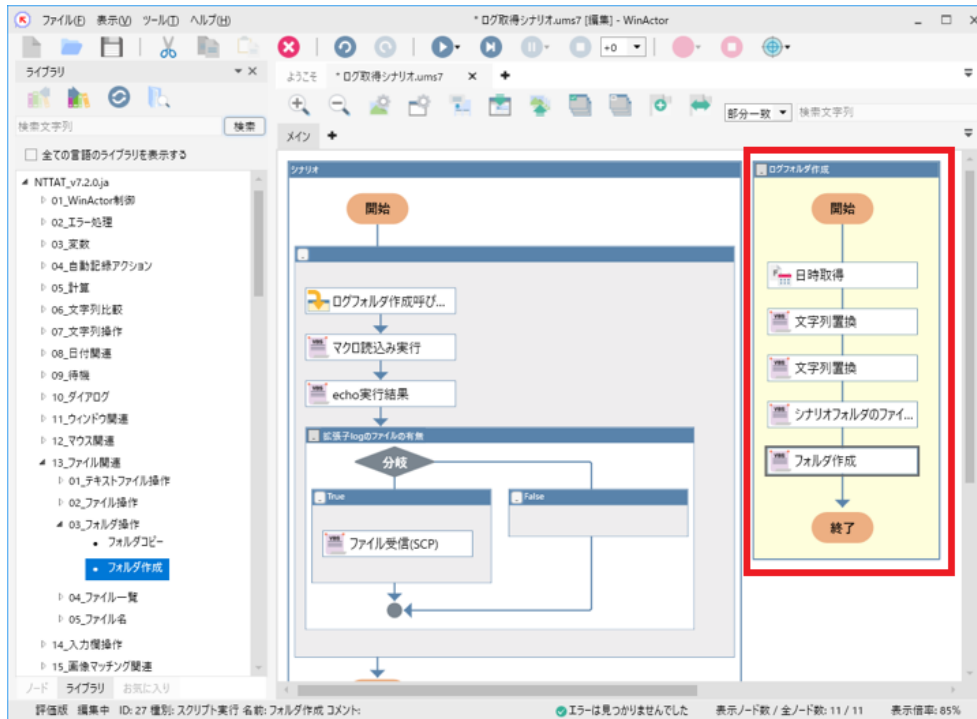


図 2-42 ログフォルダ作成の部分実行の実施

部分実行後に、エクスプローラーで C:\¥端末機能シナリオを開き、表 2-2 の④の形式で

空のフォルダが作成されたことを確認します。確認後、作成されたフォルダを削除してください。

2.6.5. ログファイルの移動処理の追加

① サブルーチン、サブルーチン呼び出し処理の作成

WinActor ノートの表示内容を保存したファイル(SSH クライアント実行ログ.txt)をログフォルダへ移動する処理を追加します。

ノードからサブルーチン呼び出し、サブルーチングループをドラッグ&ドロップで配置します。

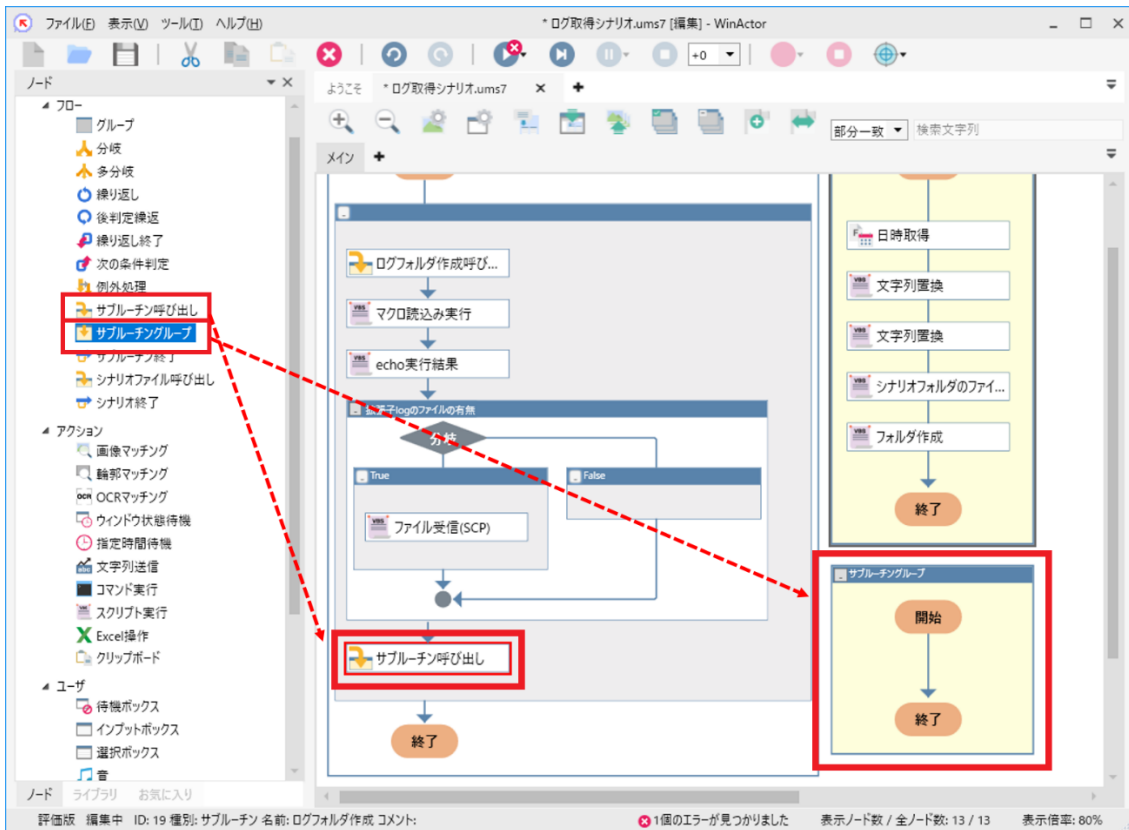


図 2-43 ノード(サブルーチン呼び出し、サブルーチングループ)の挿入

ノードの配置後、それぞれのプロパティ画面を開きます。

サブルーチングループのプロパティ画面については、名前を「ログファイル移動」に変更します。サブルーチン呼び出しについては名前を「ログファイル移動呼び出し」に変更し、サブルーチン名は「ログファイル移動」を選択します。

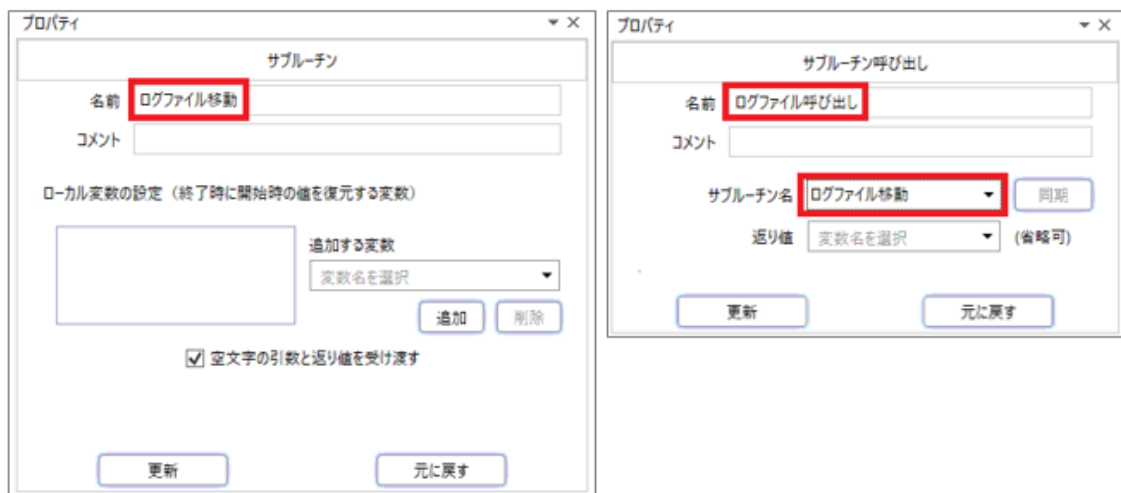


図 2-44 サブルーチングループ、サブルーチン呼び出しの設定

② サブルーチン内の処理の作成

「ログファイル移動」サブルーチン内の処理を作成します。

1. シナリオフォルダのファイルパスライブラリの追加

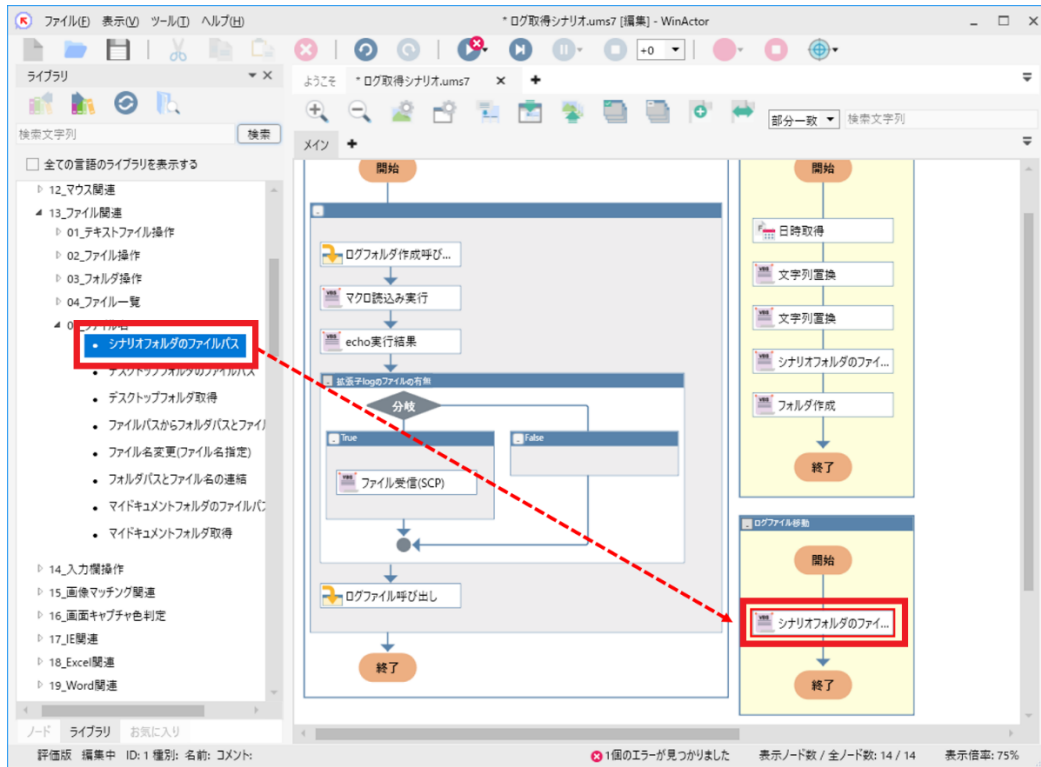


図 2-45 ライブラリ(シナリオフォルダのファイルパス)の挿入

配置したライブラリをダブルクリックしてプロパティを開きます。

ファイル名は「SSH クライアント実行ログ.txt」を入力し、ファイルパス生成結果は「SSH ログファイル名_絶対パス」を入力し [更新] ボタンをクリックします。

新しい変数として登録するか否かを確認する画面が表示され [はい] をクリックすると変数一覧画面に新しい変数として登録されます。

プロパティ

スクリプト実行

名前 シナリオフォルダのファイルパス

コメント

設定 スクリプト 注釈 バージョン情報

シナリオフォルダを起点としたファイルパスを作成します
シナリオが保存されていない場合は、空の情報となります。

例) シナリオが c:%temp フォルダに保存されており、
ファイル名に a.txt を指定した場合、
ファイルパス生成結果には c:%temp%a.txt が格納されます。

ファイル名 値⇒ SSHクライアント実行ログ.txt

ファイルパス生成結果 SSHログファイル名_絶対パス

更新 元に戻す

図 2-46 プロパティ(シナリオフォルダのファイルパス)の設定

変数一覧

グループ名	変数名	現在値	初期化しない	初期値
グループなし				
	echo実行結果		<input type="checkbox"/>	
	ホスト		<input type="checkbox"/>	192.168.56.2
	ポート番号		<input type="checkbox"/>	22
	ユーザ名		<input type="checkbox"/>	user
	パスワードファイル		<input type="checkbox"/>	cipherPassword.json
	タイムアウト値		<input type="checkbox"/>	10
	送信元ファイル		<input type="checkbox"/>	/home/user/log/*.log
	日付		<input type="checkbox"/>	
	結果フォルダ		<input type="checkbox"/>	
	SSHログファイル名_絶対パス		<input type="checkbox"/>	

図 2-47 変数一覧画面の更新確認

2. ファイル移動ライブラリの追加

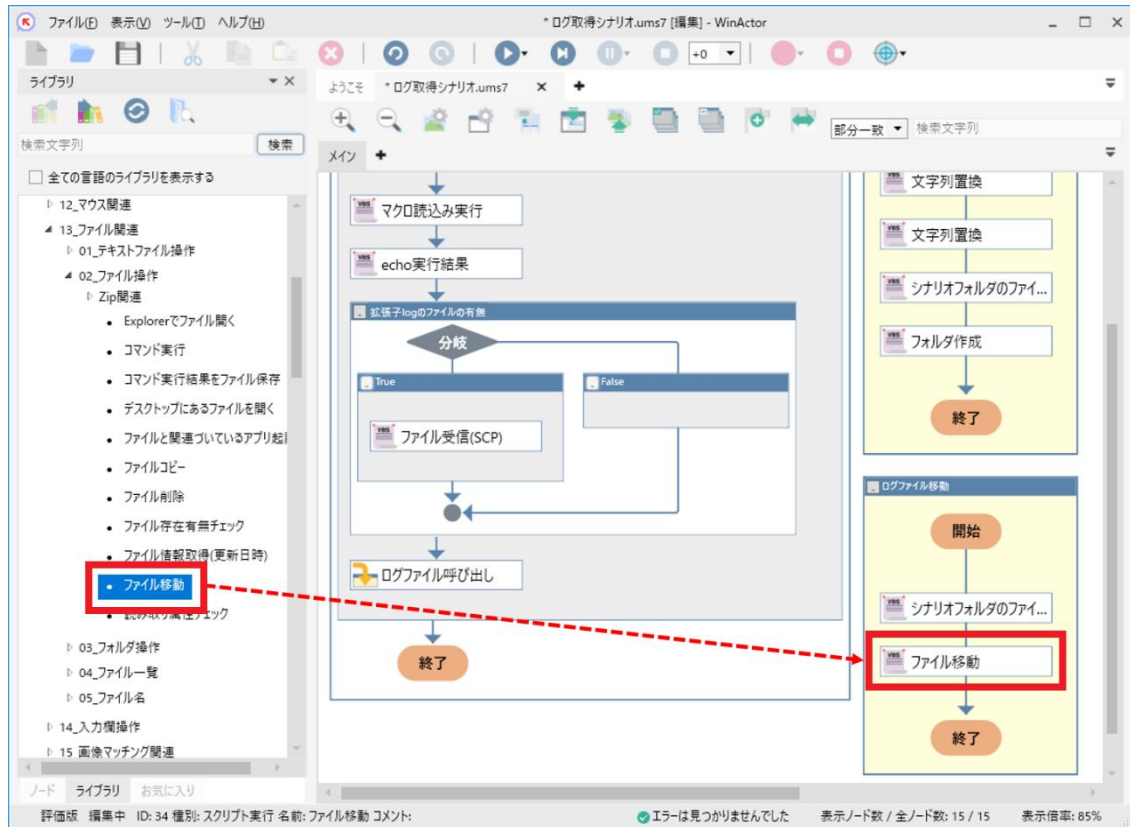


図 2-48 ライブラリ(ファイル移動)の挿入

配置したライブラリをダブルクリックしてプロパティを開きます。

移動元は「SSH ログファイル名_絶対パス」を選択し、移動先は「結果フォルダ」を入力し [更新] ボタンをクリックします。

プロパティ

スクリプト実行

名前 ファイル移動

コメント

設定 スクリプト 注釈 バージョン情報

指定されたファイルを移動します。

「移動元」：移動したいファイル名を絶対パスか相対パスで指定します。

「移動先」：移動先のフォルダ名もしくはファイル名を、絶対パスか相対パスで指定します。
「移動先」にフォルダ名を指定した場合は、ファイル名を変更せずに指定フォルダへファイルが移動します。
移動先フォルダでファイル名を変更したい場合は、「移動先」にファイル名まで含めてください。

※操作対象のファイルまたはフォルダを相対パスで指定する場合、開いているシナリオのフォルダが起点となります。

移動元 SSHログファイル名_絶対パス

移動先 結果フォルダ

更新 元に戻す

図 2-49 プロパティ(ファイル移動)の設定

ここまで修正したシナリオを上書き保存します。

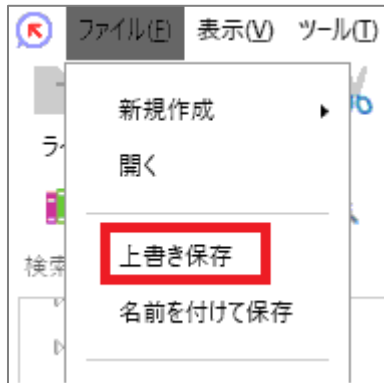


図 2-50 シナリオの上書き保存

2.7. 動作確認

シナリオの最終動作を確認します。

WinActor のツールバーの実行ボタンをクリックします。



図 2-51 シナリオの実行

シナリオ実行後、「C:\¥端末機能シナリオ」フォルダにシナリオを実行した日時のフォルダが作成され、server#1.log, server#2.log, SSH クライアント実行ログ.txt の 3 ファイルが取得されます。

SSH クライアント実行ログ.txt の取得例を図 2-52 に示します。

```
Last login: Thu Aug 29 14:07:12 2019 from 192.168.56.1
[user@demoserver ~]$ cd log

[user@demoserver log]$ date
2019年 8月 29日 木曜日 14:35:54 JST
[user@demoserver log]$ ls *.log
server#1.log server#2.log
[user@demoserver log]$ echo $?
0
[user@demoserver log]$
```

図 2-52 取得された実行ログ例

3. ライブラリ・プロパティの一覧

3.1. シェルツール

PowerShell、およびコマンドプロンプトを使用するためのライブラリについて本章で説明します。本章では基本的なシナリオの作成手順を説明します。3.1.1～3.1.4 ではシェルツールのライブラリのプロパティについて個々に説明します。

[PowerShell の実行手順]

1. シナリオへのライブラリの配置

以下のライブラリを順にシナリオへ配置してください。

3.1.1 PowerShell を開く

3.1.3 コマンドを実行する(PowerShell,コマンドプロンプト)

3.1.4 PowerShell・コマンドプロンプトを閉じる

配置した後のシナリオは図 3-1 のようになります。

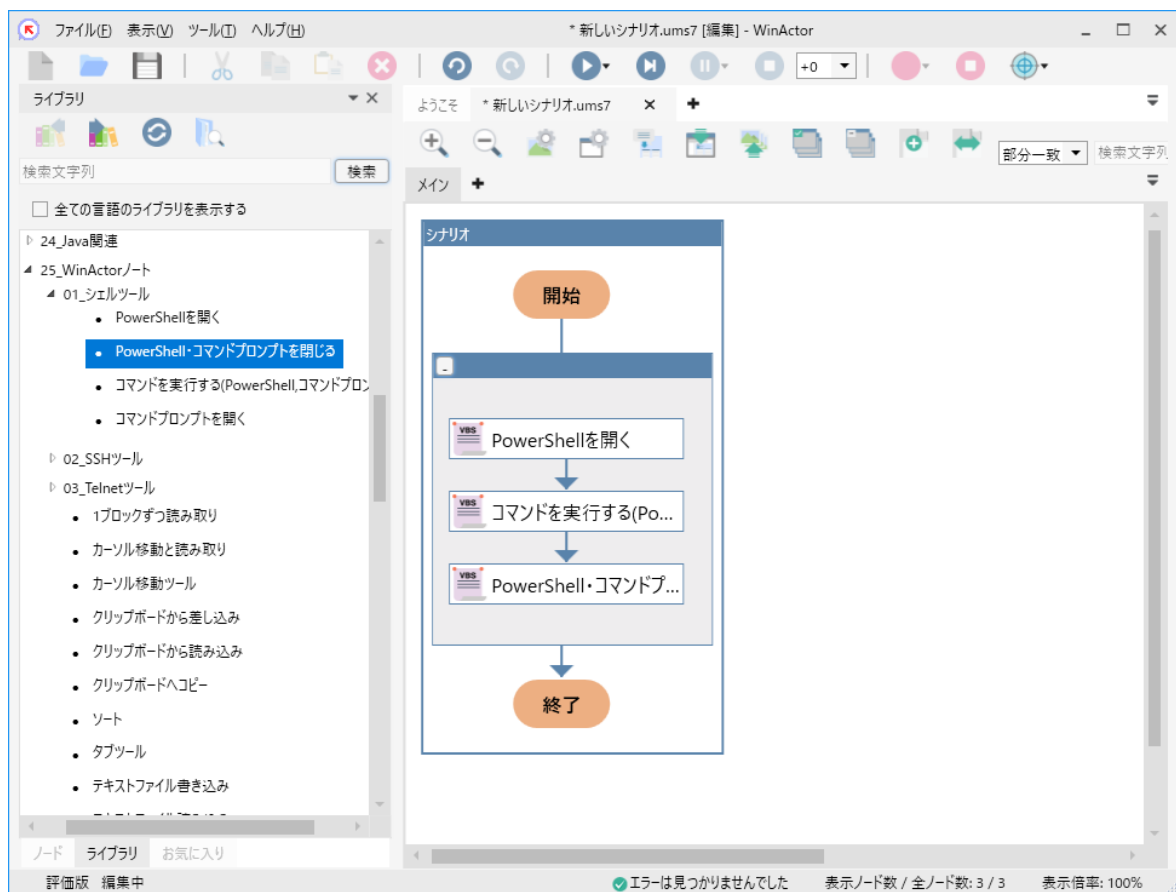


図 3-1 PowerShell を使用した基本的なシナリオ例

2. プロパティ設定

以下の章に記載の内容にしたがって、プロパティを設定してください。

3.1.1 PowerShell を開く

3.1.3 コマンドを実行する(PowerShell,コマンドプロンプト)

3.1.4 PowerShell・コマンドプロンプトを閉じる

[コマンドプロンプトの実行手順]

1. シナリオへのライブラリの配置

以下のライブラリを順にシナリオへ配置してください。

3.1.2 コマンドプロンプトを開く

3.1.3 コマンドを実行する(PowerShell,コマンドプロンプト)

3.1.4 PowerShell・コマンドプロンプトを閉じる

配置した後のシナリオは図 3-2 のようになります。

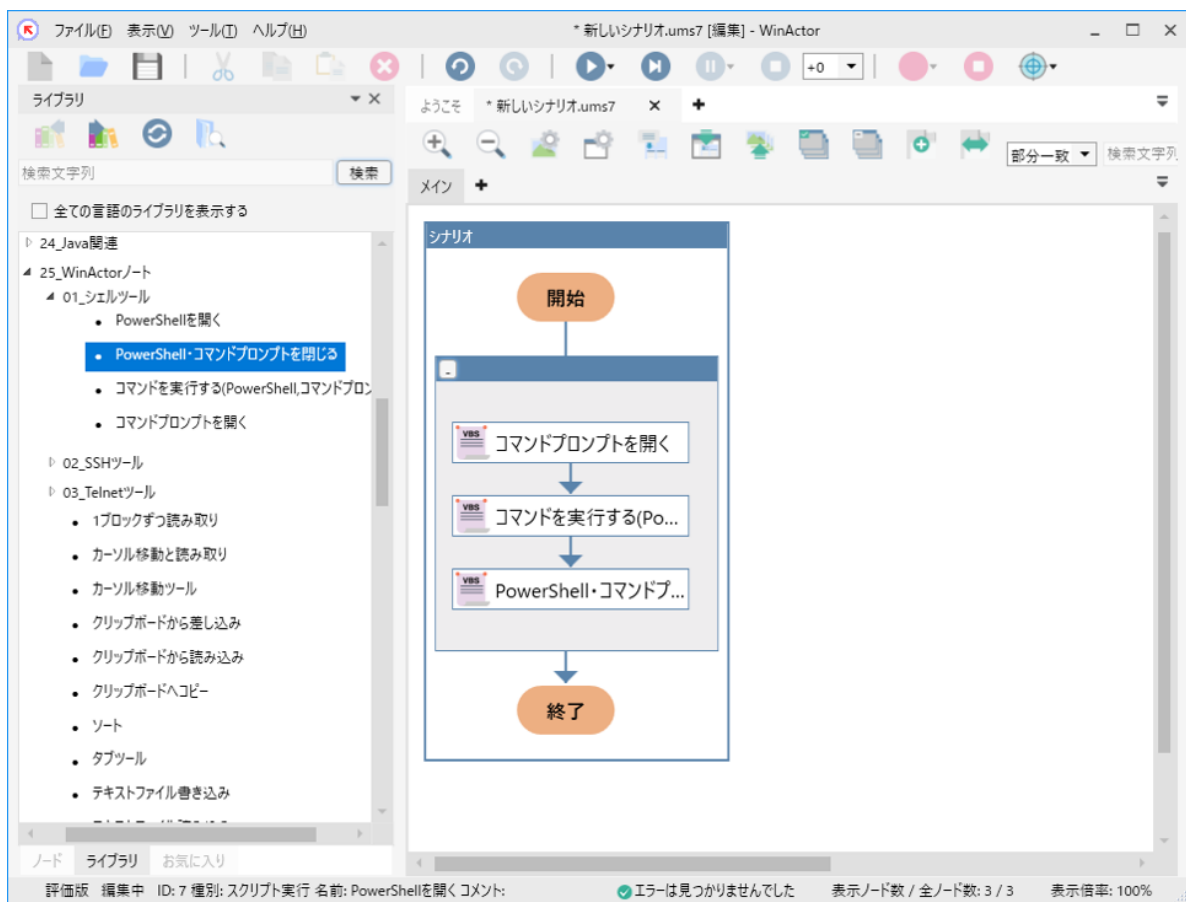


図 3-2 コマンドプロンプトを使用した基本的なシナリオ例

2. プロパティ設定

以下の章に記載の内容にしたがって、プロパティを設定してください。

3.1.2 コマンドプロンプトを開く

3.1.3 コマンドを実行する(PowerShell,コマンドプロンプト)

3.1.4 PowerShell・コマンドプロンプトを閉じる

3.1.1. PowerShell を開く

PowerShell を開きます。

表 3-1 「PowerShell を開く」ライブラリ設定項目

No.	項目	内容
①	文字コード	PowerShell への入力、および WinActor ノートへ出力する際の文字コードを指定できます。

3.1.2. コマンドプロンプトを開く

コマンドプロンプトを開きます。

表 3-2 「コマンドプロンプトを開く」ライブラリ設定項目

No.	項目	内容
①	文字コード	コマンドプロンプトへの入力、および WinActor ノートへ出力する際の文字コードを指定できます。

3.1.3. コマンドを実行する(PowerShell,コマンドプロンプト)

PowerShell,コマンドプロンプト上でコマンドを実行します。

表 3-3 「コマンドを実行する(PowerShell,コマンドプロンプト)」ライブラリ設定項目

No.	項目	内容
①	コマンド	PowerShell およびコマンドプロンプトで実行するコマンドを指定することができます。テキストのみ入力可能です。制御文字を送信することはできません。

3.1.4. PowerShell・コマンドプロンプトを閉じる

PowerShell,コマンドプロンプトのセッションを閉じます。

3.2. SSH ツール

SSH ツールで提供される SSH クライアント機能・ファイル送受信機能(SCP)を使用するためのライブラリについて本章で説明します。本章では基本的なシナリオの作成手順を説明します。3.2.1 では SSH クライアント・ファイル送受信機能(SCP)で共通のサーバへ接続する際の設定方法について、3.2.2~3.2.9 では SSH ツールのライブラリのプロパティについて個々に説明します。

[SSH クライアント機能の実行手順]

1. シナリオへのライブラリの配置

以下のライブラリを順にシナリオへ配置してください。

3.2.2 SSH クライアントを開く

3.2.4 コマンドを実行する(SSH クライアント)

3.2.5 SSH クライアントを閉じる

配置した後のシナリオは図 3-3 のようになります。

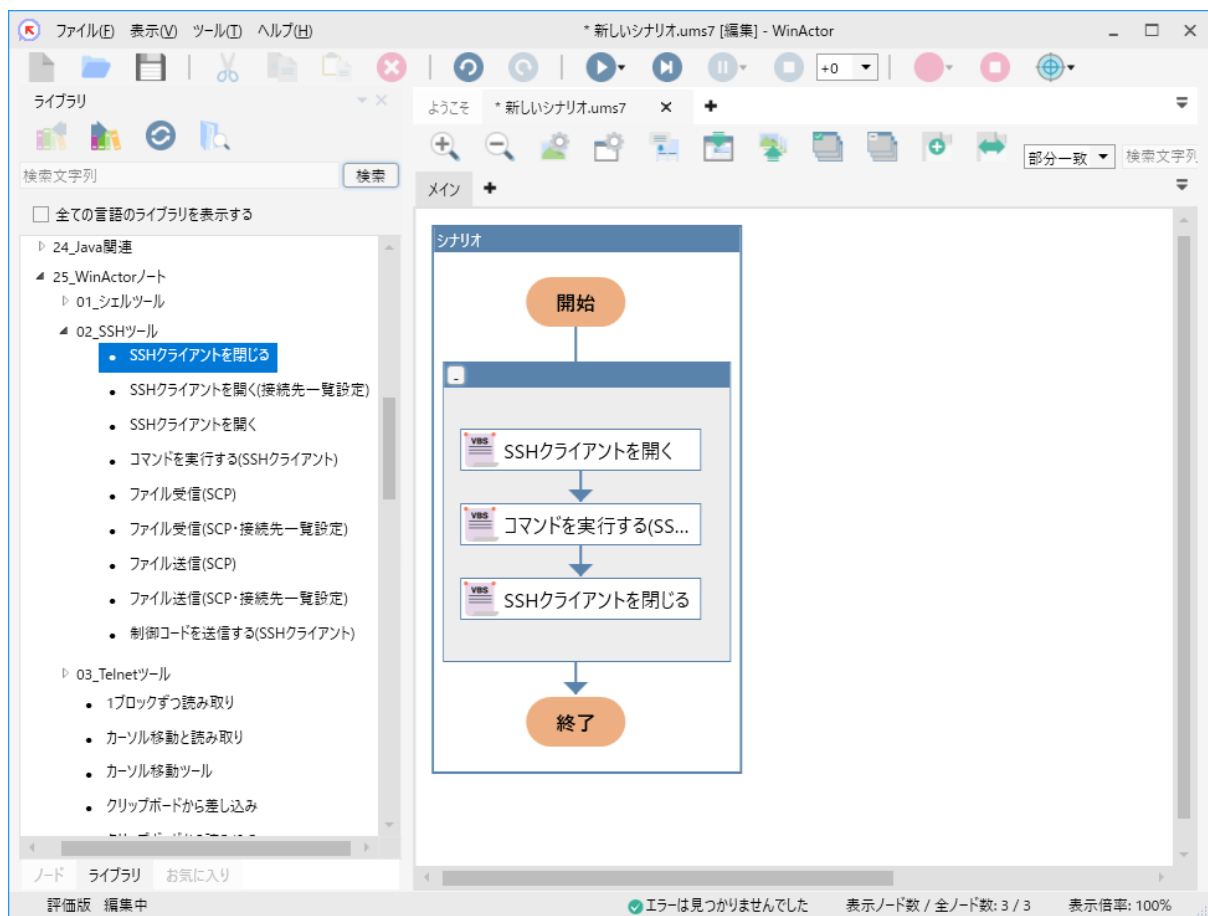


図 3-3 SSH クライアント機能を使用した基本的なシナリオ例

2. プロパティ設定

以下の章に記載の内容にしたがって、プロパティを設定してください。

3.2.2 SSH クライアントを開く

3.2.4 コマンドを実行する(SSH クライアント)

3.2.5 SSH クライアントを閉じる

[ファイル送信機能(SCP)の実行手順]

1. シナリオへのライブラリの配置

以下のライブラリをシナリオへ配置してください。

3.2.6 ファイル送信(SCP)

2. プロパティ設定

以下の章に記載の内容にしたがって、プロパティを設定してください。

3.2.6 ファイル送信(SCP)

[ファイル受信機能(SCP)の実行手順]

1. シナリオへのライブラリの配置

以下のライブラリをシナリオへ配置してください。

3.2.8 ファイル受信(SCP)

2. プロパティ設定

以下の章に記載の内容にしたがって、プロパティを設定してください。

3.2.8 ファイル受信(SCP)

3.2.1. 接続設定

SSH クライアント・ファイル送受信機能(SCP)で共通のサーバへ接続する際の設定方法について説明します。

SSH ツールでは、パスワード認証と公開鍵認証の 2 つの認証方式での SSH サーバへの接続が可能です。以下では上記 2 つの認証方式ごとに SSH サーバへ接続するために必要な設定項目について説明します。

[パスワード認証]

1. あらかじめ SSH サーバへ接続する際のパスワード情報を含むパスワードファイルを生成します。パスワードファイルの生成方法については、表 5-1 の No.3 の資料の「パスワードファイル生成ツール」の章をご参照ください。
2. 表 3-4 にしたがって値を設定します。ここで記載されている設定値は表 2-1 の例に当たっていますので、ご使用の環境に合わせて設定してください。

表 3-4 パスワード認証利用時の設定内容

項目	内容	本書の参考例における設定値
認証方式	認証方式を選択できます。	パスワード認証
文字コード	サーバへの入力、および WinActor ノートへ出力する際の文字コードを指定できます。	UTF-8
送信用改行コード	サーバへの入力する際の改行コードを指定できます。	LF(Linux, MacOS X)
ホスト	接続する SSH サーバの IPv4 アドレスを指定できます。	192.168.56.2
ポート番号	接続する SSH サーバのポート番号を指定できます。	22
ユーザ名	SSH サーバへログインする際のログイン名を指定できます。	user
パスワードファイル	SSH サーバにログインする際に必要なパスワードを含むパスワードファイルを指定できます。シナリオファイルの置かれたフォルダから相対パスで指定してください。	secret¥sshLogin.json ※シナリオファイルが置かれているフォルダに secret フォルダがあり、その中にパスワードファイル(sshLogin.json)が置かれている場合の例です。
秘密鍵ファイル	パスワード認証利用時は設定不要です。	

WinActor ノート 端末機能 シナリオ作成マニュアル

<p>コマンド 待ち文字列</p>	<p>ログイン処理が完了した際に表示されるプロンプトの末尾を含む文字列を指定できます。 複数指定したい場合は、カンマ区切りで入力してください。(例) “\$” と “#” を指定したい場合、“\$,#”。</p>	<p>“\$ “ “は入力不要です。\$の後ろはスペースです。</p>
<p>タイムアウト 値[秒]</p>	<p>SSH サーバへのログイン処理の各段階での最大待ち時間を秒単位で指定できます。ご使用になる環境に応じて適切な値を調整してください。</p>	<p>10 ※本設定値はあくまで例です。ご使用環境によっては動作しないこともあることをあらかじめご了承ください。</p>
<p>接続先一覧 ファイル</p>	<p>「接続先一覧ファイル生成ツール」で接続の確認が取れた接続先の接続先一覧ファイルを指定できます。シナリオファイルの置かれたフォルダから相対パスで指定してください。</p>	<p>[接続先一覧ファイルを設定する場合のみ] destination_hosts ※シナリオファイルが置かれているフォルダ直下に接続先一覧ファイル(destination_hosts)が置かれている場合の例です。</p>

[公開鍵認証]

1. 秘密鍵を生成します。秘密鍵の生成方法については、表 5-1 の No.3 の資料の「SSH 鍵生成ツール」の章をご参照ください。
2. 秘密鍵を生成する際、パスフレーズを設定した場合は、パスワードファイルを生成します。パスワードファイルの生成方法については、表 5-1 の No.3 の資料の「パスワードファイル生成ツール」の章をご参照ください。
3. 表 3-5 にしたがって値を設定します。ここで記載されている設定値は表 2-1 の例にしたがっていますので、ご使用の環境に合わせて設定してください。

表 3-5 公開鍵認証利用時の設定内容

項目	内容	本書の参考例における設定値
認証方式	認証方式を選択できます。	公開鍵認証
文字コード	サーバへの入力、および WinActor ノートへ出力する際の文字コードを指定できます。	UTF-8
送信用改行コード	サーバへの入力する際の改行コードを指定できます。	LF(Linux, MacOS X)
ホスト	接続する SSH サーバの IPv4 アドレスを指定できます。	192.168.56.2
ポート番号	接続する SSH サーバのポート番号を指定できます。	22
ユーザ名	SSH サーバへログインする際のログイン名を指定できます。	user
パスワードファイル	パスフレーズを付与した秘密鍵用のパスワードファイルを指定できます。シナリオファイルの置かれたフォルダから相対パスで指定してください。 秘密鍵にパスフレーズを設定しない場合は設定不要です。	[秘密鍵にパスフレーズを設定した場合のみ] secret¥sshLoginPassphrase.json ※シナリオファイルが置かれているフォルダに secret フォルダがあり、その中にパスワードファイル (sshLoginPassphrase.json) が置かれている場合の例です。
秘密鍵ファイル	秘密鍵ファイルを指定できます。シナリオファイルの置かれたフォルダから相対パスで指定してください。	secret¥sshLoginKey ※シナリオファイルが置かれているフォルダに secret フォルダがあり、その中に秘密鍵ファイル

		(sshLoginKey)が置かれている場合の例です。
コマンド 待ち文字列	ログイン処理が完了した際に表示されるプロンプトの末尾を含む文字列を指定できます。複数指定したい場合は、カンマ区切りで入力してください。 (例) "\$" と "#"を指定したい場合、"\$,#"。	“\$ “ “は入力不要です。\$の後ろはスペースです。
タイムアウト値 [秒]	SSH サーバへのログイン処理の各段階での最大待ち時間を秒単位で指定できます。ご使用になる環境に応じて適切な値を調整してください。	10 ※本設定値はあくまで例です。ご使用環境によっては動作しないこともあることをあらかじめご了承ください。
接続先一覧ファイル	「接続先一覧ファイル生成ツール」で接続の確認が取れた接続先の接続先一覧ファイルを指定できます。シナリオファイルの置かれたフォルダから相対パスで指定してください。	[接続先一覧ファイルを設定する場合のみ] destination_hosts ※シナリオファイルが置かれているフォルダ直下に接続先一覧ファイル (destination_hosts) が置かれている場合の例です。

3.2.2. SSH クライアントを開く

SSH クライアントを開きます。

表 3-6 「SSH クライアントを開く」ライブラリ設定項目

No.	項目	内容
①	認証方式	3.2.1 をご参照ください。
②	文字コード	
③	送信用改行コード	
④	ホスト	
⑤	ポート番号	
⑥	ユーザ名	
⑦	パスワードファイル	
⑧	秘密鍵ファイル	
⑨	コマンド待ち文字列	
⑩	タイムアウト値 [秒]	

3.2.3. SSH クライアントを開く(接続先一覧設定)

接続先一覧ファイルを指定して SSH クライアントを開きます。

接続先一覧ファイルの生成方法については、表 5-1 の No.3 の資料の「接続先一覧ファイル生成ツール」の章をご参照ください。

表 3-7 「SSH クライアントを開く(接続先一覧設定)」ライブラリ設定項目

No.	項目	内容
①	認証方式	3.2.1 をご参照ください。
②	文字コード	
③	送信用改行コード	
④	ホスト	
⑤	ポート番号	
⑥	ユーザ名	
⑦	パスワードファイル	
⑧	秘密鍵ファイル	
⑨	コマンド待ち文字列	
⑩	タイムアウト値 [秒]	
⑪	接続先一覧ファイル	

3.2.4. コマンドを実行する(SSHクライアント)

SSHクライアント上でコマンドを実行します。

表 3-8 「コマンドを実行する(SSHクライアント)」ライブラリ設定項目

No.	項目	内容
①	コマンド	SSHクライアントで実行するコマンドを指定することができます。テキストのみ入力可能です。制御文字を送信することはできません。
②	コマンド待ち文字列	コマンドの処理が完了したことを表すメッセージの末尾を含む文字列を指定できます。 通常は 3.2.1 の「コマンド待ち文字列」と同一のようになります。
③	タイムアウト値 [秒]	②が表示されるまでの最大待ち時間を指定できます。

3.2.5. SSHクライアントを閉じる

SSHクライアントのセッションを閉じます。

3.2.6. ファイル送信(SCP)

SCP によるファイル送信を行います。

表 3-9 「ファイル送信(SCP)」ライブラリ設定項目

No.	項目	内容
①	認証方式	3.2.1 をご参照ください。
②	ホスト	
③	ポート番号	
④	ユーザ名	
⑤	パスワードファイル	
⑥	秘密鍵ファイル	
⑦	タイムアウト値 [秒]	
⑧	送信元ファイル	送信するファイルをシナリオファイルの置かれたフォルダからの相対パスで指定できます。単一のファイルのみ指定できます。複数ファイルやフォルダは指定できません。
⑨	送信先パス	SSH サーバ上の送信元ファイル(⑧)の格納先のパスを入力できます。

3.2.7. ファイル送信(SCP・接続先一覧設定)

接続先一覧ファイルを指定して SCP によるファイル送信を行います。

接続先一覧ファイルの生成方法については、表 5-1 の No.3 の資料の「接続先一覧ファイル生成ツール」の章をご参照ください。

表 3-10 「ファイル送信(SCP・接続先一覧設定)」ライブラリ設定項目

No.	項目	内容
①	認証方式	3.2.1 をご参照ください。
②	ホスト	
③	ポート番号	
④	ユーザ名	
⑤	パスワードファイル	
⑥	秘密鍵ファイル	
⑦	タイムアウト値 [秒]	
⑧	接続先一覧ファイル	
⑨	送信元ファイル	送信するファイルをシナリオファイルの置かれたフォルダからの相対パスで指定できます。単一のファイルのみ指定できます。複数ファイルやフォルダは指定できません。
⑩	送信先パス	SSH サーバ上の送信元ファイル(⑨)の格納先のパスを入力できます。

3.2.8. ファイル受信(SCP)

SCP によるファイル受信を行います。

表 3-11 「ファイル受信(SCP)」ライブラリ設定項目

No.	項目	内容
①	認証方式	3.2.1 をご参照ください。
②	ホスト	
③	ポート番号	
④	ユーザ名	
⑤	パスワードファイル	
⑥	秘密鍵ファイル	
⑦	タイムアウト値 [秒]	
⑧	送信元ファイル	SSH サーバ上の受信するファイルを指定できます。フォルダは指定できません。「本チュートリアルで使用する環境」ではファイルにワイルドカード(*や?)を使用できます。
⑨	送信先パス	SSH サーバ上の送信元ファイル(⑧)の格納先のパスを入力できます。

3.2.9. ファイル受信(SCP・接続先一覧設定)

接続先一覧ファイルを指定して SCP によるファイル受信を行います。

接続先一覧ファイルの生成方法については、表 5-1 の No.3 の資料の「接続先一覧ファイル生成ツール」の章をご参照ください。

表 3-12 「ファイル受信(SCP・接続先一覧設定)」ライブラリ設定項目

No.	項目	内容
①	認証方式	3.2.1 をご参照ください。
②	ホスト	
③	ポート番号	
④	ユーザ名	
⑤	パスワードファイル	
⑥	秘密鍵ファイル	
⑦	タイムアウト値 [秒]	
⑧	接続先一覧ファイル	
⑨	送信元ファイル	SSH サーバ上の受信するファイルを指定できます。フォルダは指定できません。「本チュートリアルで使用する環境」ではファイルにワイルドカード(*や?)を使用できます。
⑩	送信先パス	SSH サーバ上の送信元ファイル(⑨)の格納先のパスを入力できます。

3.2.10. 制御コードを送信する(SSH クライアント)

SSH クライアント上で制御コードを送信します。

表 3-13 「制御コードを送信する(SSH クライアント)」ライブラリ設定項目

No.	項目	内容
①	制御コード	SSH クライアントで送信する制御コードを指定することができます。Ctrl+A~Ctrl+Z が指定できます。
②	コマンド待ち文字列	コマンドの処理が完了したことを表すメッセージの末尾を含む文字列を指定できます。 通常は 3.2.1 の「コマンド待ち文字列」と同一のようになります。
③	タイムアウト値 [秒]	②が表示されるまでの最大待ち時間を指定できます。

3.3. Telnet ツール

Telnet クライアント機能を使用するためのライブラリについて本章で説明します。本章では基本的なシナリオの作成手順を説明します。3.3.1～3.3.3 では Telnet ツールのライブラリのプロパティについて個々に説明します。

[Telnet クライアント機能の実行手順]

1. シナリオへのライブラリの配置

以下のライブラリを順にシナリオへ配置してください。

3.3.1 Telnet クライアントを開く

3.3.2 コマンドを実行する(Telnet クライアント)

3.3.3 Telnet クライアントを閉じる

配置した後のシナリオは図 3-4 のようになります。

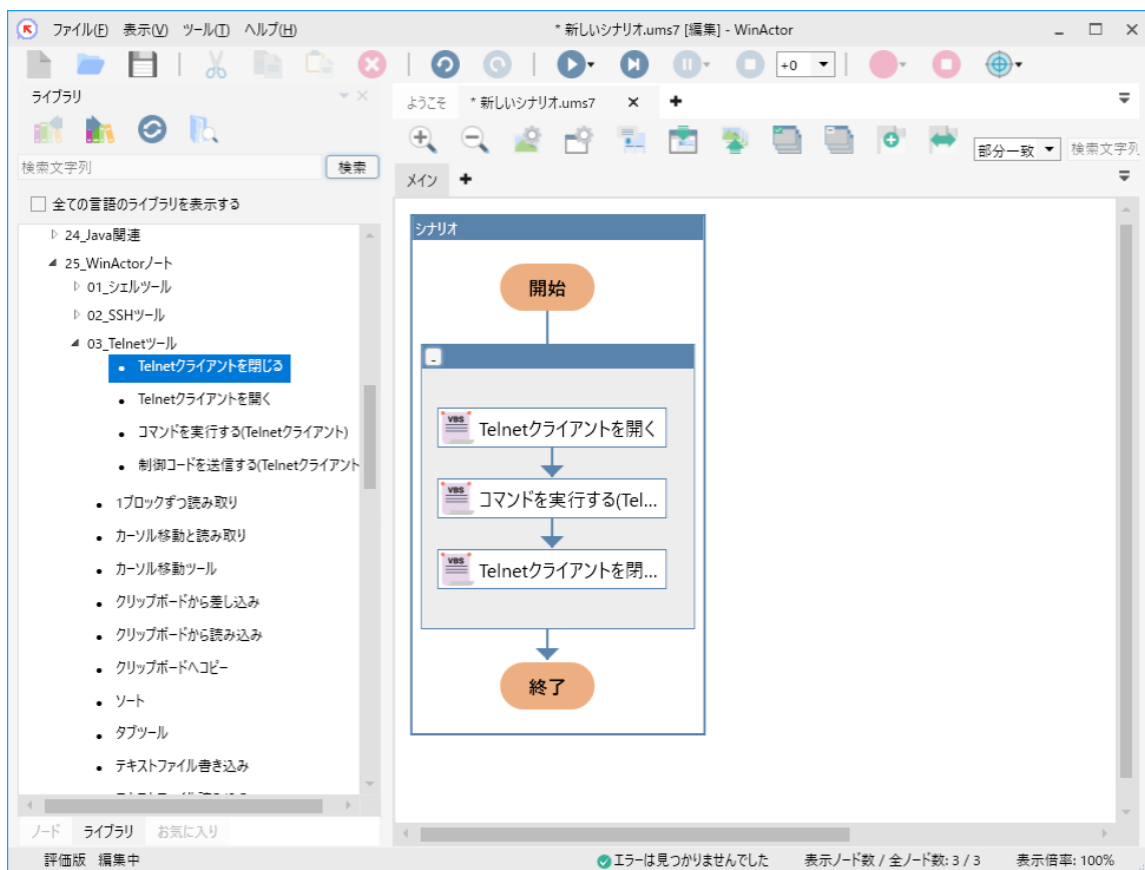


図 3-4 Telnet クライアント機能を使用した基本的なシナリオ例

2. プロパティ設定

以下の章に記載の内容にしたがって、プロパティを設定してください。

3.3.1 Telnet クライアントを開く

3.3.2 コマンドを実行する(Telnet クライアント)

3.3.3 Telnet クライアントを閉じる

3.3.1. Telnet クライアントを開く

Telnet クライアントを開きます。

表 3-14 「Telnet クライアントを開く」ライブラリ設定項目

No.	項目	内容
①	文字コード	サーバへの入力、および WinActor ノートへ出力する際の文字コードを指定できます。
②	ホスト	接続する Telnet サーバの IPv4 アドレスを指定できます。
③	ポート番号	接続する Telnet サーバのポート番号を指定できます。
④	ユーザ名	Telnet サーバへログインする際のログイン名を指定できます。
⑤	パスワードファイル	Telnet サーバにログインする際に必要なパスワードを含むパスワードファイルを指定できます。シナリオファイルの置かれたフォルダから相対パスで指定してください。
⑥	ログイン待ち文字列	Telnet サーバがユーザ名の入力を促すメッセージの末尾を含む文字列を指定できます。
⑦	パスワード待ち文字列	Telnet サーバがパスワードの入力を促すメッセージの末尾を含む文字列を指定できます。
⑧	コマンド待ち文字列	ログイン処理が完了した際に表示されるプロンプトの末尾を含む文字列を指定できます。 複数指定したい場合は、カンマ区切りで入力してください。(例) "\$ " と "# "を指定したい場合、"\$,# "。
⑨	タイムアウト値 [秒]	Telnet サーバへのログイン処理の各段階での最大待ち時間を秒単位で指定できます。ご使用になる環境に応じて適切な値を調整してください。

※ 送信用改行コードは CR+LF 固定となります。

3.3.2. コマンドを実行する(Telnet クライアント)

Telnet クライアント上でコマンドを実行します。

表 3-15 「コマンドを実行する(Telnet クライアント)」ライブラリ設定項目

No.	項目	内容
①	コマンド	Telnet クライアントで実行するコマンドを指定することができます。テキストのみ入力可能です。制御文字を送信することはできません。
②	コマンド待ち文字列	コマンドの処理が完了したことを表すメッセージの末尾を含む文字列を指定できます。 通常は 3.3.1 の「コマンド待ち文字列」と同一のものになります。
③	タイムアウト値 [秒]	②が表示されるまでの最大待ち時間を指定できます。

3.3.3. Telnet クライアントを閉じる

Telnet クライアントのセッションを閉じます。

3.3.4. 制御コードを送信する(Telnet クライアント)

Telnet クライアント上で制御コードを送信します。

表 3-16 「制御コードを送信する(Telnet クライアント)」ライブラリ設定項目

No.	項目	内容
①	制御コード	Telnet クライアントで送信する制御コードを指定することができます。Ctrl+A~Ctrl+Z が指定できます。
②	コマンド待ち文字列	コマンドの処理が完了したことを表すメッセージの末尾を含む文字列を指定できます。 通常は 3.3.1 の「コマンド待ち文字列」と同一のものになります。
③	タイムアウト値 [秒]	75 が表示されるまでの最大待ち時間を指定できます。

4. ドッキングウィンドウ

ドッキングウィンドウについての説明は「WinActor ノート 操作マニュアル」を参照。

5. 参照資料

本マニュアル中にて参照する資料について表 5-1 に示します。

表 5-1 参照資料

No.	資料名
1	WinActor 操作マニュアル
2	WinActor ユーザライブラリサンプル説明書
3	WinActor ノート 操作マニュアル
4	WinActor ノート テキスト処理シナリオ作成マニュアル



WinActor ノート
端末機能シナリオ作成マニュアル

NTTアドバンステクノロジー株式会社

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

本書は著作権法上の保護を受けています。本書の一部あるいは全部を無断で複写、複製することは禁じられています。

本マニュアルの内容は予告なく変更される場合があります。

WA7-P-20250603
