



# ITA\_User instruction manual

Ansible-driver

— Version 1.5 —

Copyright © NEC Corporation 2019. All rights reserved.

## Disclaimer

All the contents of this document are protected by copyright owned by NEC Corporation.  
Unauthorized reproduction or copying of all or part of the contents of this document is prohibited.  
The contents of this document are subject to change without prior notice in the future.  
NEC Corporation is not responsible for any technical or editorial errors or omissions in this document.  
NEC Corporation do not guarantee accuracy, usability, certainty of the content in this document.

## Trademark

- Linux is registered trademark or trademark of Linus Torvalds, registered in the U.S. and other countries.
- Red Hat is registered trademark or trademark of Red Hat, Inc., registered in the U.S. and other countries.
- Apache, Apache Tomcat, Tomcat are registered trademarks or trademarks of Apache Software Foundation.
- Ansible is a registered trademark or trademark of Red Hat, Inc.
- AnsibleTower is a registered trademark or trademark of Red Hat, Inc.

The names of other systems, company name and products mentioned in this document are registered trademarks or trademarks of their respective companies.

The ® mark and TM mark is not specified in this document.

※「Exastro IT Automation」is written as「ITA」in this document.

## Table of contents

---

Table of contents .....	2
Introduction .....	4
1 Overview of Ansible driver .....	5
1.1 About Ansible .....	5
1.2 About AnsibleTower .....	5
1.3 About Ansible driver .....	6
2 Variable handling in Ansible driver .....	7
2.1 Variable type .....	7
2.2 Extract variables and register specific values .....	9
2.3 Variable handling according to substitution value registration .....	10
3 Ansible driver console menu configuration .....	11
3.1 Menu/screen list .....	11
4 Ansible driver operation procedure .....	13
4.1 Workflow .....	13
4.1.1 Workflow of Ansible-Legacy .....	13
4.1.2 Workflow of Ansible-Legacy Role .....	16
4.1.3 Workflow of Ansible-Pioneer .....	19
5 Ansible driver function-operation method explanation .....	22
5.1 Basic console .....	22
5.1.1 Device list .....	22
5.1.2 Input operation list .....	26
5.2 Ansible common console .....	27
5.2.1 Interface information .....	27
5.2.2 Ansible Tower host list .....	32
5.2.3 Global variable list .....	34
5.2.4 Template list .....	36
5.2.5 Contents list .....	40
5.3 Ansible-Legacy／Legacy Role／Pioneer console .....	43
5.3.1 OS type master .....	43
5.3.2 Movement list .....	45
5.3.3 Playbook file list (Ansible-Legacy only) .....	48
5.3.4 Role package list (Ansible-Legacy Role only) .....	50
5.3.5 Dialog type list (Ansible-Pioneer only) .....	52
5.3.6 Dialog files (Ansible-Pioneer only) .....	53
5.3.7 Movement details .....	55
5.3.8 Nested variable maximum iteration count list (Ansible-Legacy Role only) .....	57
5.3.9 Substitution value auto-registration setting .....	59
5.3.10 Target host .....	62
5.3.11 Substitution value list .....	63
5.3.12 Check operation status .....	69
5.3.13 Execution list .....	72
5.3.14 Execution .....	73
6 How to write construction code .....	74
6.1 Describe Playbook (Ansible-Legacy) .....	74
6.2 Describe Dialog file (Ansible-Pioneer) .....	75

6.3	Describe role package (Ansible-Legacy Role) .....	87
6.4	Describe ITA readme(Ansible-Legacy Role only).....	91
6.5	Describe substitution table(Ansible-Legacy Role only) .....	93
6.6	BackYard contents .....	95
6.7	Ansible usage guideline ITA additional rules .....	98
7	Application operation.....	99
7.1	Maintenance.....	99
7.2	About the maintenance method.....	100
8	Appendix .....	101
8.1	The linkage between the input data used during Ansible execution and ITA menu.....	101
8.1.1	Ansible-Legacy input data.....	102
8.1.2	Ansible-Pioneer input data .....	104
8.1.3	Ansible-LegacyRole input data .....	106
8.2	Result data created during Ansible execution .....	108
8.2.1	Legacy/LegacyRole List of files saved in result data .....	108
8.2.2	List of files saved in Pioneer result data .....	109

## Introduction

---

This document explains the function and the operation method of ITA.

# 1 Overview of Ansible driver

This chapter explains Ansible, AnsibleTower, and Ansible driver.

## 1.1 About Ansible

Ansible is a platform construction automation tool that makes deploying applications / systems to many construction management targets easy.

Ansible can implement various operations by describing YAML textfiles called Playbook which record routine operations and executing them.

Tasks are linked to processing programs called modules, and can control various devices.

For the detailed information of Ansible, please refer to the product manual of Ansible.

The version of Ansible is 2.2.1.0. Please note that the notation according to the newest Ansible may not be used.

## 1.2 About AnsibleTower

AnsibleTower is a management platform that extends the function of Ansible, a platform construction automation tool such as "access control", "job scheduling", "task visualization", etc.

Ansible can combine "project", "inventory", "credentials" to create "job template" and execute with Ansible.

By combining multiple "Job Templates" to create a "Workflow Job Template", a more diverse workflow can be expressed.

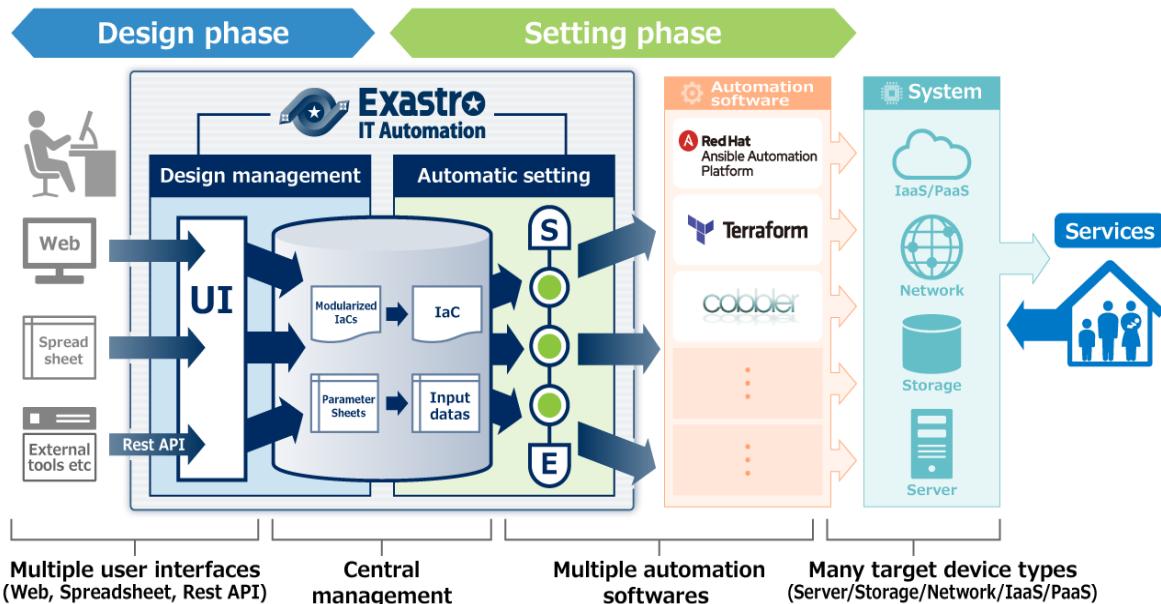
For the detailed information of AnsibleTower, please refer to the product manual of AnsibleTower.

For the ITA compatible AnsibleTower version, please refer to "System Configuration/Environment Construction Guide-Ansible-driver".

Please note that the notation according to the newest version may not be used.

## 1.3 About Ansible driver

Ansible driver operates as options of ITA system. Ansible driver selects using Ansible or AnsibleTower and perform actual operation configuration automatically according to each construction target server, storage, network device registered on ITA system.



The following 3 modes are available in Ansible driver according to the usage.

### ① Legacy mode

Configure settings to various hosts using the standard function of Ansible.

Register the construction code in a single YAML file and construct the operation pattern with that combination.

This mode is assumed to be used in works such as environment configuration of operation system and network

### ② Legacy Role mode

Same with the Legacy mode, configure settings to various hosts using the standard function of Ansible.

Register construction code as package and construct work pattern with the combination of Role. It is assumed that this mode will be used when using the Role package provided by the product department to install product or construct environment, etc.

### ③ Pioneer mode

It is possible to add individual module to Ansible and input setting in interactive mode.

Supports all devices that can login using Telnet or SSH, regardless of server, storage, or network device.

Since interacting with target device directly is required, appropriate IT skills are required.

In addition, Ansible driver can configure the variables in Playbook from screen. For details please refer to "[2 Variable handling in Ansible driver](#)" in this manual.

## 2 Variable handling in Ansible driver

### 2.1 Variable type

Ansible driver can set the specific value of Playbook variable in the setting screen in ITA.

※For the detail of setting method, please refer to "5.3.11 Substitution value list" in this manual

The six types of Playbook variable that can be handled as ITA variable are as below.

Type	Content	Legacy	Pioneer	Legacy Role
Normal variable	A variable that can define one specific value to the variable name. Please describe the variable in Playbook as {{△VAR_xxx△}}. △:half-width space xxx: half-width alphanumeric character and underscore( _ ) e.g.) VAR_users: root	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Multiple specific value variable	A variable that can define multiple specific value for a variable name. Please describe the variable in Playbook as {{△VAR_xxx△}}. △:half-width space xxx: half-width alphanumeric character and underscore( _ ) e.g.) VAR_users: - root - mysql	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nested variable	Hierarchical variables. Please describe the variable in Playbook as {{△VAR_xxx△}}. △:half-width space xxx: half-width alphanumeric character and underscore( _ ) e.g.) VAR_users: - user name: alice authorized: password Member variable names are ascii characters excluding the following 7 characters (0x20 ~ 0x7e) can be used. " . [ ] ' ¥ : For more information, please see <a href="#">Yaml syntax</a> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
Global variable	Variable registered from the "Global variable" menu.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Type	Content	Legacy	Pioneer	Legacy Role																										
ITA original variable	<p>Original variable defined by ITA.</p> <p>The following items in the basic console device list can be handled as variables.</p> <table border="1"> <thead> <tr> <th>Item name</th><th>variable name</th></tr> </thead> <tbody> <tr> <td>host name</td><td>__loginhostname__</td></tr> <tr> <td>protocol</td><td>__loginprotocol__</td></tr> <tr> <td>login user ID</td><td>__loginuser__</td></tr> <tr> <td>login password</td><td>__loginpassword__</td></tr> </tbody> </table> <p><b>The 「__」 surrounding the variable name is a pair of 2 half-width underscore.</b></p> <p>For "device list", please refer to "User instruct manual_basic console"</p> <p>The directory path during operation execution can be handled as following variable.</p> <table border="1"> <thead> <tr> <th>Item name</th><th>Variable name</th></tr> </thead> <tbody> <tr> <td>Operation directory path</td><td>__workflowdir__</td></tr> </tbody> </table> <p>By creating a file under the operation directory path in Playbook, users can download the result data file of "<u>operation execution</u>" menu.</p> <p>The directory path shared by each movement when executing Symphony is handled as the following variables.</p> <table border="1"> <thead> <tr> <th>Item name</th><th>Variable name</th></tr> </thead> <tbody> <tr> <td>Symphony</td><td>__symphony_workflowdir__</td></tr> <tr> <td>Operation directory path</td><td>__symphony_workflowdir__</td></tr> </tbody> </table> <p>By creating files under the Symphony operation directory path in Playbook, files can be shared between Movements. Also, during operation execution, the same path as __workflowdir__ is set.</p> <p>The directory path shared by each movement when executing Conductor is handled as the following variables.</p> <table border="1"> <thead> <tr> <th>Item name</th><th>Variable name</th></tr> </thead> <tbody> <tr> <td>Conductor</td><td>__conductor_workflowdir__</td></tr> <tr> <td>Operation directory path</td><td>__conductor_workflowdir__</td></tr> </tbody> </table> <p>By creating files under the Conductor operation directory path in Playbook, files can be shared between Movements. Also, during operation execution, the same path as __workflowdir__ is set.</p>	Item name	variable name	host name	__loginhostname__	protocol	__loginprotocol__	login user ID	__loginuser__	login password	__loginpassword__	Item name	Variable name	Operation directory path	__workflowdir__	Item name	Variable name	Symphony	__symphony_workflowdir__	Operation directory path	__symphony_workflowdir__	Item name	Variable name	Conductor	__conductor_workflowdir__	Operation directory path	__conductor_workflowdir__	O	O	O
Item name	variable name																													
host name	__loginhostname__																													
protocol	__loginprotocol__																													
login user ID	__loginuser__																													
login password	__loginpassword__																													
Item name	Variable name																													
Operation directory path	__workflowdir__																													
Item name	Variable name																													
Symphony	__symphony_workflowdir__																													
Operation directory path	__symphony_workflowdir__																													
Item name	Variable name																													
Conductor	__conductor_workflowdir__																													
Operation directory path	__conductor_workflowdir__																													
Substitution variable	<p>Variable "LCA_XXX" that used to handle variables in Defaults variable definition file or ITA readme other than "VAR_XXX" type in ITA.</p> <p>Please refer to "6.7 Describe substitution table(Ansible-Legacy Role only)" for details.</p>	x	x	O																										

## 2.2 Extract variables and register specific values

In each mode, variables are extracted from files uploaded to ITA such as Playbook.

The method to extract variable is as below.

### (1) Ansible-Legacy

Extract the variable definitions in the following format from the Playbook uploaded in "Playbook file list (5.3.3 Playbook file list (Ansible-Legacy only) in this manual)".

`{{△VAR_xxx△}}`

※ △:half-width space xxx: half-width alphanumeric character and underscore ( \_ )

Ansible-Legacy determines whether to register specific values as normal variables or multiple specific value variables. Please refer to "[5.3.11 Substitution value list](#)" for details.

### (2) Ansible-Pioneer

Extract the same variable definition as Ansible-Legacy from the dialog file uploaded in "Dialog files([5.3.6 Dialog files \(Ansible-Pioneer only\)](#) in this manual)"

### (3) Ansible-Legacy Role

Extract the same variable as Ansible-Legacy from the Playbook in role package uploaded in "Role package list([5.3.4 Role package list \(Ansible-Legacy Role only\)](#)in this manual)"

Please refer to "[Role package list \(5.3.4 Role package list \(Ansible-Legacy Role only\)\)](#)in this manual" for details.

Also, by creating substitution table, ITA can handle the variables other than "VAR\_xxx" defined in defaults variable definition file and ITA readme. Please refer to "[6.7 Describe substitution table\(Ansible-Legacy Role only\)](#)" for details.

It is possible to extract variables from the files uploaded to ITA such as Playbook and register their specific value.

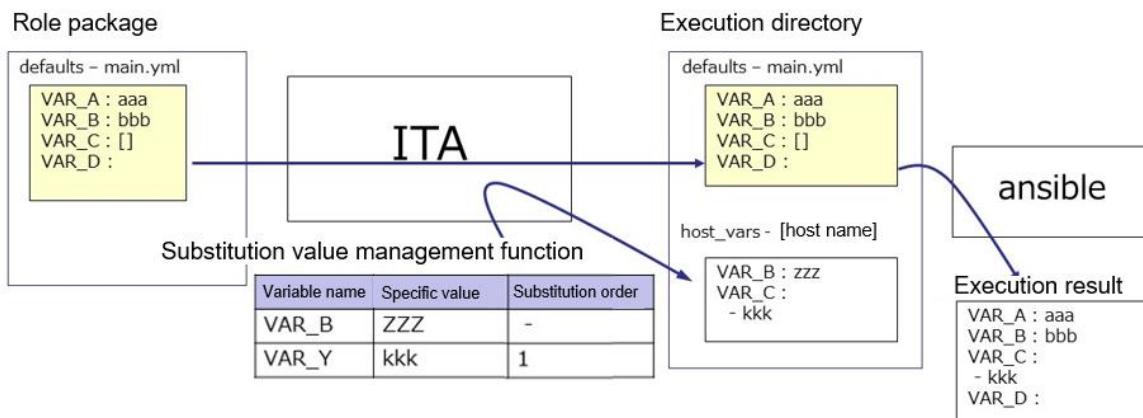
The specific value of the extracted variables are registered in "[5.3.9 Substitution value auto-registration setting](#)" and "[5.3.11 Substitution value list](#)".

The registered variable and specific value will be output in host variable file during work execution.

## 2.3 Variable handling according to substitution value registration

By using substitution value registration function, it is possible to overwrite the value of variable defined in Playbook.

The relationship between Playbook variable and the variable value registered in substitution value management function is shown as the following figure.



The value of the variable registered in the substitution value management function is output to the variable definition file (`host_vars`) for each host, and executed on each host by using the original Playbook and variable definition file as input in Ansible.

The priority of variable values in the result is as below.

- ① Value registered in substitution value management function
  - ② Value specified in Playbook variable
- Please refer to "[5.3.11 Substitution value list](#)" for details.

### 3 Ansible driver console menu configuration

This chapter explains the configuration of ITA console menu.

For the method to log in Web console and the components / basic operations of the menu screen, please refer to the "[First Step Guide](#)".

#### 3.1 Menu/screen list

##### ① ITA basic console menu

The menu list of ITA basic console used in Ansible driver is as below.

**Table 3.1-1 Basic console menu/screen list**

No	Menu group	Menu·Screen	Description
1	ITA basic console	Device list	Maintain (View/Register/Update/Discard) operation target system list.
2		Associated menu	Manage the configuration management database associated with Substitution value auto-registration setting menu.
3		Input operation list	Maintain(View/Register/Update/Discard) input operation list

##### ② Ansible common console menu

Ansible common console menu list is as below.

**Table 3.1-2 Common console menu/screen list**

No	Menu group	Menu·Screen	Description
1	Ansible common console	Interface information	Select whether to use Ansible or Ansible Tower server as the execution engine for the construction operation. Manage the path of directory shared by ITA system, Ansible driver server, and execution engine server and the connection interface information to the execution engine server.
2		Ansible Tower host list	Manage the information required to execute Rest API of Ansible Tower and the information required to transfer the construction materials to Ansible Tower.
3		Global variable list	Manage the variable commonly used in Playbook or dialog file, etc. (referred to as global variable hereafter) and their specific values.
4		Template list	Manage template files and embedded variables used in the template modules, etc. in Playbook.
5		Contents list	Manage the files and embedded variable used in each module in Playbook.

### ③ Ansible console menu

The menu list according to each Ansible console is as below.

**Table 3.1-3 Ansible driver console menu/screen list**

No	menu group			Menu·Screen	Description
	Ansible console	Legacy	Role		
1			<input type="radio"/>	OS type	Manage the type of OS of the devices.
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Movement list	Manage the list of Movements registered in Symphony.
3	<input type="radio"/>			Playbook files	Manage Playbook file.
4		<input type="radio"/>		Role package list	Manage role package.
5			<input type="radio"/>	Dialog type list	Manages the type of grouping dialog files of the same purpose as dialog type.
6			<input type="radio"/>	Dialog files	Manage the OS type associated with the dialog type and the ITA system original format work procedure file (referred to as dialog file in the following).
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Movement details	Manage the association between Movement and Playbook file.
8		<input type="radio"/>		Nested variable maximum iteration count list	Manage the maximum iteration array count if nested variable is configured as iterative array.
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Substitution value auto-registration setting	Manage Movement and variable associated to every item value of operation and host registered in the configuration management database menu.
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Target host	Manage the host used in Movement.
11	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Substitution value list	Manage the substitution value of variable.
12	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Execution	Select the Movement and Operation for work execution and indicate the execution.
13	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Check operation status	Displays the operation execution status.
14	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Execution list	Manage the operation execution history.

※ Since user operations such as configuration are not performed in Ansible RestAPI, the description is omitted here.

## 4 Ansible driver operation procedure

This chapter explains the operation procedure for using each Ansible console.

### 4.1 Workflow

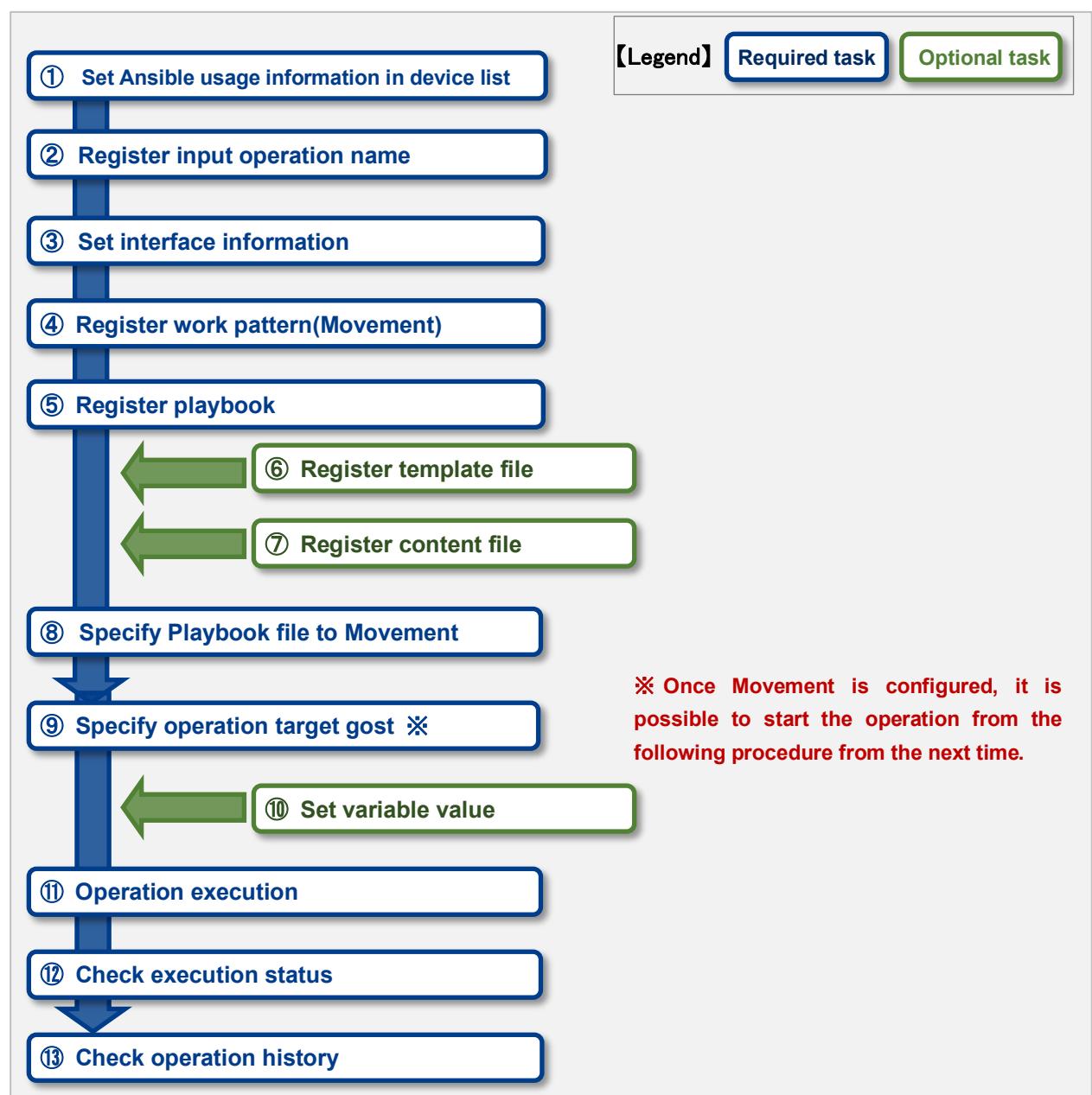
The standard workflow of each Ansible console is as follow.

The details of each operation is described in the next section.

Please refer to "User instruction manual\_Basic console" for how to use the ITA basic console.

#### 4.1.1 Workflow of Ansible-Legacy

The workflow of executing operation using Ansible-Legacy is as follows.



- **Workflow details and references**

- ① **Set Ansible usage information in device list**

Set the Ansible usage information to each devices in the device list screen of ITA basic console.

Please refer to “[5.1.1 Device list](#)” for details.

- ② **Register input operation name**

Register the input operation name for work from the input operation list screen of ITA basic console.

Please refer to “[5.1.2 Input operation list](#)” for details.

- ③ **Set interface information**

Select using whether Ansible or AnsibleTower server as the execution engine and register the connection information of the execution engine server from the interface information screen of Ansible common console.

Please refer to “[5.2.1 Interface information](#)” for details.

- ④ **Register work pattern (Movement)**

Register the Movement for operation from the Movement list screen of Ansible-Legacy console.

Please refer to “[5.3.2 Movement list](#)” for details.

- ⑤ **Register playbook**

Register the Playbook used in operation from the Playbook files screen of Ansible-Legacy console.

Please refer to “[5.3.3 Playbook file list \(Ansible-Legacy only\)](#)” for details.

- ⑥ **Register template file (execute if necessary)**

Register/Update/Discard the template file (src) and the template embedded variable used in the template module, etc. of Playbook from the template list screen of Ansible common console.

Please refer to “[5.3.4 Template list](#)” for details.

- ⑦ **Register content file (execute if necessary)**

Register the file used to configure the operation target server from the contents list screen of Ansible common console.

Please refer to “[5.3.5 Contents list](#)” for details.

- ⑧ **Specify Playbook file to Movement**

Specify Playbook file to the registered Movement from movement details screen of Ansible-Legacy console.

Please refer to “[5.3.7 Movement details](#)” for details.

- ⑨ **Specify operation target host**

Specify the operation target host from the target host screen of Ansible-Legacy console.

Please refer to “[5.3.10 Target host](#)” for details.

**⑩ Set variable value (execute if necessary)**

Set the value of the variable in the Playbook which has been registered to Movement from the substitution value list screen in Ansible-Legacy console. If variable is not used, then configuration is not required.

Please refer to "[5.3.11 Substitution value list](#)" for details.

**⑪ Operation execution**

Select and set execution date, input operation and indicate operation execution from the execution screen of Ansible-Legacy console.

Please refer to "[5.3.14 Execution](#)" for details.

**⑫ Check operation status**

The status of executed operation is displayed in real time in the "Check operation status" screen of Ansible-Legacy console. In addition, users can perform emergency stop on operation and monitor the execution log and error log.

Please refer to "[5.3.12 Check operation status](#)" for details.

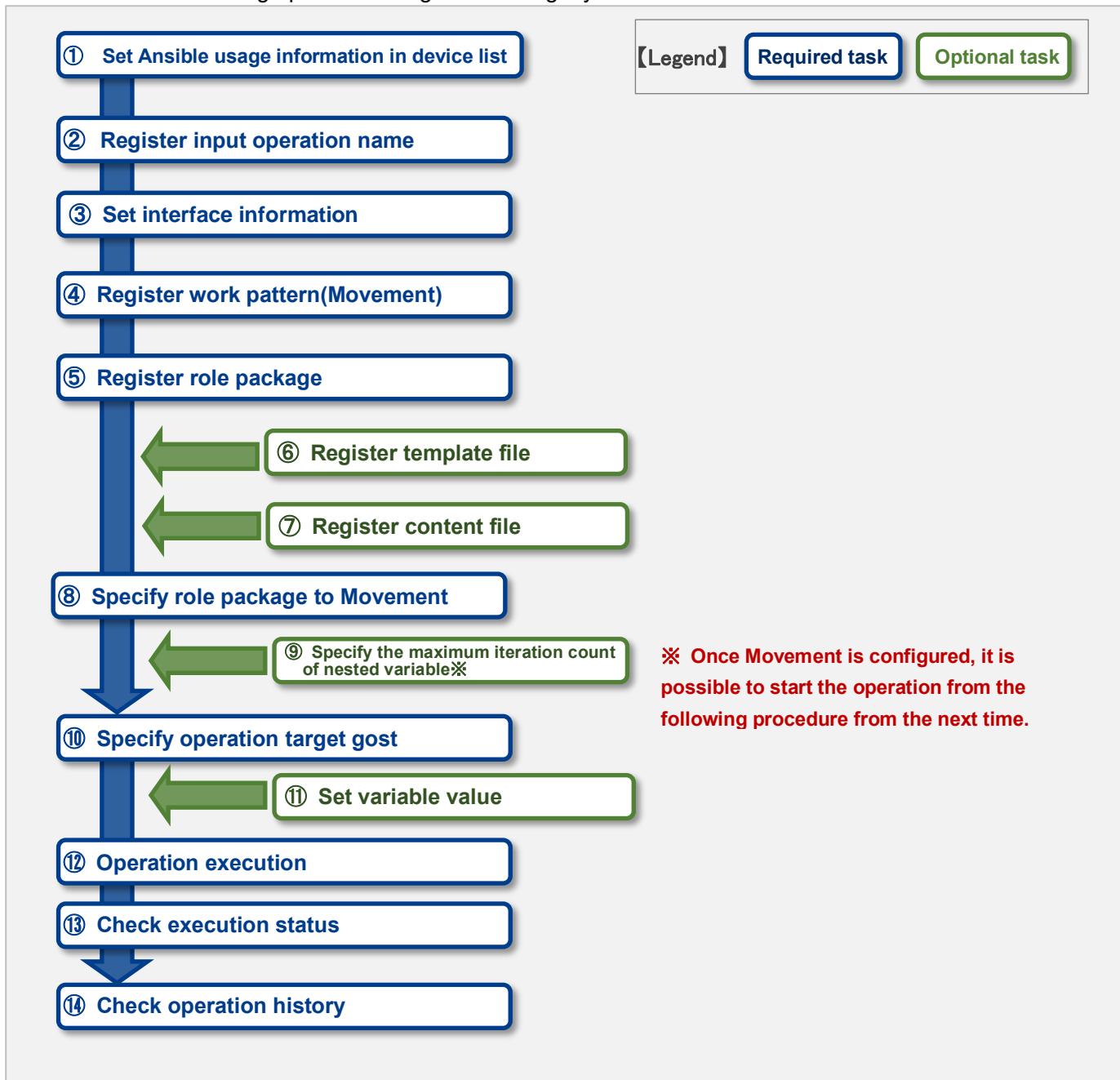
**⑬ Check operation history**

The list of executed operation is displayed in the execution list screen of Ansible-Legacy console and users can check the execution history.

Please refer to "[5.3.13 Execution list](#)" for details.

#### 4.1.2 Workflow of Ansible-Legacy Role

The workflow of executing operation using Ansible-Legacy Role is as follows.



- **Workflow details and references**

- ① **Set Ansible usage information in device list**

Set the Ansible usage information according to each devices in the device list screen of ITA basic console.

Please refer to “[5.1.1 Device list](#)” for details.

- ② **Register input operation name**

Register the input operation name from the input operation list screen of ITA basic console.

Please refer to “[5.1.2 Input operation list](#)” for details.

- ③ **Set interface information**

Select using whether Ansible or AnsibleTower server as the execution engine and register the connection information of the execution engine server from the interface information screen of Ansible common console.

Please refer to “[5.2.1 Interface information](#)” for details.

- ④ **Register work pattern (Movement)**

Register the Movement for operation from the Movement list screen of Ansible-Legacy Role console.

Please refer to “[5.3.2 Movement list](#)” for details.

- ⑤ **Register role package**

Register the role package used in operation from the role package list screen of Ansible-Legacy Role console.

Please refer to “[5.3.4 Role package list \(Ansible-Legacy Role only\)](#)” for details.

- ⑥ **Register template file (execute if needed)**

Register/Update/Discard the template file (src) and the template embedded variable used in the template module, etc. of Playbook from the template list screen of Ansible common console.

Please refer to “[5.2.4 Template list](#)” for details.

- ⑦ **Register content file (execute if needed)**

Register the file used to configure the operation target server from the contents list screen of Ansible common console.

Please refer to “[5.2.5 Contents list](#)” for details.

- ⑧ **Specify role package to Movement**

Specify the Playbook file to the registered Movement from Movement details screen of Ansible-Legacy Role console.

Please refer to “[5.3.7 Movement details](#)” for details.

- ⑨ **Specify the maximum iteration count of nested variable**

Specify the maximum iteration count of the array of member variables defined in nested variables from nested variable maximum iteration count list screen of the Ansible-Legacy Role console.

Please refer to “[5.3.8 Nested variable maximum iteration count list \(Ansible-Legacy Role only\)](#)” for details.

**⑩ Specify operation target host**

Specify the operation target host from the target host screen of Ansible-Legacy Role console.

Please refer to "[5.3.10 Target host](#)" for details.

**⑪ Set variable value (execute if needed)**

Set the value of the variable in the Playbook which has been registered to Movement from the substitution value list screen in Ansible-Legacy Role console. If variable is not used, then configuration is not required.

Please refer to "[5.3.11 Substitution value list](#)" for details.

**⑫ Operation execution**

Select and set execution date, input operation and instruct operation execution from the execution screen of Ansible-Legacy Role console.

Please refer to "[5.3.14 Execution](#)" for details.

**⑬ Check operation status**

The status of executed operation is displayed in real-time in the "Check operation status" screen of Ansible-Legacy console. In addition, users can perform emergency stop on operation and monitor the execution log and error log.

Please refer to "[5.3.12 Check operation status](#)" for details.

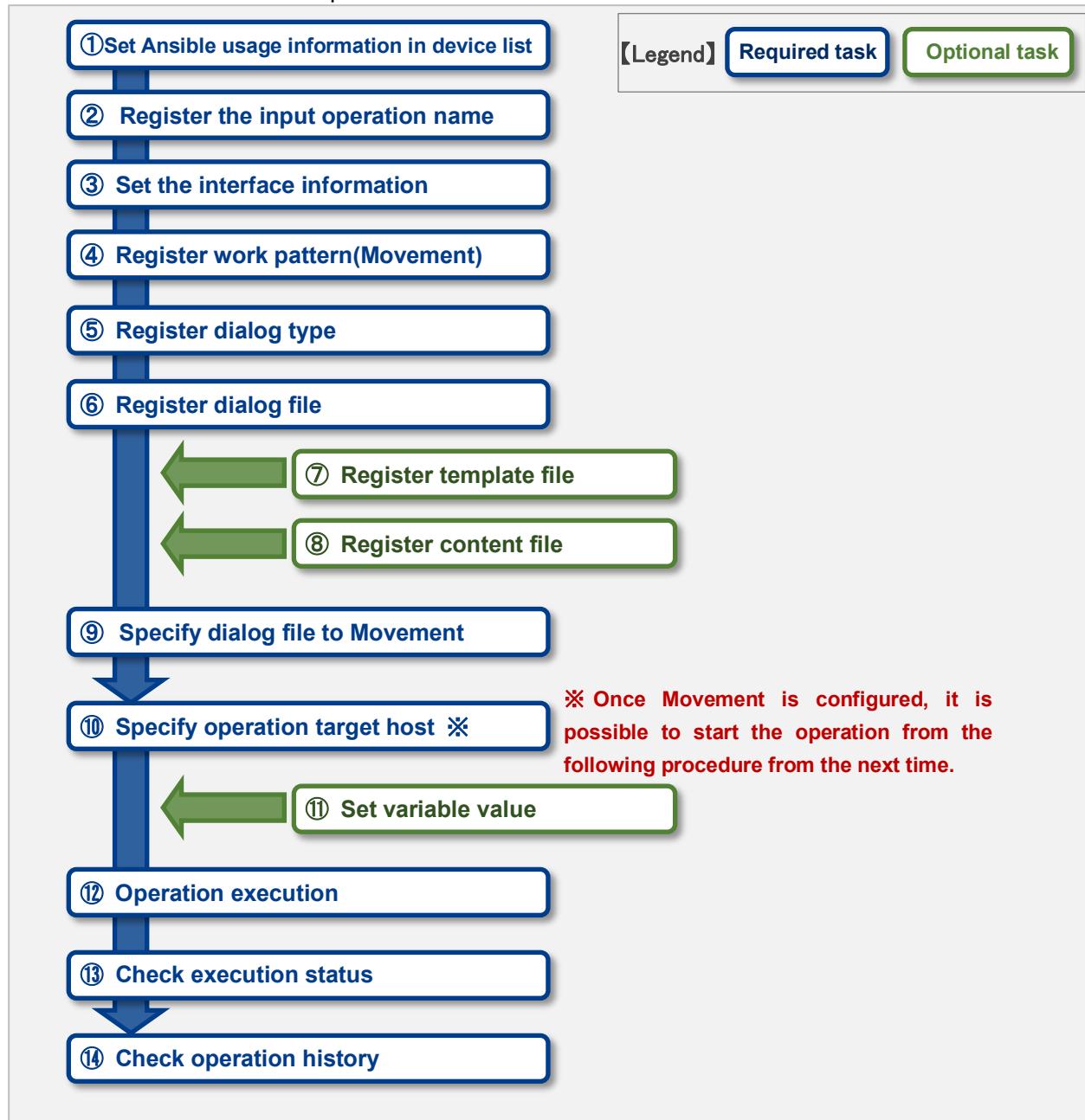
**⑭ Check operation history**

The list of executed operation is displayed in the execution list screen of Ansible-Legacy Role console and users can check the execution history.

Please refer to "[5.3.12 Execution list](#)" for details.

#### 4.1.3 Workflow of Ansible-Pioneer

The workflow to execute the operation in Ansible-Pioneer is as follows.



##### ● Workflow details and references

###### ① OS type registration

Set the OS type of the device to be operated from Pioneer.

###### ② Set Ansible usage information in device list

Set the Ansible usage information for each device from the device list screen of the ITA basic console.

Please refer to “[5.1.1 Device list](#)” for details.

###### ③ Register the input operation name

Register the input operation name from the input operation list screen of ITA basic console.  
Please refer to “[5.1.2 Input operation list](#)” for details.

**④ Register the interface information**

Select using whether Ansible or AnsibleTower server as the execution engine and register the connection information of the execution engine server from the interface information screen of Ansible common console.

Please refer to “[5.2.1 Interface information](#)” for details.

**⑤ Register work pattern (Movement)**

Register the Movement for operation from the Movement list screen of Ansible-Pioneer console.

Please refer to “[5.3.2 Movement list](#)” for details

**⑥ Register dialog type**

Register dialog type from the dialog type list screen of Ansible-Pioneer console.

Ansible-Pioneer defines the differences for each OS type in each dialog file, and combines the same purpose dialog file as dialog type to absorb (abstract) the device difference.

Please refer to “[5.3.5 Dialog type list \(Ansible-Pioneer only\)](#)” for details.

**⑦ Register dialog file**

Register dialog file according to the combination of dialog type and OS type from the dialog files screen of Ansible-Pioneer console.

Please refer to “[5.3.6 Dialog files \(Ansible-Pioneer only\)](#)” for details.

**⑧ Register template file (execute if needed)**

Register/Update/Discard the template file (src) and the template embedded variable used in the template module, etc. of Playbook from the template list screen of Ansible common console.

Please refer to “[5.3.4 Template list](#)” for details.

**⑨ Register content file (execute if needed)**

Register the file used to configure the operation target server from the contents list screen of Ansible common console.

Please refer to “[5.2.5 Contents list](#)” for details.

**⑩ Specify dialog file to Movement**

Specify dialog file to the registered Movement from movement details screen of Ansible-Legacy Role console.

Please refer to “[5.3.7 Movement details](#)” for details.

**⑪ Specify operation target host**

Specify the operation target host from the target host screen of Ansible-Pioneer console.

Please refer to “[5.3.10 Target host](#)” for details.

**⑫ Set variable value (execute if needed)**

Set the value of the variable in the Playbook which has been registered to Movement from the substitution value list screen in Ansible-Pioneer console. If variable is not used, then configuration is not required.

Please refer to “[5.3.11 Substitution value list](#)” for details.

### **(13) Operation execution**

Select and set execution date, input operation and indicate operation execution from the execution screen of Ansible-Pioneer console.

Please refer to "[5.3.14 Execution](#)" for details.

### **(14) Check operation status**

The status of executed operation is displayed in real time in the Check operation status screen of Ansible-Pioneer console. In addition, users can perform emergency stop on operation or monitor the execution log and error log.

Please refer to "[5.3.12 Check operation status](#)" for details.

### **(15) Check operation history**

The list of executed operation is displayed in the execution list screen of Ansible-Pioneer console and users can check the execution history.

Please refer to "[5.3.14 Execution list](#)" for details.

## **■Legend of Registration screen item list**

The content of the Registration screen item list are described in the next section.

① Item	② Description	③ Input required	④ Input type	⑤ Restrictions

#### **① Item**

- The item name in the submenu.

#### **② Description**

- The description for the item.

#### **③ Input required**

- ○: Items that entering contents are required for them.
- -: Items that entering contents are optional for them.

#### **④ Input type**

- Manual: Items that require manual input.
- Auto: Items whose content are entered automatically.
- Checkbox: Check box format item.
- Button: Radio button format item.
- List: List box format item.

#### **⑤ Restrictions**

- The restrictions for the item(Limitation on number of characters, etc.)

## 5 Ansible driver function · operation method explanation

This chapter explains each console function used in Ansible driver.

### 5.1 Basic console

This section describes the operation of ITA basic console.

Please refer to the ITA basic console manual for this operation and perform the operation in the ITA basic console screen.

#### 5.1.1 Device list

- (1) Register/Update/Discard information of operation target host is performed in the "Device list" menu.

This document explains the items (red frame) in the device list required for Ansible driver operations.

Please refer to "User instruction manual\_Basic console" for the description of other items.

The screenshot shows the Exastro IT Automation Basic Console interface. The left sidebar has a 'Menu' section with various options like Main menu, OS type master, Input operation list, Movement list, Symphony Interface information, Symphony class List, Symphony class editor, Symphony execution, Symphony execution checking, Symphony execution list, Export Symphony/Operation, and Contact administrator. The 'Device list' option is highlighted with a red box. The main content area has a title 'Basic Console' and a sub-section 'Device list'. It displays a table with columns: Discard, Managed system item number, HW device type, Host name, IP address, and MAC address. There are search dropdowns for each column. Below the table are 'Filter' and 'Clear filter' buttons, and an 'Auto-filter' checkbox. At the bottom, there are 'List/Update' and 'Register' buttons. The top right corner shows user info: User name [System Administrator], Login ID [administrator], Change password, and Logout.

Figure 5.1-1 Submenu screen (Device list)

- (2) Click the "Register" - "Start Registration" button to register the device information.

**Figure 5.1-2 Registration screen(Device list-common item)**

**Figure 5.1-3 Registration screen(Device list-Ansible usage information)**

- (3) The list of common item in registration screen is as follows.

Input of the columns with a red asterisk (\*) after their column name in the web screen is required.

In the case of using Ansible driver, please enter the usage information of Ansible.

If operation is executed while required column is not entered, unexpected errors may occur.

**Table 5.1-1 Registration screen item list (Device list)**

Item		Description		Input required	Input type	Restrictions
Managed system item number		A unique ID that identifies the registration information is entered automatically.		-	Auto	-
host name		Enter host name.		<input type="radio"/>	Manual	Maximum length 128 bytes
IP address		Enter IP address in xxx.xxx.xxx.xxx format.		<input type="radio"/>	Manual	Maximum length 15 bytes
EtherWake OnLan	MAC address	Enter MAC address.		-	Manual	Maximum length 17 bytes
	Network device name	Enter network device name.		-	Manual	Maximum length 256 bytes
Login user ID		Enter network device name.		<input type="radio"/>	Manual	Maximum length 30 bytes
Login password	Management	Select "●" when using ITA to manage password.		<input type="radio"/>	List	-
	Login password	Specify password.		<input type="radio"/>	Manual	Maximum length 30 bytes
ssh authentication key file		Specify the ssh authentication key file and enter the file when using key authentication. Required when specifying the ssh authentication key file if authentication method is the key authentication.		-	File	Maximum size 10K bytes

Item		Description	Input required	Input type	Restrictions	
Ansible dedicated information	Dedicated information for Legacy/Role	Authentication method	Select the authentication method when connecting the host from Ansible. • In the case of using password authentication, "●" is required to be selected in the "Management" column of Login password. • In the case of using key authentication, exchanging public key in advance is required. Also, it is required to set sudo privilege for login user in /etc/sudoers in advance. Exp) <code>Demo_user ALL=(ALL) NOPASSWD:ALL</code>	○	List	As described in the description column.
		Port no.	Enter the port number when connecting to WindowsServer with WinRm. If the column is left not entered, Ansible will connect to the default (5985) WinRM.	-	Manual	As described in the description column
		Server certificate	Enter the server certificate when https port number is specified as the WinRM connection port To omit server certificate authentication, add the following to the inventory file additional option. <code>ansible_winrm_server_cert_validation=ignore</code>	-	File	Maximum size 10K bytes
	Pioneer dedicated information	Protocol	Select the protocol (ssh/telnet) when logging into target device.	○	List	-
		OS type	Select the OS of target device. The OS types registered in the OS type master are displayed in list.	○	List	-
	Connection options		(In the case of connecting via ssh) If users want to set options other than the ssh options set in /etc/ansible.cfg/ssh_args , please enter the desired option. (In the case of connecting via telnet) If users want to set options when connecting via telnet, please enter the desired option.	-	Manual	Maximum length 512 bytes
	Inventory file addition option		Enter the option parameter of inventory file that is not set in ITA. e.g.) <code>ansible_connection: network_cli</code> <code>ansible_network_os: nxos</code>	-	Manual	Maximum length 512 bytes

Item		Description	Input required	Input type	Restrictions
Tower dedicated information	Instance group name	If AnsibleTower is in cluster configuration, select which instance group that the AnsibleTower belongs to(※1) The settings here are specified to the inventory of Tower(※2) The column will be default("Tower") of Ansible tower if not selected If AnsibleTower is not in cluster configuration, the column can be left unselected.	<input type="radio"/>	List	-
	Connection type	Set the connection type for Ansible tower authentication cedentials. Normally Machine is selected.In the case where Ansible_Connection needs to bet to local Network OS, Choose Network. If a Network is selected the user must set Platform Options other than (ansible_cnnection) for the additional inventory file options Exp) Example of Inventory file addition option settings. Set value when Network OS is ios.  ansible_network_os: ios ansible_become: yes ansible_become_method: enable  For more information about Ansible tower Ansible Tower credentials connection type, please refer to Tower Document credential type. Type for the connection type of Ansible Tower credential. For the relationship between Network OS and ansible_connection and Platform Options, refer to the Ansible document <a href="#">Platform Options</a> .	<input type="radio"/>	List	
Remarks		Free description field.	-	Manual	Maximum length 4000 bytes

※1 Although the column is an input item in basic console, it is selected among the data obtain by the Ansible driver backyard function "Tower server data association"

※2 The instance group is specified in the Tower inventory object.

### 5.1.2 Input operation list

- (1) In the "Input operation list" screen, the operations for the target host to be executed by the orchestrator are managed. Operations are selected from the menu in ITA basic console

The screenshot shows the Exastro IT Automation Basic Console interface. At the top, there is a header bar with the Exastro logo, the text 'Basic Console', and user information ('User name [System Administrator]', 'Login ID [administrator]', 'Change password', 'Logout'). On the left, a vertical sidebar titled 'Menu' lists various options: Main menu, OS type master, Device list, Input operation list (which is highlighted with a red box), Movement list, Symphony Interface information, Symphony class List, Symphony class editor, Symphony execution, Symphony execution checking, Symphony execution list, Export Symphony/Operation, and Contact administrator. The main content area has a title 'Description' and a 'Display filter' section. The 'Display filter' section contains fields for 'Discard' (set to 'Exclude discarded records'), 'No.', 'Operation ID', 'Operation name', and 'Scheduled date for execution'. Below these fields are 'Filter' and 'Clear filter' buttons, and a checked 'Auto-filter' checkbox. At the bottom of the main content area, there are four buttons: 'List/Update' (with 'Open' link), 'Register' (with 'Open' link), 'Download all and edit file uploads' (with 'Open' link), and 'Trace history' (with 'Open' link). A footer bar at the bottom of the page also has a 'Contact administrator' button.

Figure 5.1-4 Submenu screen (Input operation list)

Please refer to the related manual "User instruction manual\_Basic console" for the details of registration method.

## 5.2 Ansible common console

This section describes the operation of Ansible common console.

### 5.2.1 Interface information

- (1) In the "interface information" menu, select using whether Ansible or AnsibleTower server as the execution engine and register/update/discard the shared directory path between ITA system, Ansible driver server, and execution engine server , and the connection interface information of the execution engine server.

Update	Discard	Item number	Host	Protocol	Last update date/time	Last updated by	
<a href="#">Update</a>		1 exastro-it-automation-v1-3-0-en	https	443	Ansible	2020/01/28 10:59:30	System Administrator

Figure 5.2-1 Submenu screen (interface information)

- (2) Click the "List" - "Update" button to register the interface information.

Host*	Protocol*	Port*	execution engine*	Data relay storage path (ITA)*	Data relay storage path
exastro-it-automation-v1-3-0-en	https	443	Ansible	/exastro/data_re	/exastro/data_re

Figure 5.2-2 Registration screen (Interface information)

- (3) The item list of interface information screen is as follows.

If operation is executed while interface information not registered or multiple information is registered, **unexpected errors may occur**.

Table 5.2-1 Registration screen item list (Interface information)

Item	Description	Input required	Input type	Restrictions
Execution engine	Select the execution engine between Ansible and AnsibleTower. When AnsibleTower is selected, in order to execute ansible-vault command, Ansible Engine interface is also needed.	<input type="radio"/>	List	
Ansible Engine interface	Host	Enter the host name (or IP address) of Ansible server. It is recommended to enter host name when using HTTPS communication.	<input type="radio"/>	Manual Maximum length 128 bytes
	Protocol	Enter either http/https as the protocol with Ansible, AnsibleTower server.	<input type="radio"/>	Manual -
	Port	Enter the connection port (80/443) of Ansible, AnsibleTower. The port is usually HTTPS (443).	<input type="radio"/>	Manual -
	Execution user	Enter the execution user to execute ansible-playbook/ansible-vault command with sudo.	-	Manual Maximum length 64 bytes
	ACCESS_KEY_ID	Enter the access key used for authentication when connecting to the Ansible server.	-	Manual Maximum length 64 bytes
	SECRET_ACCE_SS_KEY	Enter the secret access key used for authentication when connecting to the Ansible server.	-	Manual Maximum length 64 bytes
Ansible Tower Interface	Host	Enter the host name (or IP address) of the execution engine (Ansible, AnsibleTower server). It is recommended to enter host name when using HTTPS communication. In addition, if AnsibleTower is constructed in cluster configuration, please enter the hostname (or IP address) where AnsibleTower is installed.	<input type="radio"/>	Manual Maximum length 128 bytes Required when the execution engine is AnsibleTower
	Protocol	Enter either http/https as the protocol with Ansible, AnsibleTower server.	<input type="radio"/>	Manual Required when the execution engine is AnsibleTower
	Port	Enter the connection port (80/443) of Ansible, AnsibleTower. The port is usually HTTPS (443).	<input type="radio"/>	Manual Required when the execution engine is AnsibleTower
	Organization name	Enter the organization name registered in AnsibleTower.	-	List Required when the execution engine is AnsibleTower
	Authentication token	Enter the user authentication token when connecting AnsibleTower server from ITA.	-	Manual Maximum length 128 bytes. Required when the execution engine is AnsibleTower

Item		Description	Input required	Input type	Restrictions
	Delete runtime data	Select whether to delete the data automatically generated by AnsibleTower during operation execution after operation is done. Select "Delete" from the pulldown list to delete.	-	List	Required when the execution engine is AnsibleTower
	Data relay storage path (ITA) ※1	Enter the directory viewed from the ITA system / Ansible driver server.	<input type="radio"/>	Manual	Maximum length 128 bytes
	Data relay storage path (Ansible/Ansible Tower)	Enter the directory viewed from the Ansible RestAPI and AnsibleTower servers.	<input type="radio"/>	Manual	Maximum length 128 bytes
	Symphony instance data relay storage path (Ansible/Ansible Tower)	Enter the directory which shares the shared directory between each movement when executing Symphony with Ansible RestAPI, AnsibleTower server.  The path viewed from the ITA system is set from the Symphony interface information menu.  Please refer to the "User instruction manual_ITA basic console" for the Symphony interface information.	<input type="radio"/>	Manual	Maximum length 128 bytes
	Conductor instance data relay storage path (Ansible/Ansible Tower)	Enter the directory which shares the shared directory between each movement when executing Conductor with Ansible RestAPI, AnsibleTower server.  The path viewed from the ITA system is set from the Conductor interface information menu.  Please refer to the "User instruction manual_ITA basic console" for the Conductor interface information.	<input type="radio"/>	Manual	Maximum length 128 bytes
	Optional parameter	Enter the Movement-common optional parameter of Ansible-Playbook command. Movement-specific optional parameters are entered in the Movement list menu.  In the case that the execution engine is Ansible:  Enter the optional parameter of Ansible-Playbook command. The -i option is set by ITA.  In the case that the execution engine is AnsibleTower:  The following option parameters can be set: -verbosity -f FORKS,--forks=FORKS -I SUBSET,--limit=SUBSET -e EXTRA_VARS,--extra-	-	Manual	Maximum length 512 bytes  Exastro1.2 does not support job slice count "-jsc, -job_slice_count="

Item	Description	Input required	Input type	Restrictions
	<p>vars=EXTRA_VARS  EXTRA_VARS: Variable name=specific value Variable name=specific value.....</p> <ul style="list-style-type: none"> <li>-t TAGS,--tags=TAGS</li> <li>-b,--become</li> <li>-D,--diff</li> <li>--skip-tags=SKIP_TAGS</li> <li>--start-at-task=START_AT_TASK</li> </ul> <p>The original optional parameters of AnsibleTower are as follows.</p> <ul style="list-style-type: none"> <li>-ufc,--use_fact_cache use fact cache</li> <li>-as,--allow_simultaneous enable simultaneous job execution</li> <li>-jsc,--job_slice_count= job slice count</li> </ul> <p>For the original optional parameters of AnsibleTower, please refer to the description of job template in the Ansible Tower user guide.</p>			
Number of parallel executions	Enter the maximum numbers of Movement (Legacy/Pioneer/Legacy-Role) that can be executed at the same time.	<input type="radio"/>	Manual	
Status monitoring cycle(milliseconds)	Enter the refresh interval of the log displayed in "5.3.12 Check operation status". Usually the value around 3000 milliseconds is recommended.	<input type="radio"/>	Manual	Minimum value 1000 milliseconds
Number of rows to display progress status	Enter the maximum display line count of the execution log, errorlog in "5.3.12 Check operation status". Usually the value around 1000 lines is recommended.	<input type="radio"/>	Manual	-
NULL link	<p>Set whether to register NULL (blank) value to substitution value list menu if the specific value in parameter sheet is NULL (blank) in the substitution value auto-registration setting menu.</p> <p>This value will be applied when "NULL link" in the substitution value auto-registration setting menu is blank.</p> <ul style="list-style-type: none"> <li>• If the "Valid" is set, any value in the parameter sheet will be registered in the substitution value list menu. (NULL value will be registered)</li> <li>• If the "Invalid" is set, only specific value in the parameter sheet will be registered in the substitution value list menu(NULL value will not be registered)</li> </ul>	<input type="radio"/>	List	-
Remarks	Free description field	-	Manual	Maximum length 4000 bytes

※1 Data relay storage paths are managed separately because directory path name may be different when operated on different servers.

Please refer to "System Configuration/Environment Construction Guide-Ansible-driver" for details.

## 5.2.2 Ansible Tower host list

In [Ansible Tower Host List], register/update/abolish the information required to execute Rest API of Ansible Tower and the information required to transfer the construction materials to Ansible Tower. If Ansible Tower is built in a cluster configuration, it is necessary to register all host information in the cluster. If it is not a cluster configuration, register the host information of the Ansible Tower (1 unit) to be corresponded.

**Figure 5.2-3 Submenu screen (Ansible Tower host list)**

(1) Click the "List"- "Update" button to register the Ansible Tower host information.

**Figure 5.2-4 Registration screen (Ansible Tower host)**

(2) The list of items on the Ansible Tower host list screen is as follows.

**Table 5.2-2 Registration screen item list (Ansible Tower host list)**

Item	Description	Input required	Input type	Restrictions
Host	Enter the host name (or IP address) of the Ansible Tower server. The host name is recommended for HTTPS communication.	<input type="radio"/>	Manual	Maximum length 128 bytes
Authentication method	Select the authentication method when connecting to the Ansible Tower server by file transfer (scp).	<input type="radio"/>	Manual	Maximum length 30 bytes
Login user	Enter the login user for connecting to the Ansible Tower server via file transfer (scp). Set and use a password for the login user and the awx user generated when installing Ansible Tower.	<input type="radio"/>	Manual	Maximum length 30 bytes
Password	This is required when password authentication is selected as the authentication method. Specify the password of the login user.	—	Manual	Maximum length 30 bytes
ssh authentication key file	If you select key authentication as the authentication method, enter the file for key authentication by specifying the ssh authentication key file.	—	File	Maximum size 10K bytes
isolated Tower	Select [●] for isolated Tower when it is built in a cluster configuration.	—	Select method	
Remarks	Free description field.	—	Manual	Maximum length 4000 bytes

### 5.2.3 Global variable list

- (1) In the "Global variable list" menu, register/update/discard the global variable name used in Playbook, dialog files, etc.

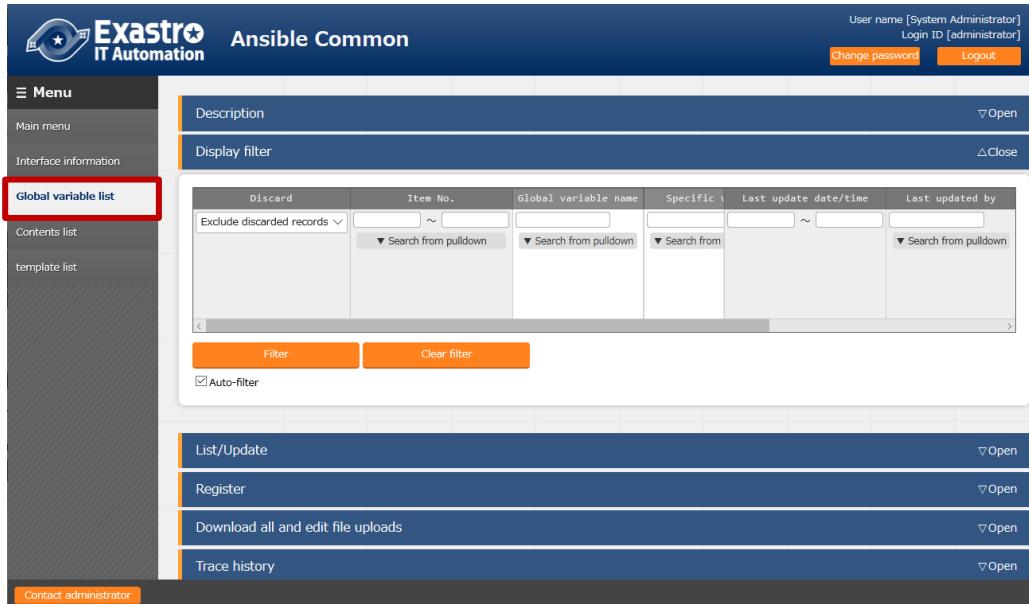


Figure 5.2-5 Submenu screen (Global variable list)

- (2) Click the "Register" - "Start Registration" button to register the operation information.

This is a registration form for a global variable. The title bar says 'Register'. The form has a table with columns: Item No., Global variable name\*, Specific value\*, Variable name description, and Remarks. The 'Item No.' column has a note 'Auto-input'. The 'Global variable name\*' and 'Specific value\*' columns have red asterisks indicating they are required. The 'Variable name description' and 'Remarks' columns are empty. At the bottom are 'Back' and 'Register' buttons. A note at the bottom left says '※\* is a required item.'

Figure 5.2-6 Registration screen (Global variable list)

(3) The item list of global variable list screen is as follows.

**Table 5.2-3 Registration screen item list (Global variable list)**

Item	Description	Input required	Input type	Restrictions
Global variable name	Enter the variable name. Enter the variable name in the "GBL_****" format. Half-width alphanumeric character and underscore (_) can be used. (Minimum length: 1 byte, maximum length: 128 bytes)	<input type="radio"/>	Manual	As described in the description column.
specific value	Enter the specific value File embedded variable "CPF_" and template embedded variable "TPF_" can be entered in the specific value column. When describing the variables, enclose the variable names with {{}} as describing the variables in the Playbook. e.g. ) Entering TPF_sample for specific value '{{△TPF_sample△}}' △: half-width space ' : recommended	<input type="radio"/>	Manual	Maximum length 8192 bytes
Variable name discription	Enter the description or comment of the variable.	-	Manual	Maximum length 256 bytes
Remarks	free description field	-	Manual	Maximum length 4000 bytes

#### 5.2.4 Template list

- (1) In the "template list" menu, register/update/discard the Jinja2 template file and the template embedded variable used in the parameter of template module and ios\_config module, etc. defined in the Playbook.

If template module is registered in the template list, the template file used in the template module etc. defined in the playbook can be specified by template embedded variable.

**Figure 5.2-7 Submenu screen(Template list)**

- (2) Click the "Register" - "Start Registration" button to register the file management information.

Template ID	Template embedded variable name*	Template files*	Variable definition
Auto-input	<input type="text"/>	<input type="button" value="Choose File"/> No file chosen <input type="button" value="Upload in advance"/> <small>Upload status:</small>	<input type="text"/>

**Figure 5.2-8 Registration screen (Template list)**

(3) The items of registration screen are as follows.

**Table 5.2-4 Registration screen item list (template list)**

Item	Description	Input required	Input type	Restrictions
Template embedded variable name	Enter the variable name embedded in parameters such as template module or ios_config module, etc. Enter the variable name in the "TPF_****" format. Half-width alphanumeric character and underscore (_) can be used.(Minimum length: 1 byte, maximum length: 128 bytes)	<input type="radio"/>	Manual	As described in the description column.
Template files	Upload the Jinja2 template file used as the parameter of module.	<input type="radio"/>	File	Text format
Variable definition	Define the variable used in the template file. If the template is used only in Ansible-Role and the variable is defined in the default variable definition file, then the variable definition column can be omitted. If the template is used only in Ansible-Role and the variable is defined in the default variable definition file, then the variable definition column can be omitted. If the variable with same name is used in multiple template, the variable definitions have to match. Error will occur during registration if the variable definitions do not match. Although the variable definition is based on the specification of Ansible, there is own specification of ITA.The notes of variable definition is described in 5.2-5-1.	-	Manual	Maximum length 4000 bytes
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

Please "Upload in advance (①)" the "template files" before "register".

Please click the "Register" button after checking the Playbook file name displayed in the "Upload status".

The screenshot shows a user interface for uploading template files. At the top, it says "Template files\*". Below that is a "Choose File" button with "sample.yml" selected. To the right of the file path is a red box containing the number ②, which points to a large orange button labeled "Upload in advance". Below the file path is another red box containing the number ①, which points to a section labeled "Upload status:" showing the message "Uploaded." and the file name "File name sample.yml Size128bytes".

**Table 5.2-5 Notes of variable definition**

Type	Notes
Normal variable	Specific value is optional. e.g.) VAR_sample_1: none VAR_sample_2:
Multiple specific value variable	Specific value is optional. e.g.) VAR_sample_1: △- none VAR_sample_2: [] Please enter 1 or more half-width space(△) before - when defining specific value. The variable definition maybe misinterpreted.
Nested variable	Specific value is required for the first variable of the nested structure. If the specific value is not defined, the variable definition maybe misinterpreted. Also, please enter 1 or more half-width space(△) before - when defining specific value. The variable definition maybe misinterpreted e.g.) VAR_sample_1: △- item1: none # The definition of specific value is required since it's the starting variable of the nested structure △ item2: # The definition of specific value is optional VAR_sample_2: △- array: △- item1: none # The definition of specific value is required since it's the starting variable of the nested structure △ item2: # The definition of specific value is optional The template with nested variable defined can only be used in Ansible-Role When used in Ansible-Role, if the variable with same name is defined in default variable definition file, etc., the definition of the variables have to match. If the definition of the variables do not match, an error will occur during registration.
Global variable	The definition of specific value is optional e.g.) GBL_sample_1: none GBL_sample_2:
ITA original variable	The definition of variable is not required.
substitution variable	The 3 kinds of variable that can be defined are as follows. • Normal variable • Multiple specific value variable • Nested variable The note of each variable definition are the same.  e.g.) LCA_sample_1: LCA_sample_2: [] LCA_sample_3:

	<p>△- item1: none                   <b># The definition of specific value is required since it's the starting variable of the nested structure</b></p> <p>△ item2:                           <b># The definition of specific value is optional</b></p> <p>The template with substitution variable defined can only be used in Ansible-Role</p>
--	--

For details, Please refer to the attachment "User Instruction Manual - Ansible-driver attachment- Ansible usage guideline with additional rules"

## ① Describe Playbook

When describing the template registered in template list menu in Playbook, describe the appropriate parameter in the template embedded variable name.

If the template embedded variable name is not used, describe the variable registered in the substitution list and the path of the file.

e.g. )

### Describe Playbook

- template: src='{{△TPF\_hosts△}}' dest=/etc/hosts  
△: half-width space

### Registration content

Template embedded variable name	Template file
TPF_hosts	/etc/hosts

**Please describe the file name in \_dest. If the file name is not specified, the work will be executed with the registered template file whose file name is added with the ITA management number in the front of the file name.**

**For example, in the case of dest=/etc/, the file name will be /etc/10-digit-number\_hosts**

## ② Describe dialog file

In the case of describing the dialog file, describe the template embedded variable name.

e.g. )

### Describe dialog file

- expect: '{{△\_loginuser\_△}}@{{△\_loginhostname\_△}}'  
exec: 'scp △ITA user@ITA host name:{{△TPF\_hosts△}}△forwarding destination'

- expect: 'password:'

exec: 'ITAuser password'  
△: half-width space

### Registration content

Template embedded variable name	Template file
TPF_hosts	/etc/hosts

**Please describe the file name in the forwarding destination.**

**If the file name is not specified, the work will be executed with the registered template file whose file name is added with the ITA management number in the front of the file name.**

**For example, in the case of forwarding destination=/etc/, the file name will be /etc/10-digit-number\_hosts**

**{{△TPF\_hosts△}} will be replaced by the absolute path during execution.**

By reading the variable definition of template with internal process, it is possible to register specific value in menu "5.3.9 Substitution value auto-registration setting" and menu "5.3.11 Substitution value list".

Since the timing of file reading is not in real time, it may take some time<sup>※1</sup> until the variables can be handled in menu "5.3.9 Substitution value auto-registration setting" and menu "5.3.11 Substitution value list".

**※1 The timing of file reading is described in " 7.2 About the maintenance method", so please refer to it.**

### 5.2.5 Contents list

- (1) In the "contents list" menu, register/update/discard the file and file embedded variable used in each module defined in the Playbook.

If the files are registered in the contents list, the file used in each module defined in the Playbook can be specified by file embedded variable.

**Figure 5.2-9 submenu screen (Contents list)**

- (2) Click the "Register" - "Start Registration" button to register the file management information.

File ID	File embedded variable name*	Files*	Remarks	Last update
Auto-input	<input type="text"/>	<input type="button" value="Choose File"/> No file chosen <input type="button" value="Upload in advance"/> <small>Upload status:</small>	<input type="text"/>	Auto-input

**Figure 5.2-10 Registration screen (Contents list)**

(3) The items of registration screen are as follows.

**Table 5.2-6 Registration screen item list (contents list)**

Item	Description	Input required	Input type	Restrictions
File embedded variable name	Enter the variable names to be embedded in the parameter of each module. Enter the variable name in the "CPF_*****" format. Half-width alphanumeric character and underscore(_) can be used.(Minimum length: 1 byte, maximum length: 128 bytes)	<input type="radio"/>	Manual	As described in the description column
Files	Upload the file used in each module.	<input type="radio"/>	File	Maximum size 20M bytes
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

Please "Upload in advance (①)" the "template files" before "register".

Please click the "Register" button after checking the Playbook file name displayed in the "Upload status".

The screenshot shows a user interface for uploading template files. At the top, there is a header labeled "Template files\*". Below it, a "Choose File" input field contains the value "sample.yml". To the right of the input field is a red-bordered button labeled "Upload in advance" with a circled "①" above it. Below the input field, a red-bordered box contains the "Upload status:" information: "Uploaded.", "File name sample.yml", and "Size128bytes". A circled "②" is placed next to the "File name" line.

## ① Describe Playbook

When describing each modules in the Playbook, describe the file embedded variable name.

e.g)

### Describe Playbook

-copy: src='{{△CPF\_hosts△}}' dest=/etc/hosts  
△: half-width space

### Registration content

File embedded variable name	Files
CPF_hosts	hosts

Please describe the file name for dest. If the file name is not specified, the work will be executed with the registered file whose file name is added with the ITA management number in the front of the file name.

For example, in the case of dest=/etc/, the file name will be /etc/10-digit-number\_hosts

-unarchive src='{{△CPF\_tool\_tgz△}}' dest=/usr/local/bin  
△: half-width space

File embedded variable name	Files
CPF_tool_tgz	tool.tgz

## ② Describe dialog file

In the case of describing the dialog file, describe the file embedded variable name.

e.g)

### Describe dialog file

### Registration content

- expect: '{{△\_\_loginuser\_△}}@{{△\_\_loginhostname\_△}}'  
exec: 'scp △ITA user@ITA host name:{{△CPF\_hosts△}}△forwarding destination'  
- expect: 'password:'  
exec: 'exec: ITAuser password'  
△:half-width space

File embedded variable name	Files
CPF_hosts	hosts

Please describe the file name in the forwarding destination.

If the file name is not specified, the work will be executed with the registered file whose file name is added with the ITA management number in the front of the file name.

For example, in the case of forwarding destination=/etc/, the file name will be /etc/10-digit-number\_hosts

`△CPF_hosts△` will be replaced by the absolute path of forwarding origin during execution.

## 5.3 Ansible-Legacy／Legacy Role／Pioneer console

The operation of Ansible-Legacy／Legacy Role／Pioneer console.

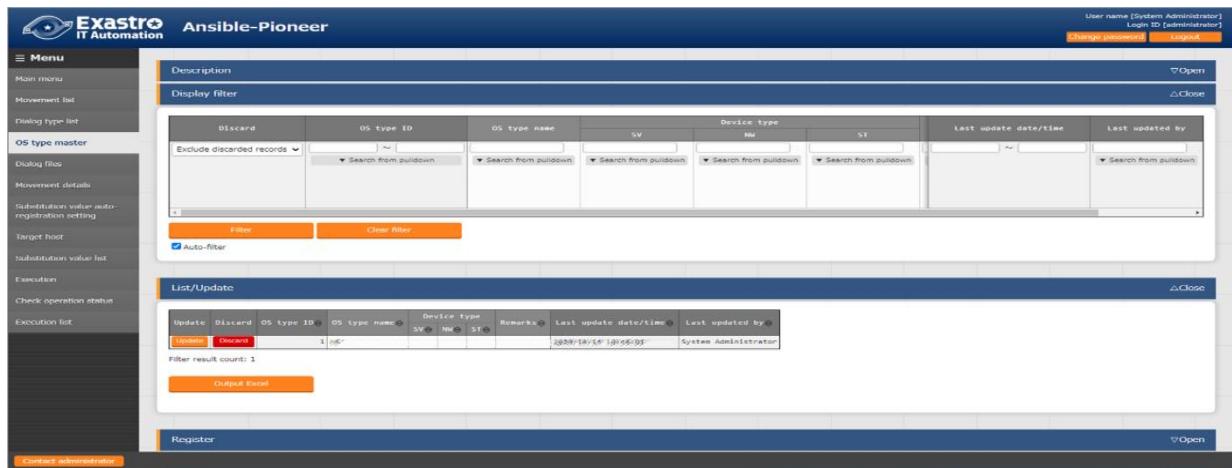
### 5.3.1 OS type master

- (1) On the [OS Type Master] screen, the OS type of the device to be operated is managed from Pioneer of ITA.  
※This menu exists only in the Ansible-Pioneer console.

The screenshot shows the Ansible-Pioneer interface. In the top right corner, there are links for 'User name [System Administrator]', 'Login ID [administrator]', 'Change password', and 'Logout'. The main content area has a title 'Ansible-Pioneer' with the Exastro logo. On the left, a vertical navigation menu lists: Main menu, Movement list, Dialog type list, OS type master (which is selected and highlighted in orange), Dialog files, Movement details, Substitution value auto-registration setting, Target host, Substitution value list, Execution, Check operation status, and Execution list. The central panel displays a table for registering OS types. The table has columns: OS type ID, OS type name\*, Device type (with sub-columns SV, NW, ST), Remarks, Last update date/time, and Last updated by. A note at the bottom of the table says '\* is a required item.' Below the table are 'Back' and 'Register' buttons. At the bottom of the central panel, there are links for 'Download all and edit file uploads' and 'Trace history'. The bottom of the page features a footer with a 'Contact administrator' link.

Figure 5.3-1 Submenu screen (OS type master)

- (2) Click the "Register"- "Start Registration" button to register the OS information.



**Figure 5.3-2 registration screen (OS type master)**

- (3) The list of items on the registration screen is as follows.

**Table 5.3-2 registration screen item list (OS type master)**

Item		Description	Input required	Input type	Restrictions
OS type ID		A unique ID that identifies the registration information will be automatically entered.	○	Auto	-
OS type name		Input the any device name.	○	Manual	Maximum length 256 bytes
Model	SV	If the device type is server, select "●"	-	List	-
	NW	If the device type is a network device, select "●"	-	List	-
	ST	If the device type is a storage device select "●"	-	List	-
Remarks		Free description field.	-	Manual	

### 5.3.2 Movement list

- (1) Register/Update/Discard Movement name in "Movement list"

The screenshot shows the Ansible-Legacy interface with the 'Movement list' option selected in the left sidebar menu. The main area displays two tabs: 'Description' (with an 'Open' button) and 'Display filter' (with a 'Close' button). Below these is a search/filter panel with fields for 'Discard', 'Movement ID', 'Movement Name', 'Orchestrator', 'Last update date/time', and 'Last updated by'. There are also dropdowns for 'Exclude discarded records', 'Search from pulldown' for Movement ID and Movement Name, and a 'Filter' button. A checkbox for 'Auto-filter' is checked. The bottom section is titled 'List/Update' (with a 'Close' button) and shows a table with columns: Update, Discard, Movement ID, Movement Name, Orchestrator, Delay timer, Host specific format, Last update date/time, and Last updated by. A single row is listed: 'Update' (orange button), 'Discard' (red button), '1 testing-one', 'Ansible Legacy', 'Host name', '2020/02/10 10:52:16', and 'System Administrator'. Navigation arrows are at the bottom of the table.

Figure 5.3-3 submenu screen (Movement list)

- (2) Click the "Register" - "Start Registration" button to register the Movement information.

The screenshot shows a registration form for Movement information. It includes fields for Movement ID (Auto-input), Movement Name (with a required asterisk), Delay timer, and a 'Dedicated information for ansible' section. This section contains 'Host specific format' (with a required asterisk), 'WinRM connection' (with a dropdown arrow), and 'Header section' (with a large text input field). Navigation arrows are at the bottom of the form.

Figure 5.3-4 Registration screen (Movement list)

(3) The list of registration screen items are as follows.

**Table 5.3-2 Registration screen item list (Movement list)**

Item	Description	Input required	Input type	Restrictions
Movement name	Enter the name of Movement	<input type="radio"/>	Manual	Maximum length 256 bytes
Delay timer	Enter the specified period (1~) if you want the warning of delay status to display when the scheduled time of Movement has delayed. (Unit:minute) The warning will not display if the column is not entered.	-	Manual	-
Host specific format	Select "Host name" if the user wants to specify the host that is not represented by an IP address. Normally IP is recommended	<input type="radio"/>	List	-
Number of parallel executions  ※ Only displayed in the Movement list of Pioneer.	Enter the number of target hosts that Ansible can execute simultaneously.  If there exists more target hosts than the configured value, after executing the amount of target hosts as the configured value, the remaining target hosts will be executed randomly.  ■ About the behavior when the column is not entered - In the case of Ansible driver, the content of configuration file(/etc/ansible.conf) in the server will be the default values. - In the case of AnsibleTower driver, the default value of AnsibleTower will be used.	-	Manual	NULL or Integer
WinRM connection *1	Select "●" if the target host is WindowsServer.	-	List	-
Header section  ※ Not displayed in the Movement list of Pioneer	Edit the parent Playbook automatically generated by ITA from the beginning to the tasks or roles section.  The following will be applied if the column is not entered.  Ansible: <pre>- hosts: all   remote_user: \${__loginuser__}   gather_facts: no   become: yes</pre>  Ansible Tower: <pre>- hosts: all   gather_facts: no   become: yes</pre>  ※ In case of connecting with winrm, become:yes can't be applied.	-	Manual	Maximum length 512 bytes

Optional parameter	<p>Enter the Movement-specific optional parameter of Ansible-Playbook command.</p> <p>In the case that the execution engine is Ansible:</p> <ul style="list-style-type: none"> <li>Enter the optional parameter of Ansible-Playbook command.</li> <li>The -i option is set by ITA</li> </ul> <p>In the case that the execution engine is Ansible Tower:</p> <ul style="list-style-type: none"> <li>The following option operator can be set</li> <li>-verbosity</li> <li>-f FORKS,--forks=FORKS</li> <li>-l SUBSET,--limit=SUBSET</li> <li>-e EXTRA_VARS,--extra-vars=EXTRA_VARS</li> <li>EXTRA_VARS: Variable name=specific value Variable name=specific value .....</li> <li>-t TAGS,--tags=TAGS</li> <li>-b,--become</li> <li>-D,--diff</li> <li>--skip-tags=SKIP_TAGS</li> <li>--start-at-task=START_AT_TASK</li> </ul> <p>The original optional parameters of AnsibleTower are as follows.</p> <ul style="list-style-type: none"> <li>-ufc,--use_fact_cache use fact cache</li> <li>-as,--allow_simultaneous enable simultaneous job execution</li> <li>-jsc,--job_slice_count= job slice count</li> </ul> <p>For the original optional parameters of AnsibleTower, please refer to the description of job template in the Ansible Tower user guide.</p>	-	Manual	Maximum length 256 bytes
<b>Virtualenv</b> <b>※Displayed when the execution engine is AnsibleTower</b>	Select the Ansible execution environment where virtualenv is constructed. The Ansible execution environment used when installing Tower will be used if this column is not selected.	-	List	
Remarks	free description field	-	Manual	Maximum length 4000 bytes

\*1 In the case other than WindowsServer, please be sure to select blank from the list. Also, Ansible-Pioneer is always operating with gather\_facts=false parameter.

**[Notes]**

**In the case of selecting "●" in the WinRM connection column, all connection hosts will be considered as WindowsServer.**

### 5.3.3 Playbook file list (Ansible-Legacy only)

- (1) Register/update/discard the Playbooks created by users in the "Playbook files" menu.  
※This menu only exists in the Ansible-Legacy console.  
Please refer to " 6.1 Describe Playbook (Ansible-Legacy)" about describing Playbook.

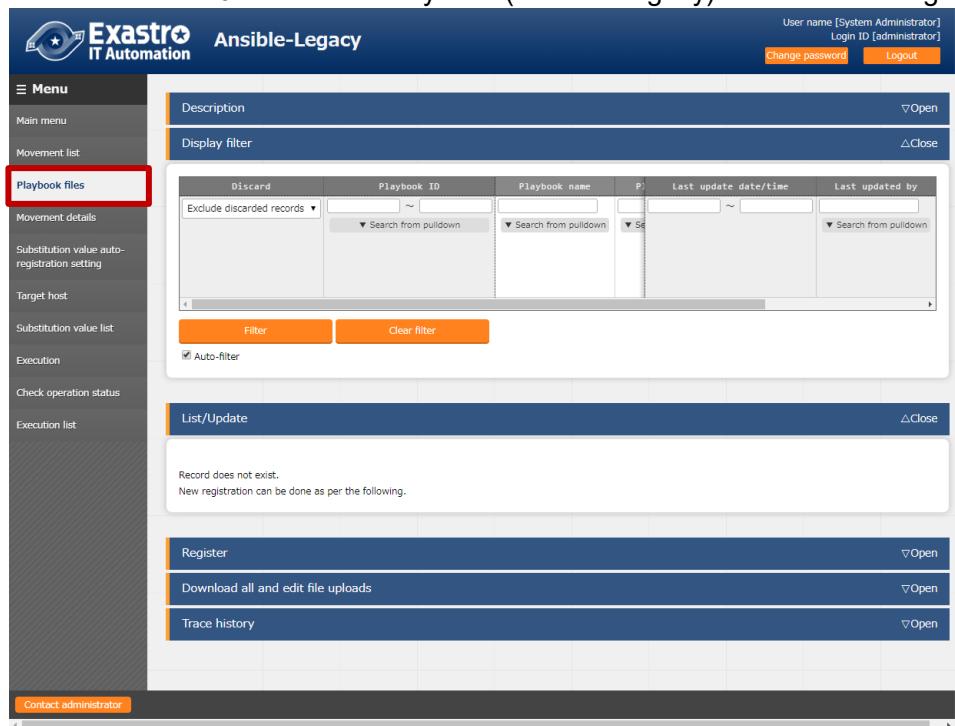


Figure 5.3-3 Submenu screen (Playbook files)

Click the "Register" - "Start Registration" button to register the Playbook.

The screenshot shows the 'Register' screen for Playbook files. It has fields for Playbook ID (Auto-input), Playbook name (required), and Playbook files (Choose File, Upload in advance, Upload status). There is also a Remarks field and a Last update date/time field set to Auto-input. At the bottom are Back and Register buttons.

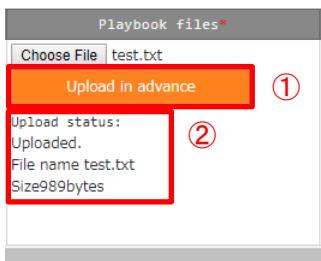
Figure 5.3-4 Registration screen (Playbook files)

(2) The list of registration screen items are as follows.

**Table 5.3-3 Registration screen item list (Playbook files)**

Item	Description	Input required	Input type	Restrictions
Playbook name	Enter the Playbook name to be managed in ITA.	<input type="radio"/>	Manual	Maximum length 256 bytes
Playbook files	Upload the created Playbook file. Please make sure that the playbook file is created with UTF-8 Code when uploading it. Playbook files other than UTF-8 without BOM will result in an upload error.	<input type="radio"/>	File	Maximum size 20M bytes
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

Please "Upload in advance (①)" the "Playbook files" before "register". Please click the "Register" button after checking the Playbook file name displayed in the "Upload status"



The internal process will extract the variables defined in Playbook files. Users can register specific value of the extracted variables in menu "[5.3.9 Substitution value auto-registration setting](#)" and menu "[5.3.11 Substitution value list](#)".

Since the timing of extraction is not in real time, it may take some time<sup>※1</sup> until the variables can be handled in menu "[5.3.9 Substitution value auto-registration setting](#)" and menu "[5.3.11 Substitution value list](#)".

**※1 The timing of extraction is described in " 7.2 About the maintenance method", so please refer to it.**

### 5.3.4 Role package list (Ansible-Legacy Role only)

- (1) Register/upload/discard the role package file created by the users.

※This menu only exists in Ansible-Legacy Role console.

Please compress the directory of the hierarchy level which contains "roles" into zip file and register role package file with the zip file.

Please refer to " 6.3 Describe role package (Ansible-Legacy Role)" for the structure of role package directory.

The screenshot shows the Ansible-LegacyRole interface. On the left, there is a sidebar with a menu. The 'Role package list' option is highlighted with a red box. The main content area has three tabs: 'Description' (with a 'Open' button), 'Display filter' (with a 'Close' button), and 'List/Update'. The 'List/Update' tab is active and displays a message: 'Record does not exist. New registration can be done as per the following.' Below this, there are three buttons: 'Register' (with a 'Open' button), 'Download all and edit file uploads' (with a 'Open' button), and 'Trace history' (with a 'Open' button). At the bottom of the sidebar, there is a 'Contact administrator' button.

**Figure 5.3-7 Submenu screen (Role package list)**

- (2) Click the "Register" - "Start Registration" button to register the Playbook.

The screenshot shows the 'Register' screen. It has fields for 'Item No.' (Auto-input), 'Role package name\*' (empty), and 'Role package file (ZIP format)\*'. The 'Role package file (ZIP format)\*' field contains a 'Choose File' button with the message 'No file chosen' and an 'Upload in advance' button. There is also an 'Upload status:' field. To the right, there are fields for 'Remarks' (empty) and 'Last update date/tir' (Auto-input). At the bottom, there are 'Back' and 'Register' buttons. A note at the bottom left says '※ \* is a required item.'

**Figure 5.3-8 Registration screen (Role package list)**

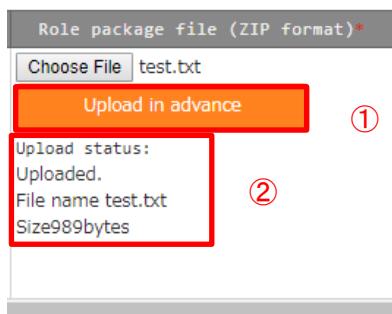
(3) The list of registration screen items are as follows.

**Table 5.3-4 Registration screen item list (Role package list)**

Item	Description	Input required	Input type	Restrictions
Role package name	Enter the role package name to be managed in ITA.	<input type="radio"/>	Manual	Maximum length 256 bytes
Role package file	Upload the created role package file (zip format). Please make sure that the role package file is created with UTF-8 Code and without BOM when uploading it. If a playbook file other than UTF-8 without BOM is included, an error will occur during registration. For details, please refer to the 6.3 Role package (Ansible-Legacy Role).	<input type="radio"/>	File	Maximum size 20M bytes
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

Please "Upload in advance (①)" the "Role package file" before "register".

Please click the "Register" button after checking the role package file name displayed in the "Upload status".



The internal process will extract the variables defined in Role package files. Users can register specific value of the extracted variables in menu "5.3.9 Substitution value auto-registration setting" and menu "5.3.11 Substitution value list".

Since the timing of extraction is not in real time, it may take some time<sup>※1</sup> until the variables can be handled in menu "5.3.9 Substitution value auto-registration setting" and menu "5.3.11 Substitution value list".

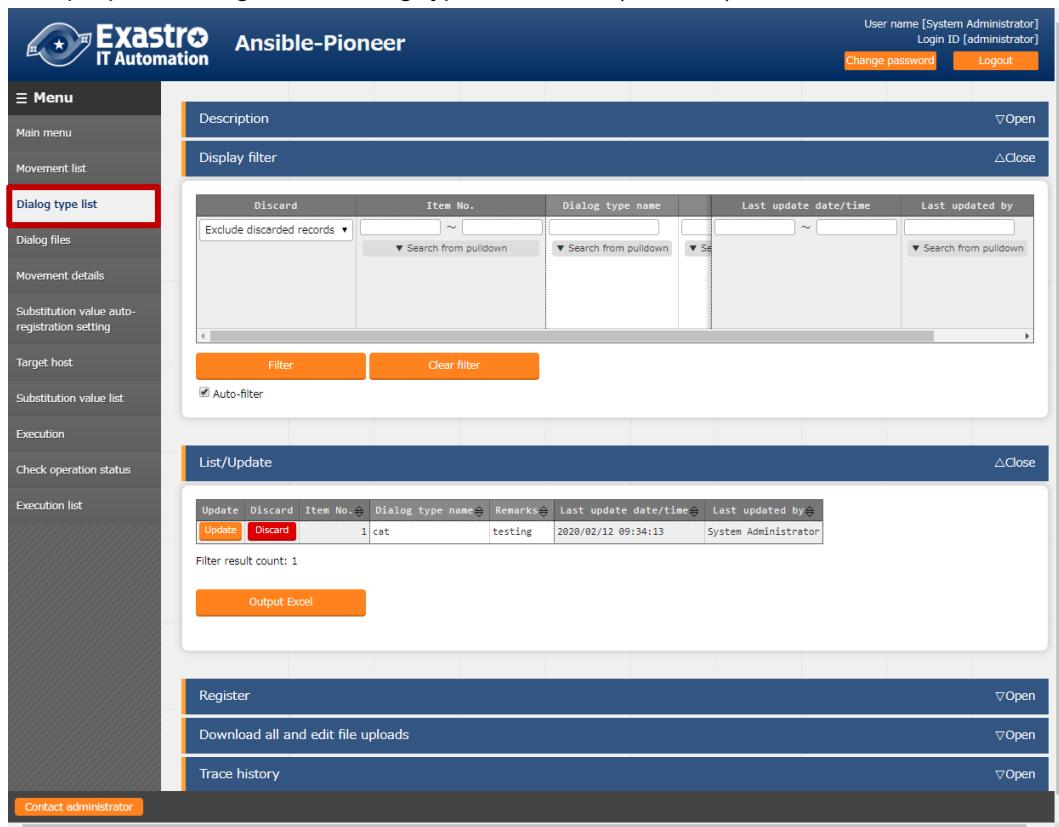
**※1 The timing of extraction is described in " 7.2 About the maintenance method", so please refer to it.**

### 5.3.5 Dialog type list (Ansible-Pioneer only)

- (1) Register/update/discard dialog type in the "dialog type list" menu

This menu only exists in the Ansible-Pioneer console

Ansible-Pioneer defines the differences for each OS type in each dialog file, and combines the same purpose dialog file as dialog type to remove (abstract) the device difference.



**Figure 5.3-9 Submenu screen (dialog type list)**

- (2) Click the "Register" - "Start Registration" button to register the operation information.

Item No.	Dialog type name*	Remarks	Last update date/time	Last updated by
Auto-input			Auto-input	Auto-input

**Figure 5.3-10 Registration screen (Dialog type list)**

- (3) The list of registration screen items are as follows.

**Table 5.3-5 Registration screen item list (Dialog type list)**

Item	Description	Input required	Input type	Restrictions
Dialog type name	Enter the name of dialog type	○	Manual	Maximum length 256 bytes
Remarks	Free description field	-	Manual	Maximum length 4000 bytes

### 5.3.6 Dialog files (Ansible-Pioneer only)

- (1) Register/update/discard the dialog file created by users in "dialog files" menu.  
※This menu only exists in Ansible-Pioneer console.
- (2) Please refer to "[6.2 Describe Dialog file \(Ansible-Pioneer\)](#)" for describing the dialog file, etc.  
Register dialog files for each combination of dialog type and OS type.  
Please register dialog file of each "OS type" with the same "dialog type" in the case of supporting multiple OS types with one "dialog type".

**Figure 5.3-11 Submenu screen (dialog files)**

- (3) Click the "Register" - "Start Registration" button to register the dialog files.

**Figure 5.3-12 Registration screen (dialog files)**

(4) The list of registration screen items are as follows.

**Table 5.3-6Registration screen item list (Dialog files)**

Item	Description	Input required	Input type	Restrictions
Dialog type	The dialog type registered in the dialog type list menu will be displayed. Select the dialog type of dialog file to be registered.	<input type="radio"/>	List	-
OS type	The OS type registered in the OS type master menu will be displayed. Select the OS type of dialog file to be registered.	<input type="radio"/>	List	-
Dialog file	Upload the dialog file according to the dialog type and OS type.	<input type="radio"/>	File	Maximum size 20M bytes
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

Please "Upload in advance (①)" the "dialog file" before "register".

Please click the "Register" button after checking the dialog file name displayed in the "Upload status".

Detailed description: The screenshot shows a user interface for uploading a file. At the top, there's a label 'Dialog file\*' followed by a 'Choose File' button with the path 'test.txt'. Below this is a red-bordered button labeled 'Upload in advance' with a circled ① above it. Further down is a red-bordered box containing the 'Upload status' information with a circled ② above it. The status box includes 'Uploaded.', 'File name test.txt', and 'Size989bytes'.

The internal process will extract the variables defined in dialog files. Users can register specific value of the extracted variables in menu "[5.3.9 Substitution value auto-registration setting](#)" and menu "[5.3.11 Substitution value list](#)".

Since the timing of extraction is not in real time, it may take some time※1 until the variables can be handled in menu "[5.3.9 Substitution value auto-registration setting](#)" and menu "[5.3.11 Substitution value list](#)".

※1 The timing of extraction is described in " 7.2 About the maintenance method", so please refer to it.

### 5.3.7 Movement details

- (1) Register/update/discard the files executed in the Movement in "Movement details" menu.

Discard	Associated item No.	Movement	Playbook files	Include order	Remark	Last update date/time	Last updated by
Exclude discarded records	~	▼ Search from pulldown					

**List/Update**

Update	Discard	Associated item No.	Movement	Playbook files	Include order	Remark	Last update date/time	Last updated by
Update	Discard	1 1:testing-one	create_dir		1		2020/02/10 11:44:53	System Administrator

Filter result count: 1

**Register**

Download all and edit file uploads

Trace history

**Figure 5.3-13 Submenu screen (Movement details)**

※The screen is from Ansible Legacy.

- (2) Click the "Register" - "Start Registration" button to register the details of Movement.

Associated item No.*	Movement*	Playbook files*	Include order*	Remarks
Auto-input	▼	▼	▼	▼

※\* is a required item.

**Table setting**

**Register**

**Figure 5.3-14 Registration screen (Movement details)**

(3) The list of registration screen items are as follows.

- In Ansible-Legacy console

**Table 5.3-7 Registration screen item list (Movement details in Ansible-Legacy console)**

Item	Description	Input required	Input type	Restrictions
Movement	The Movement registered in the Movement list will be displayed. Select the Movement.	<input type="radio"/>	List	-
Playbook file	The Playbook file registered in "5.3.3 Playbook file list (Ansible-Legacy only)" will be displayed. Select the Playbook file.	<input type="radio"/>	List	-
Include order	Enter the execution order of playbook files (unique value starts from 1). Playbook files will be executed as the entered include order (ascending).	<input type="radio"/>	Manual	Half-width integer
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

- In Ansible-Legacy Role console

**Table 5.3-8 Registration screen item list (Movement details in Ansible-Legacy Role console)**

Item	Description	Input required	Input type	Restrictions
Movement	Same as Ansible-Legacy	<input checked="" type="radio"/>	List	-
Role package name	The role package registered in the role package list menu will be displayed. Select the role package to be executed.	<input checked="" type="radio"/>	List	-
Role name	The role names included in role package selected in role package name are displayed. Select the role in the role package to be executed.	<input checked="" type="radio"/>		-
Include order	Same as Ansible-Legacy	<input checked="" type="radio"/>	Manual	Half-width integer
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

**[Notes]**

Please do not register one Movement to multiple role packages.

If Movement is executed with multiple role packages registered, unexpected error will occur.

- In Ansible-Pioneer console

**Table 5.3-9 Registration screen item list (Movement details in Ansible-Pioneer console)**

Item	Description	Input required	Input type	Restrictions
Movement	Same as Ansible-Legacy.	<input checked="" type="radio"/>	List	-
Dialog type	The dialog type registered in "5.3.6 Dialog files" (Ansible-Pioneer only) will be displayed. Select the dialog type of dialog file to be executed. The dialog file associated with the OS type and dialog type of every host are execution target.	<input checked="" type="radio"/>	List	-
Include order	Same as Ansible-Legacy	<input checked="" type="radio"/>	Manual	Half-width integer
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

### 5.3.8 Nested variable maximum iteration count list (Ansible-Legacy Role only)

- (1) In the "Nested variable maximum iteration count list" menu, update the maximum iteration count of member variable array defined as nested array in the nested variable which is defined in the role package registered in "5.3.4 Role package list (Ansible-Legacy Role only)". Click the update button of the member variable that the user want to change and update the maximum iteration count.

The screenshot shows the Exastro IT Automation Ansible-LegacyRole interface. The top navigation bar includes the Exastro logo, the title 'Ansible-LegacyRole', and user information ('User name [System Administrator]', 'Login ID [administrator]', 'Change password', 'Logout'). On the left, a sidebar menu lists various options: Main menu, Movement list, Role package list, Movement details, Nested variable maximum iteration count list (which is highlighted with a red box), Substitution value auto-registration setting, Target host, Substitution value list, Execution, Check operation status, Execution list, and Contact administrator. The main content area is titled 'Display filter' and contains a table with columns: Discard, Item No., Variable name, Member variable name (iteration), and maximum iteration co. Below the table are 'Filter' and 'Clear filter' buttons, and an 'Auto-filter' checkbox. At the bottom of the content area are three buttons: 'List', 'Download all and edit file uploads', and 'Trace history'. The 'List' button is highlighted with a blue box.

Figure 5.3-15 Submenu screen (Nested variable maximum iteration count list)

- (2) Click the "List" - "Update" button to update the maximum iteration count.]  
**(※Not the registration button)**

Item No.	Variable name	Member variable name (iteration)	maximum iteration count*		last update date/time	Last updated by
1	VAR_users	-	2		Auto-input	Auto-input

Figure 5.3-16 Registration screen (Nested variable maximum iteration count list)

(3) The list of registration screen items are as follows.

**Table 5.3-10 Registration screen item list (Nested variable maximum iteration count list)**

Item	Description	Input required	Input type	Restrictions
maximum iteration count	Enter the maximum iteration count of the array in the range of 1~99,999,999.	○	Manual	Value 1~99,999,999
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

The display of member variable names are the variables of each hierarchy level scoped with ".".

Also, if the first level is nested array, the member variable name will be displayed as "-".

e.g.)

Variable definition	Display of member variable	default value of the maximum iteration count
VAR_users:	-	1
- name: alice		
authorized:		
- /tmp/alice/onekey.pub		
- create_users:	create_users	2
Name: root		
password: xxxxxxxx		
- create_users:		
Name: mysql		
password: xxxxxxxx		

The internal process initially registers the iteration count of member variable defined in the nested variable which is defined in the role package. After the initial registration, the iteration count can be updated in the "Nested variable maximum iteration count list" menu.

Also, since initial registration and update of iteration count is not in real time, it may take some time※1 until the variables can be handled in menu "[5.3.9 Substitution value auto-registration setting](#)" and menu "[5.3.11 Substitution value list](#)".

※1 The timing of extraction is described in " 7.2 About the maintenance method", so please refer to it.

### 5.3.9 Substitution value auto-registration setting

- (1) Link the parameter sheet created in menu creation function with the variables in the Movement. The registered information will reflected to "substitution value list" menu and "target host" menu by internal process.

The reflection rule is described in “6.6 BackYard contents 6.6(2) Substitution value auto-registration setting”.

The screenshot shows the Ansible-LegacyRole interface. At the top, there is a header with the Exastro logo, the title 'Ansible-LegacyRole', and user information ('User name [System Administrator]', 'Login ID [administrator]', 'Change password', 'Logout'). On the left, a sidebar menu lists various options: Main menu, Movement list, Role package list, Movement details, Nested variable maximum iteration count list, Substitution value auto-registration setting (which is highlighted with a red box), Target host, Substitution value list, Execution, Check operation status, and Execution list. The main content area displays a table titled 'Description' with columns for 'Discard', 'Item No.', 'Menu group', 'Last update date/time', and 'Last updated by'. Below the table are buttons for 'Filter', 'Clear filter', and 'Auto-filter'. To the right of the table, there are four buttons: 'List/Update', 'Register', 'Download all and edit file uploads', and 'Trace history'. At the bottom of the screen, there is a 'Contact administrator' link.

**Figure 5.3-17 Submenu screen (Substitution value auto-registration setting)**

※The screen is from Ansible-Legacy Role

- (2) Click the "Register" - "Start Registration" button to set the substitution value auto-registration.

Item No.	Parameter sheet		Registration method*	Movement	IaC variable		
	Menu group:Menu	Item			Key variable		
	Variable name	Member variable name			Substitutio		
Auto-input	▼	Select menu	▼	▼	Select Movement	Select variable name	▼

**Figure 5.3-18 Registration screen (Substitution value auto-registration setting)**

**Table 5.3-11 corresponding column list (Substitution value auto-registration setting)**

Column		Legacy	Legacy Role	Pioneer
Menugroup:Menu		○	○	○
Item		○	○	○
Registration method		○	○	○
Movement		○	○	○
Key variable	Variable name	○	○	○
	Member variable name	—	▲	—
	Substitution name	△	△	△
Value variable	Variable name	○	○	○
	Member variable name	—	▲	—
	Substitution name	△	△	△
NULL link		●	●	●

○: Required

●: Optional

△: Required only if multiple specific value can be set to the selected variable.

▲: Required only if the selected variable is nested variable

—: Not displayed

(3) The list of registration screen items is as follows.

**Table 5.3-12 Registration screen item list (Substitution value auto-registration setting)**

Column	Description	Input required	Input type	Restrictions
Menugroup:Menu	The menu of parameter list is displayed. Select the menu of association target.	○	List	-
item	The item of selected parameter list menu is displayed. Select the item of association target.	○	List	-
Registration method	Value type: Select to set the setting value of item as the specific value of the associated variable. Key type : Select to set the name of item as the specific value of the associated variable. If the setting value of the item is blank, it cannot be linked. Key-Value type : Select to set the name(Key) and setting value(Value) of item as the specific value of the associated variable.	○	List	-
Movement	The Movement registered in the Movement list will be displayed. Select the Movement.	○	List	-

Column		Description	Input required	Input type	Restrictions
Key variable	Variable name	The variables used in the file registered in Movement details menu are displayed. Select the variable to associate with its specific value in key type.	<input type="radio"/> or <input checked="" type="checkbox"/>	List	Required if the registration method is key type of key-value type.
	Member variable name	If nested variable is selected in the variable name column, the member variable of nested variable will be displayed. Select the member variable.	<input type="radio"/> or <input checked="" type="checkbox"/>	List	
	Substitution name	Required only if multiple specific value can be set to the selected variable Enter the substitution order (1~) of specific value. Value will be substituted in ascending order following the entered value. Please enter the substitution order (from 1) even if there are no more specific value.	<input type="radio"/> or <input checked="" type="checkbox"/>	Manual	Blank or positive integer.
Value variable	Variable name	The variables used in the file registered in Movement details menu are displayed. Select the variable to associate with its specific value in value type.	<input type="radio"/> or <input checked="" type="checkbox"/>	Manual	Required if the registration method is key type of key-value type.
	Member variable name	If nested variable is selected in the variable name column, the member variable of nested variable will be displayed. Select the member variable.	<input type="radio"/> or <input checked="" type="checkbox"/>	List	-
	Substitution name	Required only if multiple specific value can be set to the selected variable Enter the substitution order (1~) of specific value. Value will be substituted in ascending order following the entered value. Please enter the substitution order (from 1) even if there are no more specific value.	<input type="radio"/> or <input checked="" type="checkbox"/>	Manual	Blank or positive integer.
NULL link		Set whether to register NULL (blank) value to substitution value list menu if the specific value in parameter sheet is NULL (blank). •If the "Valid" is set, any value in the parameter sheet will be registered in the substitution value list menu. (NULL value will be registered) •If the "Invalid" is set, only specific value in the parameter sheet will be registered in the substitution value list menu (NULL value will not be registered) •If the column is blank, the "NULL link" value in Ansible interface information menu will be applied.	-	List	-
Remarks		Free description field.	-	Manual	Maximum length 4000 bytes.

※ Please refer to “5.3.11 Substitution value list” for the description of member variable name.

### 5.3.10 Target host

- (1) Register/update/discard the Movement and host associated with Operation in the "Target host" menu.

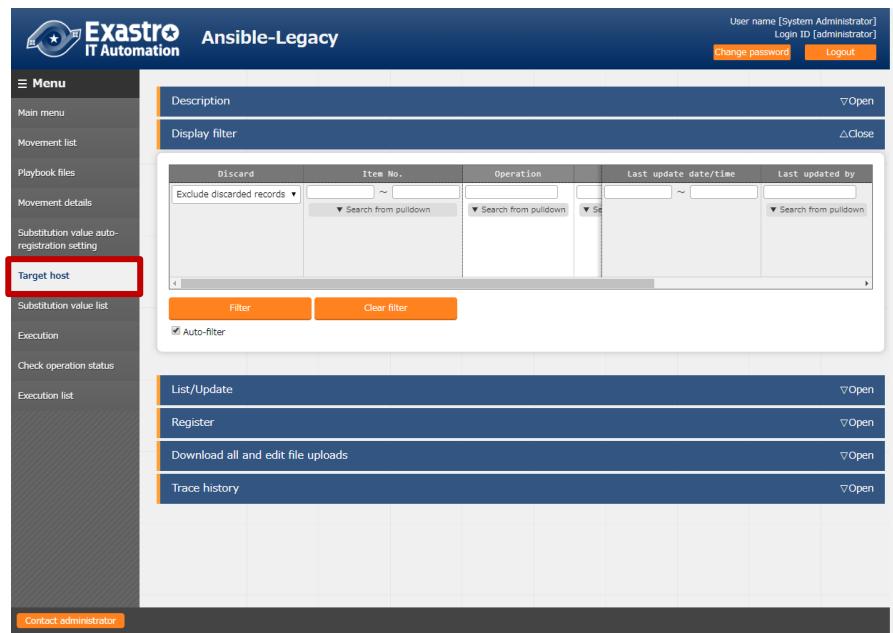


Figure 5.3-19 Submenu screen (Target host)

- (2) Click the "Register" - "Start Registration" button to register the operation target host.

Figure 5.3-20 Registraton screen (Target host)

- (3) The list of registration screen items are as follows.

Table 5.3-13 Registration screen item list (Target host)

Item	Description	Input require	Input type	Restrictions
operation	The Operation registered in the input operation list is displayed. Select the Operation.	<input type="radio"/>	List	-
Movement	The Movement registered in the Movement list is displayed. Select the Movement to associate with Operation.	<input type="radio"/>	List	-
Host	The host name registered in the device list will be displayed. Select the host to be associated with the Operation.	<input type="radio"/>	List	-
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

### 5.3.11 Substitution value list

- (1) Register/update/discard the substitution value of variable.

Users can perform maintenance (view/register/update/discard) of the specific values that are substituted with variable "VAR\_" of Playbook and template file to be used in target Movement for each operation.

Users can also maintain the specific values that are substituted with variable "LCA\_" other than "VAR\_" according to the definition of substitution table. Please refer to "[6.7 Describe substitution table\(Ansible-Legacy Role only\)](#)" for details.

The registered variable information will be output into host variable file (under host\_vars/) during operation execution.

**Figure5.3-21 Submenu screen (Substitution value list)**

※The screen is from Ansible-Legacy Role.

- (2) Click the "Register" - "Start Registration" button to manage the substitution value.

**Figure 5.3-22 Registration screen (Substitution value list)**

The variable name in substitution value list menu is reflected from the uploaded Playbook and the information registered in the substitution value auto-registration setting menu.

※ The timing of extraction is described in "[7.2 About the maintenance method](#)", so please refer to it.

(3) The list of registration screen items are as follows.

**Table 5.3-14 corresponding column list (Substitution value list)**

column	Legacy	Legacy Role	Pioneer
Operation	○	○	○
Movement	○	○	○
host	○	○	○
variable name	○	○	○
Member variable name	—	▲	—
Substitution order	△	△	△
Default value(display only)	—	○	—

○: Required

△: Required only if multiple specific value can be set to the selected variable

▲: Required only if the selected variable is nested variable

—: Not displayed

**Table 5.3-15 Registration screen item list (Substitution value list)**

Item	Description	Input required	Input type	Restrictions
Operation	The Operation registered in the operation target host is displayed. Select the Operation.	<input type="radio"/>	List	-
Movement	The Movement associated with the Operation selected from the data registered in the target host menu is displayed. Select the Movement.	<input type="radio"/>	List	-
Host	The host associated with the Operation and Movement selected from the data registered in the target host menu is displayed. Select the host.	<input type="radio"/>	List	-
Variable name	The variable name attached with the Movement selected from the data registered in the Movement details menu is displayed. Select the variable.	<input type="radio"/>	List	-
Member variable name	If nested variable is selected in the variable name column, the member variable of nested variable will be displayed. Select the member variable.	<input type="radio"/> or <input type="checkbox"/>	List	-
Specific value	Enter the specific value used in Operation / Movement / Host. File embedded variable "CPF_" and template embedded variable "TPF_" can be entered in the specific value column. When describing the variable, enclose the variable name in {} as describing them in the Playbook. e.g.) Entering TPF_sample as specific value. '{{△TPF_sample△}}'     △: Half-width space ': recommended	<input type="radio"/>	Manual	Maximum length 1024 bytes
Substitution order	Required only if multiple specific value can be set to the selected variable. Enter the substitution order (1~) of specific value. Value will be substituted in ascending order following the entered value. Please enter the substitution order (1~) even if there are no multiple specific value.	<input type="radio"/> or <input type="checkbox"/>	Manual	Blank or positive integer
Default value	The specific value of variable selected in the variable name or member variable name column set in the default variable definition file(defaults->main.yml) is displayed Please refer to " 6.4 Describe ITA readme(Ansible-Legacy Role only)" for details . True is displayed when the specific value is "Yes","Y", or "y". False is displayed when the specific value is "No","N", or "n".	-	Display only	-
Remarks	Free description field.	-	Manual	Maximum length 4000 bytes

## 【The display content of member variable name】

Selecting member variable is required only if the variable is nested variable.

Only the variable that requires specific value is displayed in the member variables.

The display of variable names of each hierarchy level is scoped with "." .

If the variable is in nested array, the variables are scoped with "[]" at the iteration position (0~).

The iteration array count is set in "5.3.8 Nested variable maximum iteration count list (Ansible-Legacy Role only)".

e.g.)

Variable definition	Display of member variable
VAR_users:	
- name: alice	[0].name
authorized:	[0].authorized
- /tmp/alice/onekey.pub	
mysql:	
password: mysql-password	[0].mysql.password
hosts:	[0].mysql.hosts
- "127.0.0.1"	
- "localhost"	
- name: bob	[1].name
authorized:	[1].authorized
- /tmp/alice/onekey.pub	
mysql:	
password: mysql-password	[1].mysql.password
hosts:	[1].mysql.hosts
- "127.0.0.1"	※mysql is the variable which indicates the hierarchy
- "localhost"	directory, so is not displayed in member variable.

The information registered in "substitution value auto registration setting" menu is reflected to "substitution value list" menu and "target host" menu by internal process.

※ The timing of extraction is described in " 7.2 About the maintenance method", so please refer to it.

## ① Entering the substitution order

In Ansible-Legacy, if the substitution order is not entered, the variable will be treated as normal variable.

If the substitution order is entered, the variable will be handled as multiple specific value variable.

Please enter the substitution order although multiple specific value is not required (one specific value is sufficient) if the variable is multiple specific value variable.

In Ansible-Legacy Role, by selecting variable name or member variable name, it is possible to enter substitution order only for multiple specific value variables.

Please enter if the variable is multiple specific value variables.

In Ansible-Pioneer, if the substitution order is not entered, the variable will be handled as normal variable.

If the substitution order is entered, the variable will be handled as multiple specific value variable

Please enter the substitution order although multiple specific value is not required (one specific value is sufficient) if the variable is multiple specific value variable.

In each mode, it is no problem although the substitution order is not consecutive for specific multiple concrete value variables.

e.g.)

**Registration in substitution  
value list menu**

Host	Variable	Specific Value	Substitution order
HOST_A	VAR_std	value1	
HOST_A	VAR_list_a	value2	10
HOST_A	VAR_list_b	value3	100
HOST_A	VAR_list_b	value4	200

**The content output to the host variable file of HOST\_A**

```
VAR_std: value1
VAR_list_a:
- value2
VAR_list_b:
- value3
- value4
```

## ② Output to the host variable file

The specific value of variable registered in substitution value list menu will be output to host variable file.

In Ansible-Legacy and Ansible-Pioneer, if the specific value of variable used in Playbook or dialog file is not registered in substitution value list menu during operation execution, unexpected error will occur.

In Ansible-Legacy Role, only the variable registered in substitution value list menu the will be output to host variable file during operation execution.

It is same for nested variables that only the member variable registered specific value will be output.

e.g.)

**Variable definition**

```
VAR_users:
  - name: alice
    authorized:
      - /tmp/alice/onekey.pub
  mysql:
    password: mysql-password
    hosts:
      - "127.0.0.1"
      - "localhost"
  - name: bob
  omitted
```

**Registration in substitution value list menu**

Host	Variable	Member variable	Specific value	Substitution order
HOST_A	VAR_users:	[0].name	value1	
HOST_A	VAR_users	[1].authorized	value2	

**The content output to the host variable file of HOST\_A**

```
VAR_users:
  - name :value1
  - .authorized: value2
```

### ③ Default value check option

In the "System settings" of "ITA Management console", users can set the parameter to display warning message and not register the specific value when registering the specific value of the variable whose default value does not match between multiple roles.

This parameter is not registered by default. Please register if necessary.

The content to register in system settings is as follows.

Also, please refer to "User instruction manual\_Ansible-Management console" for system settings.

**Table 5.3-16 Registration content in system settings**

Item	Input value	Input required
ID	ANSIBLE_DEF_VAL_CHK	<input type="radio"/>
Item name	Any desired string	-
Setting value	1: Parameter enabled Contents other than 1 or record not registered: Parameter disabled.	<input type="radio"/>
Remarks	Any desired string	

### 5.3.12 Check operation status

- (1) Monitor the status of operation execution.

The screenshot shows the Exastro IT Automation Ansible-Legacy web interface. The left sidebar has a red box around the 'Check operation status' item under the 'Execution' section. The main content area has a blue header 'Ansible-Legacy'. Below it, there are two tabs: 'Description' (selected) and 'Target Operation'. The 'Description' tab contains a table with the following data:

Item	Value
Execution No.	3
Execution type	Normal
Status	Completed
execution engine	Ansible
Caller symphony	test
Execution user	System Administrator

Below this table, there are two sections: 'Movement' and 'Operation'. The 'Movement' section has a table with the following data:

ID	Name
1	testing-one

The 'Operation' section has a table with the following data:

No.	Name	ID
1	execution	1

At the bottom, there is a 'Progress status(Execution log)' section and an orange 'Contact administrator' button.

Figure 5.3-23 Submenu screen (Check operation status)

## ① Display of execution status

"Status" is displayed according to the execution status.

Also, the details of the execution status is displayed in execution log and error log

In the "execution type", "Dry run" is displayed when performing dry run, "Normal" will be displayed for other cases

If the status ends with an unexpected error, the cause is incomplete registration of web contents, message will be displayed in error log.

In addition, in the case that communication with Ansible RestAPI fails due to incomplete registration in "5.2.1 Interface information", message will not be displayed in error log.

In this case, error information will be record in application log. Please check the application log if necessary.

The symphony which the operation is executed from is displayed in "Caller symphony"

The column will be blank if the operation is executed directly from Ansible-Legacy, Pioneer, LegacyRole driver.

The login user when clicking the "execute" or "dry run" button in the "exeuction" menu will be displayed in "Execution User".

## ② Host management

By clicking the "confirmation" button, "5.3.10 Target host" will display and the host filtered by the operation and Movement of operation target will be displayed.

## ③ Substitution value confirmation

By clicking the "confirmation" button, "5.3.11 Substitution value list" will display and the substitution value filtered by the operation and Movement of operation target will be displayed.

## ④ Emergency stop/ Schedule cancellation

It is possible to stop the construction operation by clicking the "Emergency stop" button

In addition, for the "scheduled execution" operation before execution, the "schedule cancellation" button will display. Cancel the scheduled execution by clicking the "schedule cancellation" button.

## ⑤ Display of execution log

When executed with Ansible Tower, the playbook is executed for each device to be built grouping by the item values such as user, password, and instance groove in the device list. The device to be built, and the execution log of ansible is divided.

Further, by specifying the number of job slices in the optional parameter of the Movement list, the grouped devices to be built are further divided by the number of job slices, the playbook is executed, and the ansible execution log is also divided.

When the execution log is divided, the pull-down of the display log file is displayed, and can select the log file and want to display.



These two are the following types of log file names displayed in the pull-down list of display log files.

exec.log: A log file that summarizes all execution logs.

Files other than exec.log: Are split execution log files. The file naming conventions is as follows.

Ita\_<mode name>\_executions\_jobtpl\_<work number>\_<group number>\_<serial number>

Table 5.3-17 Naming elements for split execution log files.

Element	Content
Modename	Executed mode name legacy/pioneer/legacy_role
Execution number	Executionnumber of execution list menu.
Group number	Serial number from 1 that is grooved by the item value of the user, password, instance groove etc. of the device list and the device to be built.
Serial number	Serial number from 1 that divides the group by setting the number of job slices. If 0, no division of job slicing was done.

## ⑥ Log filter

Execution log and error log can be filtered. By entering the string that the user wants to search in the filter box of each log and checking the "Display only corresponding lines" checkbox, only the corresponding line will be displayed.

The display refresh cycle and the maximum display line count of execution and error log can be set in "Status monitoring cycle (milliseconds)" and "Number of rows to display progress status" of "5.2.1 Interface information" menu.

## ⑦ Input data

Users can download files such as the executed Playbook.

Please refer to "8.1 The linkage between the input data used during Ansible execution and ITA menu" for the configuration of input data.

## ⑧ Result data

Users can download files such as execution log and error log.

### 5.3.13 Execution list

- (1) The history of operation can be viewed here.

The operation list table and graph will display by specifying criteria and clicking the "filter" button.

By hovering the mouse cursor over each graph, the download button of each graph will be displayed on the top-right of the graph.

Users can download the graph by clicking the download button of each displayed graph.

By clicking the "Check execution status" button, the screen will transit to "5.3.12 Check operation status" and the details of execution status can be viewed.

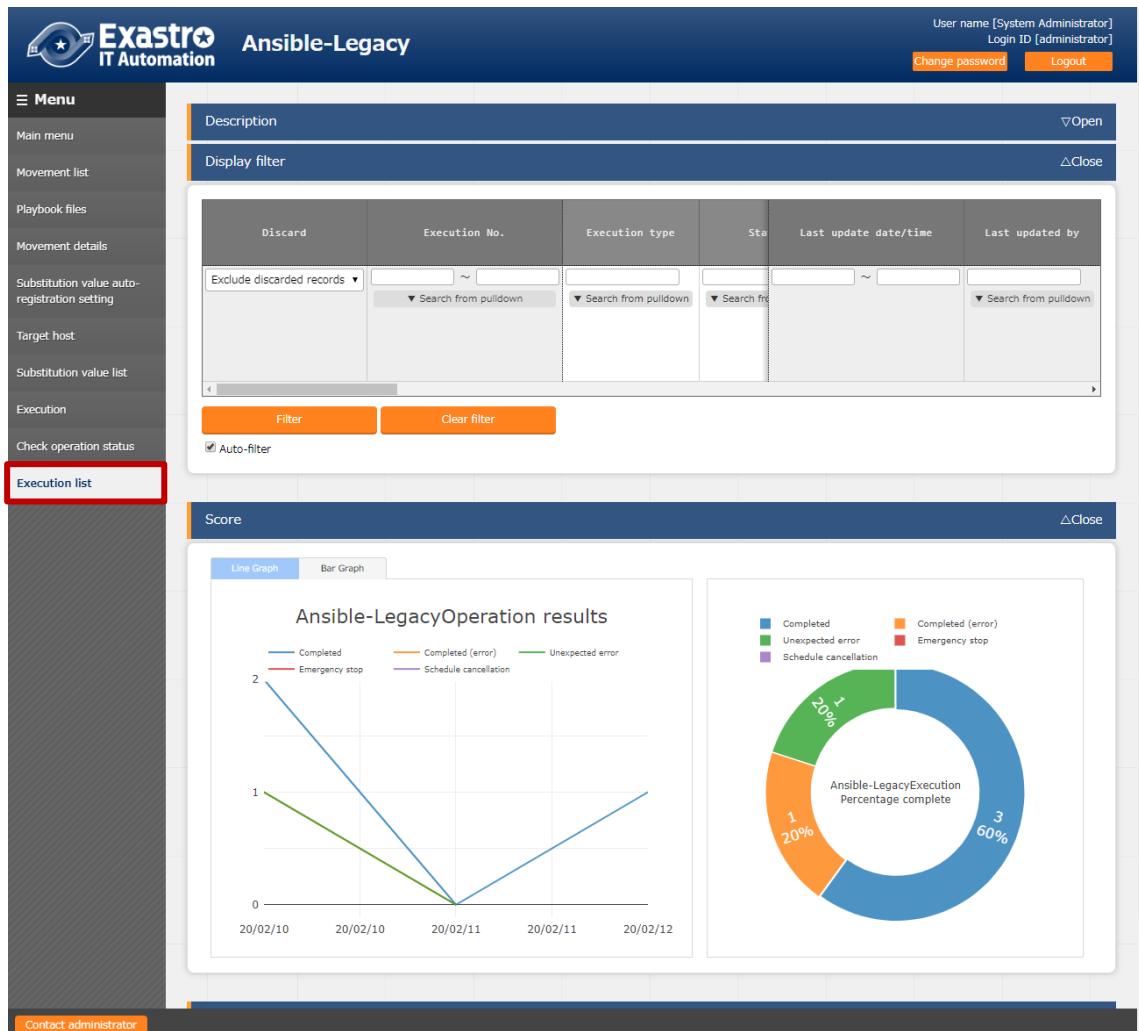


Figure 5.3-24 Submenu screen (Execution list)

### 5.3.14 Execution

- (1) Indicate Operation execution. Select the radio button from the Movement list and operation list and click the execution button, the screen will transit to "5.3.12 Check operation status" and the operation will be executed.

The screenshot shows the Ansible-Legacy software interface. The top navigation bar includes 'User name [System Administrator]', 'Login ID [administrator]', 'Change password', and 'Logout'. The left sidebar, titled 'Menu', lists various options: Main menu, Movement list, Playbook files, Movement details, Substitution value auto-registration setting, Target host, Substitution value list, **Execution** (which is highlighted with a red box), Check operation status, and Execution list. The main content area contains three sections: 'Movement [Filter]' and 'Movement [List]' (with a table showing one entry: Movement ID 1, Name testing-one, Orchestrator Ansible Legacy, Host name, Last update date/time 2020/02/10 10:52:16, Last updated by System Administrator); 'Operation [Filter]' and 'Operation [List]' (with a table showing two entries: Operation ID 1, Name execution, Scheduled date for execution 2020/02/10 10:43, Last ex 2020/02/12 08:33:05, Last update date/time 2020/02/10 14:31:41, Last updated by Legacy execution procedure; Operation ID 2, Name legacy, Scheduled date for execution 2020/02/10 14:31:41, Last ex 2020/02/10 14:31:41, Last update date/time 2020/02/10 14:31:41, Last updated by System Administrator); and a summary table at the bottom with Movement ID 1, Movement Name testing-one, Operation ID 1, and Operation Name execution. At the bottom are 'Dry run' and 'Execute' buttons.

**Figure 5.3-25 Submenu screen (Execution)**

※The screen is from Ansible Legacy

#### ① Dry run

By clicking the "Dry run" button, dry run can be executed without actually constructing the target device. In the case of dry run, the operation of each mode is as follows.

Driver	Action
Ansible-Legacy	Execute the playbook by specifying the -check parameter to the Ansible-Playbook command.
Ansible-Legacy Role	Execute the role by specifying the - check parameter to the Ansible-Playbook command.
Ansible-Pioneer	Only perform the connection check to target device

#### ② Specify scheduled date/time

Execution can be scheduled by entering "Scheduled date/time" column.  
Only future date/time can be registered for "Scheduled date/time"

## 6 How to write construction code

### 6.1 Describe Playbook (Ansible-Legacy)

Playbooks uploaded to 5.3.3 Playbook file list (Ansible-Legacy only) are included in the Playbook file generated by ITA and executed.

#### (1) Header section

The playbook does not require a header section when being uploaded.

The default value of the header section is fixed, but you can change it in the header section of "5.3.2 Movement list".

Defult value of header section	
• For Ansible engine	• For AnsibleTower
- hosts: all	- hosts: all
remote_user: "{{ __loginuser__ }} "	gather_facts: no
gather_facts: no	become: yes
become: yes	

#### (2) tasks section

Please refer to the official manual for the basic Playbook format.

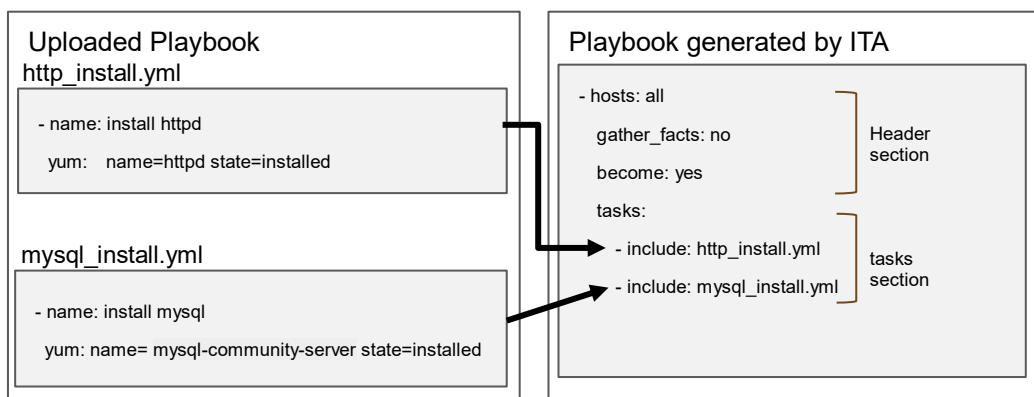
Please use UTF-8 for the character encoding.

Please adjust the indent in the Playbook to multiple of 2.

e.g.)

```
-△name: Service script setting
△△template:
△△△△src: "{{ item.src }}"
△△△△dest: "{{ item.dest }}"
△△△△owner: "{{ item.owner is none |ternary('root', item.owner) }}"
△△△△group: "{{ item.group is none |ternary('bacula', item.group) }}"
△△△△mode: "{{ item.mode is none |ternary('0654', item.mode) }}"
    ~omitted~
△: Half-width space
```

Uploaded Playbooks are included in the include order of "[5.3.7 Movement details](#)".



## 6.2 Describe Dialog file (Ansible-Pioneer)

The dialog file in Ansible-Pioneer incorporates an ITA-specific module into Ansible.

The dialog file is in ITA-specific format.

Please use UTF-8 for the character encoding.

### (1) Structure of dialog file

The dialog file is constructed with 2 types of section.

Section name	Usage
Conf	Specify the timeout value according to timeout parameter. timeout value: 1~3600(unit:second)
exec_list	Construct the target host with 4 kinds of dialog command.

Describe the timeout parameter in the beginning of dialog file then describe the dialog comment in the later part.

Comments can be described in the same format with Ansible basic format.

```
e.g.)  
# Comment  
conf:  
△△timeout: 10  
exec_list:  
※△: Half-width space  
Please enter 2 half-width space before "timeout:"
```

### (2) Dialog command

There are 4 kinds of dialog commands as follows.

Module	Use
exec	Input command to the target host
expect	Waiting for the output of expected string (prompt) from the contents that the target host outputs to the standard output.
state	Input the command to target host. The contents of the standard output until the prompt is output to the standard output are analyzed by external shell, and the result is determined.
command	Loops and conditional branching can be performed before and after inputting commands to the target host.

#### ① expect module

Waiting for the output of expected string (prompt) from the contents that the target host outputs to the standard output.

Describe the expected string **in regular expression**.

When the expected string is received, proceed to the next. In addition, if the string is not received within the time specified by the time out parameter, the dialog file will terminate abnormally.

```
e.g.) Waiting for the prompt of password entry via telnet connection  
△△-△expect:△'Password'  
※△: half-width space  
Please enter 2 half-width space before "- expect:"  
It is recommended to enclose the waiting string with quotation.
```

## ② exec module

Input command to the target host.

exec module and expect module are used in pairs.

e.g.) Wait for the password entry prompt via telnet connection and input password.

△△-△expect:△'Password'

△△△△exec:△itapassword

**※△: Half-width space**

**Please enter 4 half-width space before the description of "exec:"**

**It is recommended to enclose the waiting string with quotation if necessary.**

## ③ state module

Input the command to target host. The contents of the standard output until the prompt is output to the standard output are analyzed by external shell, and the result is determined.

### The format of state module

Parameter	Required/ Optional	Description
△△-△state:△xxx	Required	Specify the input command.
△△△△prompt:△xxx	Required	Specify the waiting prompt. The prompt can be described in regular expression.
△△△△shell:△xxx	Optional	Specify the shell file name to check the result with the created shell.  If the exit code of the created shell is 0, the result is determined as normal, and the others are abnormal.  This parameter is not required when checking the result with default shell.  The default shell will grep the contents of standard output with the string specified by parameter (-).  If there is at least one matching row, the result is determined normal, and if there is no matching row, the result is determined to be abnormal.  In addition, if the parameter is not specified, the result will be determined as abnormal. In the case of using the parameter to save the command result (standard output) to the file specified by stdout_file, please specifiy yes for ignore_errors.
△△△△parameter: △△△△△△-△xxx △△△△△△-△xxx	Optional	Specify the string to search for the result (standard output) of the input command. If the shell is specified, the strings will become the parameters during shell execution. Enumerate the criteria strings if there are multiple of them.
△△△△stdout_file:△xxx	Optional	The file to save the result (standard output) of the input command.
△△△△success_exit:△xxx	Optional	Please set this file to the shell parameter if the shell parameter if specified.  Specify "yes" to exit the dialog normally if the search result is normal and specify no to proceed to the next. "no" is set on default.

Parameter	Required/ Optional	Description
△△△△ ignore_errors:△ xxx ※△:Half-width space	Optional	Specify "yes" to proceed to the next even if the search result is abnormal. "no" is set on default.

#### Exp1)

cat the hosts file and grep the displayed results with parameter value. If there is line containing 139.0.0.1 or lalhost, the result is determined as normal and proceed to the next. If there is no such line, the result is determined as abnormal and the dialog file is terminated abnormally.

exec\_list:

```
- state: cat /etc/hosts
prompt: root@{{ __loginhostname__ }}
parameter:
- 139.0.0.1
- lalhost
```

#### Exp2)

cat the hosts file and grep the displayed results with parameter value. If there is line containing 139.0.0.1 or lalhost, the result is determined as normal and terminate normally according to the success\_exit:yes setting. If there is no such line, the result is determined as abnormal and the dialog file is terminated abnormally

exec\_list:

```
- state: cat /etc/hosts
prompt: root@{{ __loginhostname__ }}
parameter:
- 139.0.0.1
- lalhost
success_exit: yes
```

#### Exp3)

cat the hosts file and grep the displayed results with parameter value. If there is line containing 139.0.0.1 or lalhost, the result is determined as normal and proceed to the next. If there is no such line, the result is determined as abnormal and proceed to the next according to the ignore\_errors:yes setting.

exec\_list:

```
- state: cat /etc/hosts
prompt: root@{{ __loginhostname__ }}
parameter:
- 139.0.0.1
- lalhost
ignore_errors: yes
```

#### Exp4)

cat the hosts file and use the user created shell to grep the displayed results with parameter value. If there is line containing 139.0.0.1 or lalhost, the result is determined as normal and proceed to the next. If there is no such line, the result is determined as abnormal and the dialog file is terminated abnormally

exec\_list:

```
- state: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  shell: /tmp/grep.sh
  stdout_file: /tmp/stdout.txt
  parameter:
    - 139.0.0.1
    - lalhost
```

User created shell(/tmp/grep.sh)

```
#!/bin/bash
STDOUT=/tmp/STDOUT.tmp
STDERR=/tmp/STDERR.tmp
cat /tmp/stdout.txt|grep $1|grep $2 | wc -l >${STDOUT} 2>${STDERR}
RET=$?
if [ $RET -ne 0 ]; then
    EXIT_CODE=$RET
else
    if [ -s ${STDERR} ]; then
        EXIT_CODE=1
    else
        CNT=`cat ${STDOUT}`
        if [ ${CNT} -eq 0 ]; then
            EXIT_CODE=1
        else
            EXIT_CODE=0
        fi
    fi
fi
/bin/rm -rf ${STDOUT} ${STDERR} >/dev/null 2&>1
exit ${EXIT_CODE}
```

#### Exp5)

cat the hosts file and save the displayed result to the file specified by stdout\_file then proceed to the next. If the “no” parameter is set to the default shell, the result will be determined as abnormal.

Set ignore\_errors:yes to proceed to the next. exec\_list:

```
- state: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  stdout_file: {{ __symphony_workflowdir__ }}/hosts
  ignore_errors: yes
```

④ command module

Loops and conditional branching can be performed before and after inputting commands to the target host.

The format of command module

Parameter	Required/ Optional	Description
$\Delta\Delta - \Delta\text{command}:\Delta\text{xxx}$	Required	Specify the input command.
$\Delta\Delta\Delta\Delta\text{prompt}:\Delta\text{xxx}$	Required	Specify the waiting prompt. It can be described in regular expression.
$\Delta\Delta\Delta\Delta\text{timeout}:\Delta\text{xxx}$	Optional	Specify the timer to wait for the prompt after the command is sent. If the parameter is omitted, then conf->timeout is used.
$\Delta\Delta\Delta\Delta\text{register}:\Delta\text{xxx}$	Optional	Save the information of the standard output to any string specified after sending the command. When using with_items to loop, the information of the standard output after the last command input is saved. This variable can be used in condition judgment (can only be used in condition judgment). However, saving the information of standard output for every variable name is not possible. Previous information will be overwritten.
$\Delta\Delta\Delta\Delta\text{with_items}:$ $\Delta\Delta\Delta\Delta\Delta\Delta-\Delta\text{xxx}$ $\Delta\Delta\Delta\Delta\Delta\Delta-\Delta\text{xxx}$	Optional	The variable set in with_items are multiple specific value variable. The number of specific value of the variables set in with_items does not have to be the same. If the numbers are not the same, the process will loop with the maximum number of the specific value of each variable. The specific value of the variables whose specific value is not enough is handled as null value. However, the above does not apply to prompt or timeout. If the specific value is insufficient in prompt or timeout, error will occur. The scope of each variable is item.X(X is from 0 to 99). The application range of variable (item.X) is other than register/when. Please set the variable as the follows when using with_items in prompt and timeout. prompt: {{ $\Delta$ VAR_prompt_XXX $\Delta$ }} timeout: {{ $\Delta$ VAR_timeout_XXX $\Delta$ }} ( $\Delta$ is half-width space. XXX is made up of any desired half-width alphanumeric character or underscore)
$\Delta\Delta\Delta\Delta\text{when}:$ $\Delta\Delta\Delta\Delta\Delta\Delta-\Delta\text{xxx}$ $\Delta\Delta\Delta\Delta\Delta\Delta-\Delta\text{xxx}$	Optional	The condition judgement before command executes. Execute command if the condition matches. Move to the next “command” line if the condition doesn't match. Conditional expression Judging variable definition VAR_xx is define                  Variable defined        true VAR_xx is undefined              Variable undefined    true ※define/undefine can be specified only for ITA variable(VAR_xx) Judging variable specific value VAR_xx/register variable relational_operator string VAR_xx/register variable relational_operator VAR_xx VAR_xx/register variable match(Regular expression string /VAR_xx) VAR_xx/register variable no match(Regular expression string /VAR_xx)

Parameter	Required/ Optional	Description						
		<p>※Relational operators are 「==」、「!=」、「&gt;」、「&gt;=」、「&lt;」、「&lt;=」</p> <p>※The 「&gt;」、「&gt;=」、「&lt;」、「&lt;=」 relational operators are assumed to be used for numerical values.</p> <p>※It is not required to enclose string and regular expression string with single or double quotations.</p> <p style="padding-left: 2em;">VAR_status match(active)    VAR_status == active</p> <p>Compound condition with and / or</p> <p>When processing with or condition, add OR between judge conditions</p> <ul style="list-style-type: none"> <li>- VAR_status == 1 OR VAR_status == 2</li> </ul> <p>When processing with and condition, describe the statement in multiple lines.</p> <ul style="list-style-type: none"> <li>- VAR_status == 1 OR VAR_status == 2</li> <li>- VAR_sub_status == 1</li> </ul>						
△△△△exec_when: △△△△△△-△xxx △△△△△△-△xxx	Optional	<p>Judge condition for every loop (continue condition)</p> <p>Perform condition judgement if with_items is described.</p> <p>If the condition matches, execute command of the corresponding loop.</p> <p>If the condition doesn't match, move on to the next loop.</p> <p>Conditional expression</p> <p>Judging variable definition</p> <table border="0"> <tr> <td>VAR_xx is define</td> <td>Variable defined</td> <td>true</td> </tr> <tr> <td>VAR_xx is undefined</td> <td>Variable undefined</td> <td>true</td> </tr> </table> <p>※define/undefine can be specified only for ITA variable(VAR_xx)</p> <p>Judging variable specific value</p> <p>VAR_xx/register variable relational_operator string</p> <p>VAR_xx/register variable relational_operator VAR_xx</p> <p>VAR_xx/register variable match(Regular expression string /VAR_xx)</p> <p>VAR_xx/register variable no match(Regular expression string /VAR_xx)</p> <p>※Relational operators are 「==」、「!=」、「&gt;」、「&gt;=」、「&lt;」、「&lt;=」</p> <p>※The 「&gt;」、「&gt;=」、「&lt;」、「&lt;=」 relational operators are assumed to be used for numerical values.</p> <p>※It is not required to enclose string and regular expression string with single or double quotations.</p> <p style="padding-left: 2em;">VAR_status match(active)    VAR_status == active</p> <p>Compound condition with and / or</p> <p>Perform the same description format as "when:"</p>	VAR_xx is define	Variable defined	true	VAR_xx is undefined	Variable undefined	true
VAR_xx is define	Variable defined	true						
VAR_xx is undefined	Variable undefined	true						
△△△△failed_when: △△△△△△-△xxx △△△△△△-△xxx	Optional	<p>Condition judgment for the stdout content after command execution(for every loop)</p> <p>Perform condition judgment even if with_items is not described</p> <p>If the condition matches, the result is normal</p> <p>If the condition doesn't match, the result is abnormal and the dialog file is terminated abnormally</p> <p>Conditional expression</p> <p>Judging variable specific value</p> <p>stdout relational_operator string</p> <p>stdout relational_operator VAR_xx</p>						

Parameter	Required/ Optional	Description
※△: half-width space		<p>stdout match(regular expression string/VAR_xx)      stdout no match(regular expression string/VAR_xx)</p> <p>※Relational operators are 「==」、「!=」、「&gt;」、「&gt;=」、「&lt;」、「&lt;=」</p> <p>※The 「&gt;」、「&gt;=」、「&lt;」、「&lt;=」 relational operators are assumed to be used for numerical values.</p> <p>※It is not required to enclose string and regular expression string with single or double quotations.</p> <p>VAR_status match(active)    VAR_status == active      Compound condition with and / or      Perform the same description format as “when:”</p>

```

Exp1)
conf:
  timeout: 30

exec_list:
# If waiting for strings other than prompt is required, use the combination of expect/exec.
# In the case that the password is required.
- expect: 'password:'
  exec:   '{{ __loginpassword__ }}'

# If the ITA variable, VAR_hosts_make, is described in host variable file, cat the host file.
# If the variable is not described, skip the command.
- command: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  when:
    - VAR_hosts_make is define

```

```

Exp2)
conf:
    timeout: 30

exec_list:
# If waiting for strings other than prompt is required, use the combination of expect/exec.
# In the case that the password is required.
- expect: 'password:'
  exec:   '{{ __loginpassword__ }}'

# If the ITA variable, VAR_hosts_make, is described in host variable file, cat the host file.
# If the variable is not described, skip the command."
# Use cat to save the contents of the standard output hosts file to result_stdout.
- command: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  register: result_stdout
  when:
    - VAR_hosts_make is define

# If the ITA variable, VAR_hosts_make, is described in host variable file, cat the host file.
# If the variable is not described, skip the command.
# Execute the command for the numbers of the specific values of the multiple specific value
# variable set in the with_items.
# From the result of condition judgment for each loop, if "ip address host name" does not
# correspond to the hosts file, execute command.
# Add "IP_address host_name" to the last line of hosts file by using echo.
- command: 'echo {{ item.0 }} {{ item.1 }} >> /etc/hosts'
  prompt: 'root@{{ __loginhostname__ }}'
  when:
    - VAR_hosts_make is define
  with_items:
    - '{{ VAR_hosts_ip }}'      # item.0
    - '{{ VAR_hosts_name }}'   # item.1
  exec_when:
    - result_stdout no match('{{ item.0 }} *{{ item.1 }}')

```

```

Exp3)
conf:
    timeout: 30

exec_list:
# If waiting for strings other than prompt is required, use the combination of expect/exec.
# In the case that the password is required.
    - expect: 'password:'
      exec:   '{{ __loginpassword__ }}'

# Execute the command for the numbers of the specific values of the multiple specific value
# variable set in the with_items
# Execute auto startup configuration.
    - command: 'systemctl enable {{ item.0 }}'
      prompt: 'root@{{ __loginhostname__ }}'
      with_items:
        - '{{ VAR_service_name_list }}' # item.0

# Execute the command for the numbers of the specific values of the multiple specific value
# variable set in the with_items
# Execute service startup
    - command: 'systemctl start {{ item.0 }}'
      prompt: 'root@{{ __loginhostname__ }}'
      with_items:
        - '{{ VAR_service_name_list }}' # item.0

Execute the command for the numbers of the specific values of the multiple specific value
variable set in the with_items.
Output the service status to standard output.
If the content of result output to standard output contains the regular expression of item.1, the
result is right.
For example, in the case that the specific value of VAR_service_status_list is set to running and
the service is running, "running" in "Active: active(running)" matches so the result is right. (Move
on to the next loop)
In the case that condition doesn't match, the result is determined as abnormal and the dialog
file terminates abnormally.
    - command: 'systemctl status {{ item.0 }}'
      prompt: 'root@{{ __loginhostname__ }}'
      with_items:
        - '{{ VAR_service_name_list }}' # item.0
        - '{{ VAR_service_status_list }}' # item.1
      failed_when:
        - stdout match('{{ item.1 }}')

```

```

Exp4)
conf:
    timeout: 30

exec_list:
# If waiting for strings other than prompt is required, use the combination of expect/exec.
# In the case that the password is required.
    - expect: 'password:'
      exec:   '{{ __loginpassword__ }}'

# Execute the command for the numbers of the specific values of the multiple specific value
# variable set in the with_items.
# When describing the command with "{{item.0}}" only, enclose it with double-quotation.
# Please note the numbers of specific value when using with_items in prompt or timeout.
# prompt→command→prompt→command→prompt…(loops thereafter), it is required to plus 1
# to the command count. (Same for timeout)
    - command: "{{ item.0 }}"
      prompt: '{{ item.1 }}'
      timeout: '{{ item.2 }}'
      with_items:
        - '{{ VAR_command_list }}' # item.0
        - '{{ VAR_prompt_list }}' # item.1
        - '{{ VAR_timeout_list }}' # item.2

```

Variable name	Specific value
VAR_command_list	systemctl status {{ item.1 }}
VAR_service_name_list	ky_pioneer_execute-workflow.service

```

Exp5)
conf:
    timeout: 30

exec_list:
# If waiting for strings other than prompt is required, use the combination of expect/exec.
# In the case that the password is required.
    - expect: 'password:'
      exec:   '{{ __loginpassword__ }}'

# Execute the command for the numbers of the specific values of the multiple specific value
# variable set in the with_items.
# Setting the specific value of substitution value list to {{item.X}} is possible. In that case, please
# make the numerical value described in the specific value of item.X which is described in the
# dialog file grows.
# The command executed in this example is "systemctl status ky_pioneer_execute-
# workflow.service".
    - command: "{{ item.0 }}"
      prompt: 'root@{{ __loginhostname__ }}'
      with_items:
        - '{{ VAR_command_list }}' # item.0
        - '{{ VAR_service_name_list }}' # item.1

```

```

Exp6)
conf:
    timeout: 30

exec_list:
# If waiting for strings other than prompt is required, use the combination of expect/exec.
# In the case that the password is required.
    - expect: 'password:'
        exec:   '{{ __loginpassword__ }}'

# Example of compound condition using and / or.
# When processing with or condition, describe the if statement horizontally.
# When processing with and condition, describing the statement in multiple lines.
# "when" is used as the example here but the same applies to exec_when and failed_when.
    - command: echo aaa
        prompt: 'root@{{ __loginhostname__ }}'
    when:
        - 10 == 9 OR 10 != 9 OR 10 >= 9
        - 10 > 9 OR 10 <= 9 OR 10 < 9

```

and condition

## ⑤ localaction module

Execute command on Ansible/Ansible Tower server.

The format of localaction module

Parameter	Required/ Optional	Description
△△-△localaction:△xxx	Required	Specify the command to be executed.
△△△△ ignore_errors:△xxx ※△:Half-width space	Optional	Specify “yes” to continue if the execution result of the command is abnormal. If “no” is specified, the dialog will end if the result of execution is abnormal. Default is “no”.

Exp1)

During Symphony execution, create a directory to output the hosts file for every host in the shared directory ({{ \_\_symphony\_workflowdir\_\_ }}) of each Movement.

exec\_list:

```

    - localaction: mkdir -p 755 {{ __symphony_workflowdir__ }}/{{ __loginhostname__ }}
        ignore_errors: yes
    - state: cat /etc/hosts
        prompt: '{{ __loginuser__ }}@{{ __loginhostname__ }}'
        stdout_file: {{ __symphony_workflowdir__ }}/{{ __loginhostname__ }}/hosts
        ignore_errors: yes

```

### (3) Regular expression

The strings described in the following command and parameter are evaluated in regular expression.

- expect module
- The prompt parameter of state module
- The prompt parameter of command module

When the string is described in regular expression contains metacharacter "()\{}.", etc., inserting escape character "\\$" before metacharacters is required.

Exp1)

When waiting for the following command, the red characters are metacharacters.

XAMPP Developer Files [Y/n] exec\_list:

Inserting escape character "\\$" before metacharacters is required.

XAMPP Developer Files \\$[Y\\$n\\$] exec\_list:

State module and command module extracts the result (standard output) of the executed command. The notes of the extraction are as follows.

- ① The delimitation between the result (standard output) of the executed command and the prompt.

The delimitation between the result (standard output) of the executed command and the prompt is performed by the string specified in the prompt parameter. When judging the result of the executed command (standard output) or saving it to a file, please do not describe a preceding match with .\* in regular expression. The result (standard output) of the executed command can not be extracted.

.\* Example of preceding match with .\* in regular expression.

'.\*[\\$\#\\$\\$\%] \\$'

- ② Support of escape sequence

Depending on the target device, an Operating System Command sequence may be added immediately before the prompt sent from the target device.

Escape sequences immediately before the string specified by the prompt parameter are excluded.

### (4) Notes when using multiple specific value variable

The only parameter in the dialog file that can use multiple specific value variable is the with\_items parameter of command module. If multiple specific value variable is used in other cases, the operation execution will turn out to be error.

## 6.3 Describe role package (Ansible-Legacy Role)

Please refer to the Ansible best practices official manual for the basic format.

Please use UTF-8 for the character encoding.

This section describes the directory that is required to be in the zip file of role package file uploaded in "5.3.4 Role package list (Ansible-Legacy Role only)" and the process in ITA.

(Parent directory)

|

|—site.yml

site.yml(master Playbook) is created in ITA.

|

The file will be overwritten if exists.

|

|—hosts

hosts file is created in ITA.

|

The file will be overwritten if exists.

|

|—group\_vars

host group variables are not handled.

|

The group\_vars directory will be removed if exists.

|

|—host\_vars

Host variables are created in ITA.

|

This directory will be overwritten if it exists.

|

|—ITA readme

ITA readme is defined for every role. Error doesn't occur even if the file doesn't exist

|

The naming rule of ITA readme file name:

|

ita\_readme\_[role\_name].yml

e.g.)

Role name: mysql File name: ita\_readme\_mysql.yml

Role name: mysql/install File name: ita\_readme\_mysql%install.yml

※If role is in a deep directory hierarchy,

replacing "/" in the role name with "%" is required.

|

|—substitution table

Substitution table is defined for every role. Error doesn't occur even if the file doesn't exist

|

The naming rule of substitution table file name

|

ita\_translation-table\_[role\_name].txt

e.g.)

Role name: mysql File name: ita\_translation-table\_mysql.txt

Role name: mysql/install File name: ita\_translation-table\_mysql%install.txt

※If role is in a deep directory hierarchy,

replacing "/" in the role name with "%" is required.

|

| ITA does not concern if other directory or file exists other the directory or file above.

|—roles

Error occur during upload if the roles directory does not exist.

|

|—[role name①]

Error occur during upload if the role name directory does not exist.

|

Handle the directory containing tasks directory as a role.

|

Errors doesn't occur even if the directory hierarchy is deep.

|

```

|   |   |-readme.mc    ITA does not concern with the file.
|   |   |-tasks        The tasks directory is required.
|   |   |   |- main.yml   Error occur during upload if main.yml does not exist.
|   |   |   \- user_files  Files other than main.yml can be placed here.
|   |   |       \- user.yml  Files other than main.yml can be placed in subdirectory.
|   |
|   |   |-handlers      Doesn't concern if handlers directory exists or not.
|   |   |   |- main.yml   Doesn't concern if main.yml exists or not
|   |   |   \- user_files  Files other than main.yml can be placed here.
|   |   |       \- user.yml  Files other than main.yml can be placed in subdirectory.
|   |
|   |   |-templates     Doesn't concern if templates directory exists or not.
|   |   |   |- hosts.j2    Files can be placed in subdirectory.
|   |   |   \- user_files
|   |   |       \- user.j2
|   |
|   |   |-files         Doesn't concern if files directory exists or not.
|   |   |   |- sudoers     Doesn't concern if file and subdirectory exists or not.
|   |   |   |   The file content is not concerned.
|   |
|   |   |-vars          Doesn't concern if vars directory exists or not.
|   |   |   |- main.yml   Doesn't concern if file and subdirectory exists or not.
|   |   |   |   The file content is not concerned.
|   |
|   |   |-defaults      Doesn't concern if defaults directory exists or not.
|   |   |   |- main.yml   Doesn't concern if main.yml exists or not.
|   |   |   \- user_files  Files other than main.yml can be placed here.
|   |   |       \- user.yml  Files other than main.yml can be placed in subdirectory.
|   |
|   |   |-meta          Doesn't concern if meta directory exists or not.
|   |   |   |- main.yml   Doesn't concern if file and subdirectory exists or not.
|   |   |   |   The file content is not concerned.
|   |
|   |   |   ITA does not concern if other directory or file exists other the directory or file above.
|   |
|   \- [role name②]    There is no specific limit on the number of roles.

```

## (1) Master playbook

A master playbook created with ITA consists of a header section and a roles section.

### ① header section

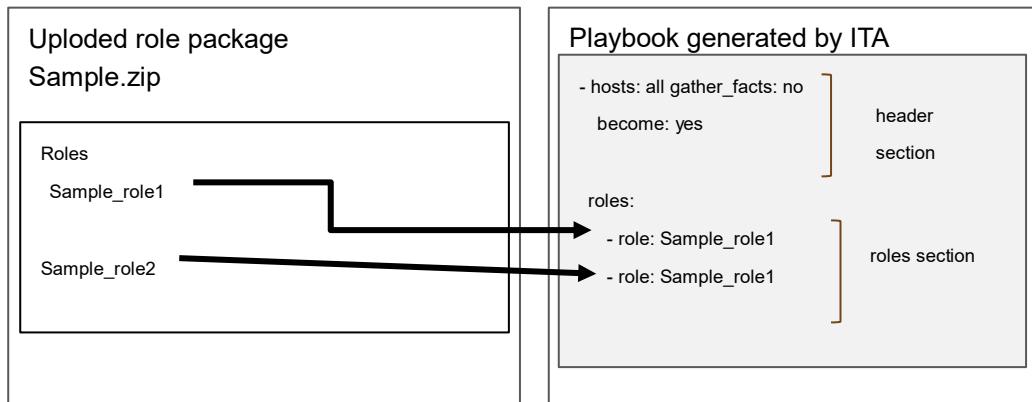
The default value of the header section is fixed, but you can change it in the header section of "5.3.2. Movement list".

Defult value of vector section

• For Ansible engine	• For AnsibleTower
- hosts: all	- hosts: all
remote_user: "{{ __loginuser__ }} "	gather_facts: no
gather_facts: no	become: yes
become: yes	

### ② roles section

Uploaded role package are included in the include order of "5.3.7 Movement details".



## (2) Unique management of variable name

The variable information registered in the substitution value list of ITA is handled as host variable. Variable names in all role packages of each drivers are uniquely managed.

When using same the variable name between roles with different variable structure, error will occur during upload

For example, in the case that the normal variable and nested variable or the nested structure is different between nested variables, etc.

## (3) ITA original specification of default variable definition file (defaults-> main.yml)

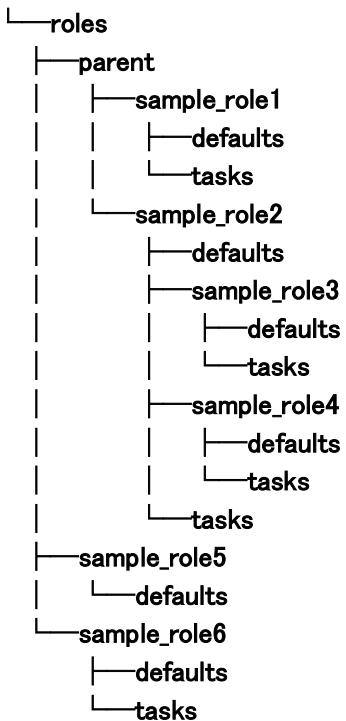
There is ITA original specification of the description (variable definition) of default variable definition.

Please refer to the attachment file "User Instruction Manual - Ansible-driver attachment Ansible usage guideline with additional rules" for details.

## (4) Notes on subdirectories of a predetermined directory in Ansible Role Directory Structure

If a subdirectory that has the same name of the predetermined directory is created under the predetermined directory in Ansible Role Directory structure (For example, a "files" directory under the "files" directory, etc.), error will occur during operation execution.

## (5) Points to note when the role name in the role package is set to the directory hierarchy, the following directory hierarchy role package will be explained as an example.



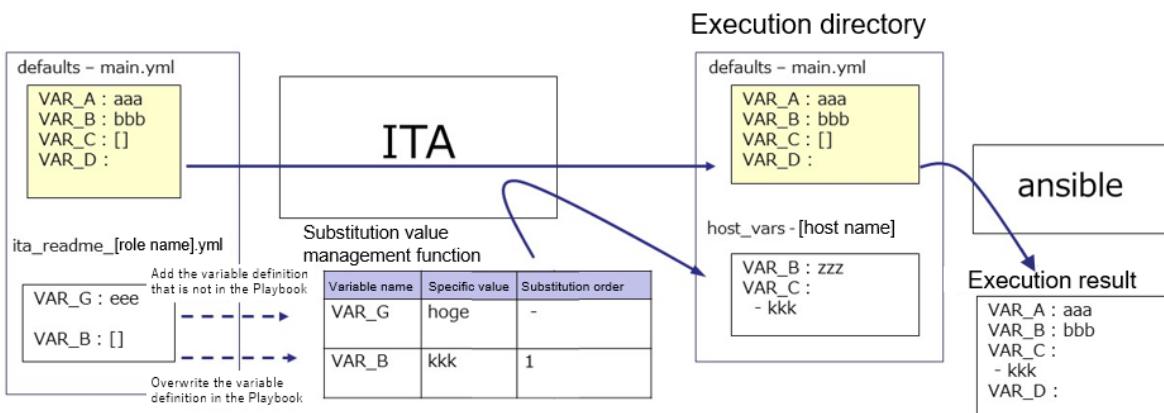
- ① The directory recognized as a role is the directory containing the tasks directory.  
In this example. The directory hierarchy (role name) handled by role is the following three.
  - parent/sample\_role1
  - parent/sample\_role2
  - sample\_role6
- ② Expection of directory hierarchy with multiple tasks directories  
Parent/sample\_role2/sample\_role3 and parent/sample\_role2/sample\_role4 also have tasks directories, Since there is a tasks directory in parent/sample\_role2 and it is recognized as a role, but it is not treated as a role.

## 6.4 Describe ITA readme(Ansible-Legacy Role only)

The substitution value management function interprets the variable type defined in defaults variable definition file and sets the variable value of each variable and its' member variable.

In the cases such as not wanting to define variable directly in the Playbook, etc. if variable is not defined in defaults variable definition file, variable value can be specified in the substitution value management function by setting the variable definition in ITA readme file.

The relation between ITA readme file and substitution value management function is as the following figure.



If the variable that is not in Playbook is defined in ITA readme file, the value of defined variable can be set in the substitution value management function.

Furthermore, if a type different from the variable in the Playbook is defined in the ITA readme file, the value can be registered in the substitution value management function with the overwritten variable type.

The value set in the substitution value management function is output to the variable definition file (`host_vars`) of each host, and is executed on each host by using the original Playbook and variable definition file as input in Ansible.

ITA readme file is only used to provide variable definition to substitution value management function, the variable and variable value defined in ITA readme don't affect the execution of Ansible.

It is optional to create ITA readme. If the variable definition in ITA readme and defaults variable definition file overlaps, the following rules will be used to handle the situation.

**Table 6.4-1 variable adoption rule**

<b>defaults variable definition file</b>	<b>ITA readme</b>	<b>Adoption source of variable definition</b>
defined	undefined	defaults variable definition file
undefined	defined	ITA readme
defined	defined	ITA readme

In addition, the default value displayed in "5.3.11 Substitution value list" is processed following the rule below.

**Table 6.4-2 default value display rule**

<b>defaults variable definition file</b>	<b>ITA readme</b>	<b>The method to handle the default value</b>
defined	undefined	Adopt the defaults variable definition file.
undefined	defined	Handle as no default value.
defined	defined	Adopt the defaults variable definition file. However, the rule is applied only when the definition variable matches. If the variable definition doesn't match, the variable is handled as no default value.

ITA readme is separated from role package during work execution.

The variable and specific value registered in ITA readme can't be applied.

## 6.5 Describe substitution table(Ansible-Legacy Role only)

The substitution table is a file set for making setting the specific value of variable other than "VAR\_xxx" defined in defaults variable definiton file or ITA readme in "5.3.11 Substitution value list" possible.

Define the link between the "arbitrary variable" defined in the default variable definition file or ITA readme with the "substitution variable" handled in substitution management function.

### (1) The format of substitution table

The format is as follows in text format.

The combination of substitution variable and arbitrary variable has to be unique within single role.

Substitution variable(\$s\*):(\$s+)arbitrary variable

substitution variable:LCA\_\*\*\*

\*\*\* :half-width alphanumeric character and underscore(\_) can be used.

(Minimum length: 1byte, Maximum length: 256 bytes)

Arbitrary variable: (Minimum length: 1byte, Maximum length: 256 bytes)

(\$s\*): More than 0 half-width space

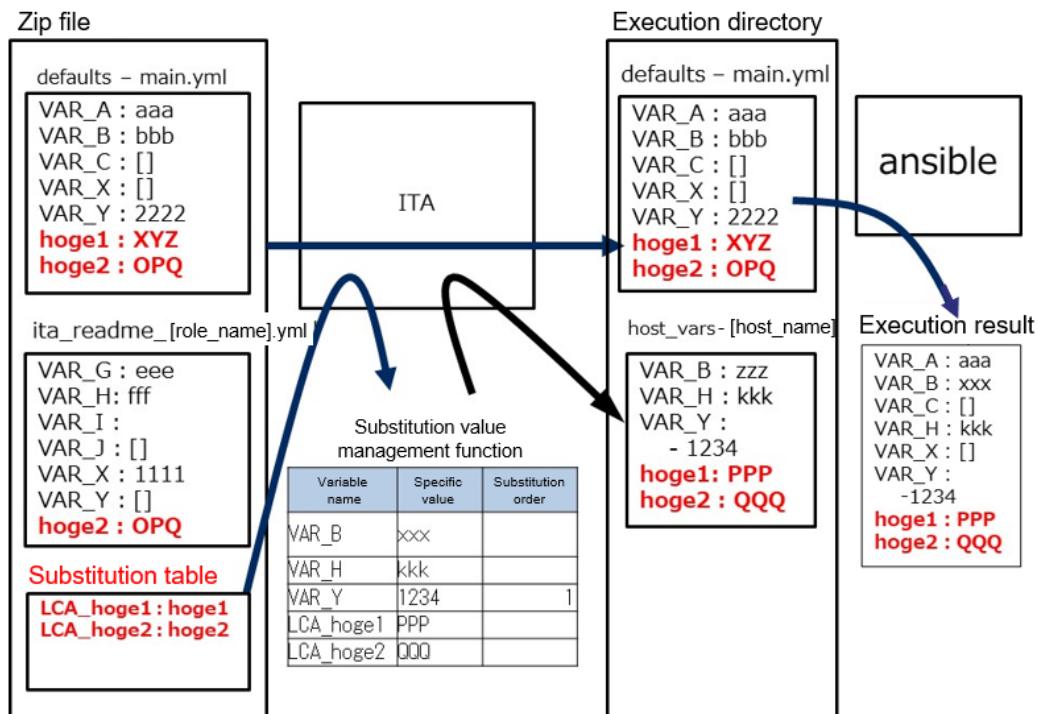
(\$s+): More than 1 half-width space

e.g.)

LCA\_var1: var1

# The line starts with # is comment line

The relationship of substitution value management function is as the following figure.

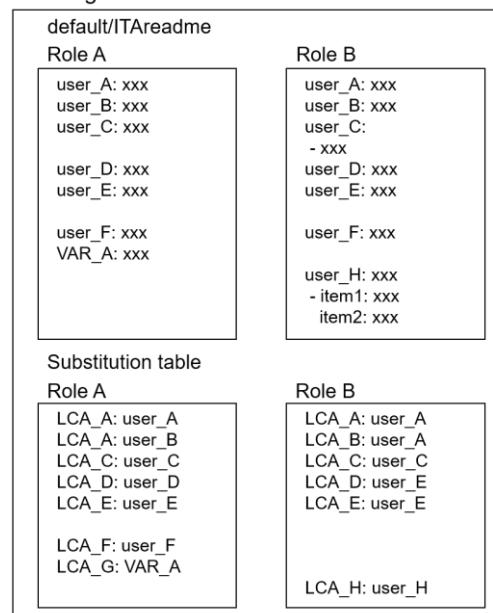


## (2) Notes

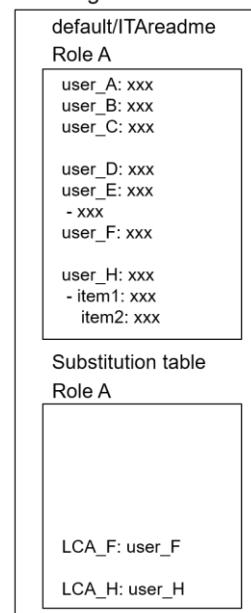
Listing the notes when creating substitution table.

Case	The behavior of ITA	Remarks
The substitution table exists but defaults variable definition file and ITA readme doesn't exist. (For every role)	Translation table can't be read.	
The variable begins with VAR_ is defined as arbitrary variable	Error occurs when uploading role package.	
Uses any variables not defined in the defaults variable definition file and ITA readme	Error occurs when uploading role package.	
Definition of substitution variables are duplicated in the role	Error occurs when uploading role package.	Package A->Role A LCA_A: user_A/LCA_A: user_B
Definition of arbitrary variables are duplicated in the role	Error occurs when uploading role package.	Package A->Role B LCA_A: user_A/LCA_B: user_A
The structure of arbitrary variable differs between roles	Error occurs when uploading role package.	Package A->Role A/B LCA_C: user_C
The combination of substitution variable and arbitrary variable is not unique in role package	Error occurs when uploading role package.	Package A Role A LCA_D: user_D Role B LCA_D: user_E
The structure of arbitrary variable differs between role packages	Error doesn't occurs when uploading role package, but the substiution variable is not displayed in substitution value list.	Package A->Role A LCA_F: user_F Package B->Role A LCA_F: user_F
Nested arbitrary variable is defined between role packages	The nested structure is the same so error does not occur but the setting of nested iteration count is common setting for each package.	Package A->Role B LCA_H: user_H Package B->Role A LCA_H: user_H

Package A



Package B



## 6.6 BackYard contents

### (1) Variable auto registration

When the variable analysis target file is uploaded, extract variable from the uploaded file.

**Table 6.6-1 Variable handling of the file uploaded in each mode**

Menu	Legacy	Legacy Role	Pioneer
Playbook files	○	×	×
Role package list	×	○	×
Dialog files	×	×	○

The extract timing depends on the startup cycle of the automatic process.

#### ※Unique management of variable name

The extracted variable names in all role packages of each mode are uniquely managed.

Since the variable structure is defined in the default variable definition file, the notes when the variable structure is different in each file are as follows

- Single role package

When using same the variable name between roles with different variable structure.

※In the case that the normal variable and nested variable or the nested structure is different between nested variables, etc.

=> Error occurs during file upload.

- All role package

When using same the variable name between role packages with different variable structure.

=> Error occurs during file upload.

### (2) Substitution value auto-registration setting

The Operation of parameter sheet and Movement linked with the item setting value of every host and the variable information are reflected to Target host menu and Substitution value list menu as the association target.

The reflection timing also depends on the startup cycle of the automatic process as described above.

Target host and substitution value list menu is updated by multiple operators. Reflection will not be performed if the last updater is other operator (not Backyard).

When user wants to reflect the data in substitution value auto-registration setting menu, please perform operations such as discard the applicable record in substitution value list value or disable the applicable record in other BackYard process.

The reflection rules to Target host menu and Substitution value list menu are as follows.

- ① When reflecting the information registered in substitution value auto-registration setting to substitution value list.

The status of substitution value list	Doesn't exist applicable record	Exist applicable record			Applicable record is being discarded	
		=	≠ Specific value			
		Specific value	Last updated by	BackYard process		
Reflection in substitution value list	Add new record	-	Update the specific value of the applicable record	-	Restore the discarded record	

※Applicable record: The record that has same Operation+host+Movement+variable name + (member variable)+(include order)

- ② When reflecting the information that is not registered in "substitution value auto-registration setting" menu (registered only in "substitution value" menu) to substitution value list

The status of substitution value list	Exist applicable record		
	Last updated by		
	BackYard process	Other operators	
Reflection in substitution value list	Discard the applicable record	-	-

- ③ When reflecting the information registered in substitution value auto-registration setting to target host menu

The status of operation target host	Doesn't exist applicable record	Exist applicable record	Applicable record is being discarded
Reflection in operation target host	Add new record	-	Restore the discarded record

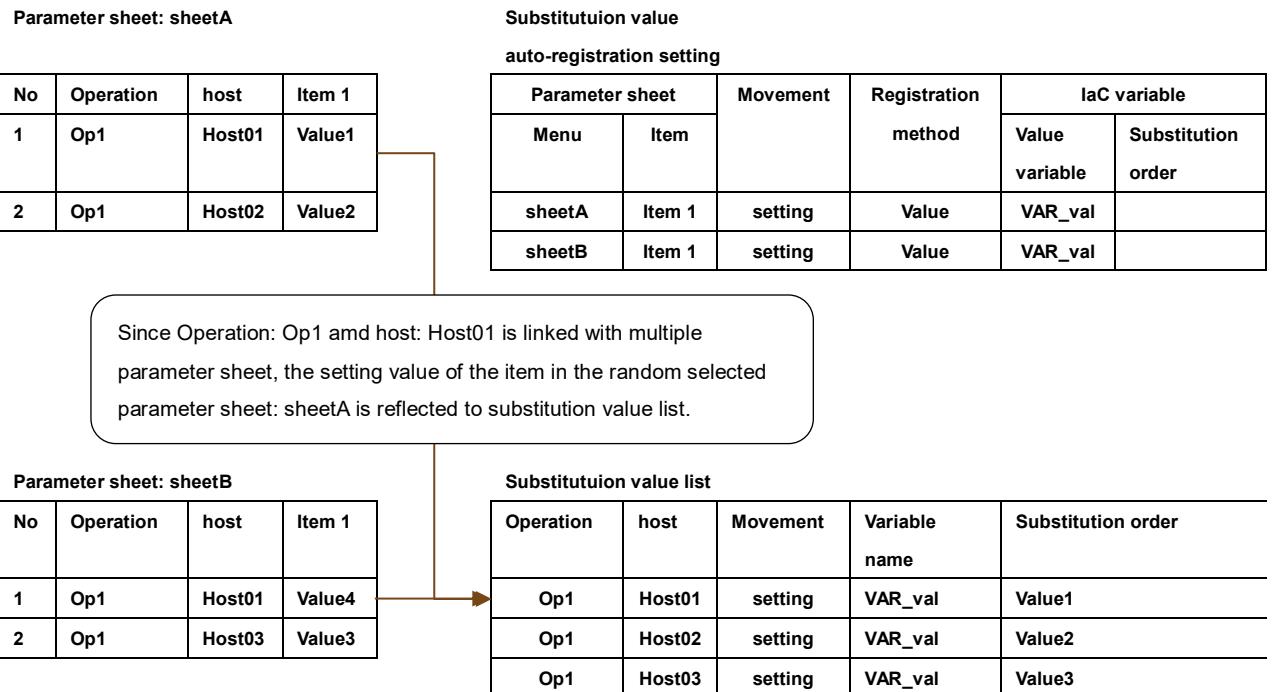
※ Applicable record: The record that has same Operation+host+Movement

- ④ When reflecting the information that is not registered in substitution value auto-registration setting(only registered in target host menu) to target host menu

The status of substitution value list	Exist applicable record		
	Last updated by		
	BackYard process	Other operators	
Reflection in substitution value list	Discard the applicable record	-	-

- ⑤ When link to multiple items is registered to for the same Movement, variable, and substitution order.

When multiple parameter sheets are linked with same operation and host, one item is selected randomly and reflected to substitution value list.



## **6.7 Ansible usage guideline ITA additional rules**

A Playbook creation guideline for using ITA to execute on Ansible.

Please refer to the attachment "User Instruction Manual - Ansible-driver attachment- Ansible usage guideline with additional rules" for details.

## 7 Application operation

The operation to utilize ITA system contains not only inputs by user from the browser screen of client PC but also operations according to system operation and maintenance. The available operation and maintenance are as follows.

### 7.1 Maintenance

The required file to start/stop/restart Ansible driver independent processes are as follows.

Description	Target file name
Legacy/pioneer/legacyRole execution monitor Execute the unexecuted Operation.	ky_legacy_execute-workflow.service
legacy variable automatic registration Extract variable from the uploaded file.	ky_legacy_varsautolistup-workflow.service
legacy automatic registration setting Reflect the information set in auto-registration setting to substitution value list and operation target host menu.	ky_legacy_valautostup-workflow.service
pioneer automatic registration setting Reflect the information set in auto-registration setting to substitution value list and operation target host menu.	ky_pioneer_valautostup-workflow.service
legacyRole variable automatic registration Extract variable from the uploaded file.	ky_legacy_role_varsautolistup-workflow.service
legacyRole automatic registration setting Reflect the information set in auto-registration setting to substitution value list and operation target host menu.	ky_legacy_role_valautostup-workflow.service
AnsibleTower data synchronization Obtain setting information from Ansible-tower.	ky_ansible_towermasterSync-workflow.service

The target files are stored in “/usr/lib/system/system”.

The method to start/stop/restart process are as follows.

Please execute the command with root privilege.

- ① Start process

```
# systemctl start ky_legacy_execute-workflow.service
```

- ② Stop process

```
# systemctl stop ky_legacy_execute-workflow.service
```

- ③ Restart process

```
# systemctl restart ky_legacy_execute-workflow.service
```

Please substitute each target file name to start / stop / restart the process.

## 7.2 About the maintenance method

- ① Change to NORMAL level

Rewrite the 8<sup>th</sup> line of the following file from “DEBUG” to “NORMAL”.

Log level setting file: <installation directory>/ita-root/confs/backyardconfs/ita\_env

- ② Change to DEBUG level

Rewrite the 8<sup>th</sup> line of the following file from “NORMAL” to “DEBUG”.

Log level setting file: <installation directory>/ita-root/confs/backyardconfs/ita\_env

- ③ Change the startup cycle

Change the 5<sup>th</sup> parameter of ExecStart in each target file. (Unit: second)

Please use the default value of startup cycle excluding exceptions.

```
ExecStart=/exastro/ita-root/backyards/common/ky_lopcall-php-procedure.sh  
/usr/local/bin/php           /usr/local/bin/php           /exastro/ita-  
root/backyards/ansible_driver/ky_pioneer_varsautolistup-workflow.php   /exastro/ita-  
root/logs/backyardlogs 10 NORMAL > /dev/null 2>&1
```

After rewriting the file, the change takes effect after restarting the process.

Log file output destination: <installation directory>/ita-root/logs/backyardlogs

## 8 Appendix

---

### 8.1 The linkage between the input data used during Ansible execution and ITA menu

Extract information from each ITA menu and create the "input data" that is required for Ansible execution. At that time, the password in device list menu is encrypted. Ansible-Legacy and Ansible-LegacyRole encrypts with ansible-vault, while pioneer encrypts with the original method of ITA. The "Input data" can be downloaded from "[5.3.12 Check operation status](#)" in ZIP format. Executing the following command in Ansible directly is also possible.

Ansible-Legacy

```
ansible-playbook (option) -i hosts --vault-password-file password_file playbook.yml
```

Ansible-Pioneer

```
ansible-playbook (option) -i hosts playbook.yml
```

Ansible-LegacyRole

```
ansible-playbook (option) -i hosts --vault-password-file password_file site.yml
```

The password file name is at user's desire. Please set the password to the password file with the value of the content in "ITA installation target directory/ita-root/confs/commonconfs/ansible\_vault\_accesskey.txt", decoded in the order of rot13, base64.

The relationship between various data and the ITA menu is as follows.

### 8.1.1 Ansible-Legacy input data

#### 【Parent directory】

child_playbooks	The directory containing user created Playbooks.		
	Ansible-Legacy	Playbook files	Playbook file
	Ansible-Legacy	Movement details	Include order
template_files	The directory containing the template file used in Playbook that is going to be executed.		
	Ansible-Legacy	Template list	Template file
	Ansible-Legacy	Movement details	Include order
copy_files	The directory containing the file that is going to be deployed on operation target server.		
	Ansible-Legacy	Contents list	Files
	Ansible-Legacy	Movement details	Include order
host_vars	The directory containing the host information of the operation target host and the definition file of various variable.		
	Ansible common	Global variable list	Variable name/specific value
	Ansible-Legacy	Substitution value list	Variable name/specific value
	Ansible-Legacy	Template list	Template file
	Ansible-Legacy	Contents list	File variable name
	Ansible-Legacy	Movement details	Include order
	Ansible-Legacy	Interface information	Data relay storage path(ITA)
	Ansible-Legacy	Interface information	Symphony instance data relay storage path(Ansible)
	Basic console	Device list	Protocol
	Basic console	Device list	Login user ID
	Basic console	Device list	Login password
	Basic console	Device list	※Encrypted with ansible-vault
	Basic console	Device list	Host name

ssh_key_files	The directory in which the specified ssh authentication key file is stored when using key authentication as the authentication method.		
	Basic console	Device list	ssh authentication key file
winrm_ca_files	The directory in which the file that defines the connection information when connecting to WinRM is stored.		
	Basic console	Device list	WinRM connection information
AnsibleExecOption.txt	Parameter for AnsiblePlaybook execution		
	Ansible common	Interface information	Option parameter
	Ansible-Legacy	Movement list	Number of parallel executions
hosts	The file describing the operation execution target host		
	Basic console	Device list	host name
	Basic console	Device list	IP address
	Basic console	Device list	Login user ID
	Basic console	Device list	Login password
	Basic console	Device list	※Encrypted with ansible-vault
	Basic console	Device list	Connection options ※The parameter of ansible_ssh_extra_args
	Basic console	Device list	ssh authentication key file
	Basic console	Device list	Server certificate
	Basic console	Device list	Inventory file additional option
playbook.yml	The file calls the whole information of playbook and host information and executes Ansible.		
	Ansible-Legacy	Playbook files	Playbook file
	Ansible-Legacy	Movement details	Include order
	Ansible-Legacy	Movement details	gather_facts

### 8.1.2 Ansible-Pioneer input data

【Parent directory】			
template_files	The directory containing the template file used in Playbook that is going to be executed.		
	{ Ansible-Pioneer      Template list      Template file Ansible-Pioneer      Movement details      Include order		
copy_files	The directory containing the file that is going to be deployed on operation target server.		
	{ Ansible-Pioneer      Contents list      Files Ansible-Pioneer      Movement details      Include order		
ssh_key_files	The directory in which the specified ssh authentication key file is stored when using key authentication as the authentication method.		
	{ Basic console      Device list      ssh authentication key file		
winrm_ca_files	The directory in which the file that defines the connection information is stored when connecting to WinRM.		
	{ Basic console      Device list      WinRM connection information		
host_vars	The directory in which the host information of the operation target host and the definition file of various variable is stored.		
	{ Ansible common      Interface information      Data relay storage path(ITA) Ansible common      Interface information      Symphony instance data relay storage path(Ansible) Ansible common      Global variable list      Variable name/specific value Ansible-Pioneer      Substitution value list      Variable name/specific value Ansible-Pioneer      Template list      Template file Ansible-Pioneer      Movement details      Include order Ansible-Pioneer      Contents list      File variable name Ansible-Pioneer      Movement details      Include order Basic console      Device list      Login password Basic console      Device list      ※Encrypted with ITA original method Basic console      Device list      Host name Basic console      Device list      Connection options Basic console      Device list      Protocol Basic console      Device list      Login user ID		
dialog_files	The directory in which user created dialog files is stored.		

	Ansible-Pioneer	Dialog files	Dialog file																		
	Ansible-Pioneer	Movement details	Include order																		
AnsibleExecOption.txt	Parameter for AnsiblePlaybook execution.																				
	<table border="0"> <tr> <td>Ansible common</td> <td>Interface information</td> <td>Option parameter</td> </tr> </table>			Ansible common	Interface information	Option parameter															
Ansible common	Interface information	Option parameter																			
hosts	The file describing the operation execution target host																				
	<table border="0"> <tr> <td>Basic console</td> <td>Device list</td> <td>Host name</td> </tr> <tr> <td>Basic console</td> <td>Device list</td> <td>IP address</td> </tr> <tr> <td>Basic console</td> <td>Device list</td> <td>Login user ID</td> </tr> <tr> <td>Basic console</td> <td>Device list</td> <td>Login password</td> </tr> <tr> <td>Basic console</td> <td>Device list</td> <td>※Encrypted with ITA original method ssh authentication key file</td> </tr> <tr> <td>Basic console</td> <td>Device list</td> <td>Inventory file additional option</td> </tr> </table>			Basic console	Device list	Host name	Basic console	Device list	IP address	Basic console	Device list	Login user ID	Basic console	Device list	Login password	Basic console	Device list	※Encrypted with ITA original method ssh authentication key file	Basic console	Device list	Inventory file additional option
Basic console	Device list	Host name																			
Basic console	Device list	IP address																			
Basic console	Device list	Login user ID																			
Basic console	Device list	Login password																			
Basic console	Device list	※Encrypted with ITA original method ssh authentication key file																			
Basic console	Device list	Inventory file additional option																			
playbook.yml	The file calls the whole information of playbook and host information and executes Ansible.																				
	<table border="0"> <tr> <td>Ansible-Pioneer</td> <td>Interface information</td> <td>Data relay storage path(ITA)</td> </tr> </table>			Ansible-Pioneer	Interface information	Data relay storage path(ITA)															
Ansible-Pioneer	Interface information	Data relay storage path(ITA)																			

### 8.1.3 Ansible-LegacyRole input data

【Parent directory】		
copy_files	The directory in which the file that is going to be deployed on operation target server is stored.	
	Ansible-LegacyRole	Contents list
	Ansible-LegacyRole	Movement details
		Files
		Include order
roles	The directory containing the user created role.	
	Ansible-LegacyRole	Role package list
		Role package file (ZIP format)
ssh_key_files	The directory in which the specified ssh authentication key file is stored when using key authentication as the authentication method.	
	Basic console	Device list
		ssh authentication key file
winrm_ca_files	The directory in which the file that defines the connection information when connecting to WinRM is stored.	
	Basic console	Device list
		WinRM connection information
host_vars	The directory containing the host information of the operation target host and the definition file of various variable.	
	Ansible common	Interface information
	Ansible common	Interface information
	Ansible common	Global variable list
	Ansible-LegacyRole	Substitution value list
	Ansible-LegacyRole	Template list
	Ansible-LegacyRole	Movement details
	Ansible-LegacyRole	Contents list
	Ansible-LegacyRole	Movement details
	Basic console	Device list
		Data relay storage path(ITA)
		Symphony instance data relay storage path(Ansible)
		Variable name/specific value
		Variable name/specific value
		Template file
		Include order
		File variable name
		Include order
		Protocol
		Login user ID
		Login password
		※Encrypted with ansible-vault
		Host name

AnsibleExecOption.txt Parameter for AnsiblePlaybook execution.

Ansible common	Interface information	Option parameter
Ansible-LegacyRole	Movement list	Number of parallel executions

hosts The file describing the operation execution target host.

Basic console	Device list	host name
Basic console	Device list	IP address
Basic console	Device list	Login user ID
Basic console	Device list	Login password
Basic console	Device list	※Encrypted with ansible-vault
Basic console	Device list	Connection options
Basic console	Device list	※The parameter of ansible_ssh_extra_args
Basic console	Device list	ssh authentication key file
Basic console	Device list	Server certificate
Basic console	Device list	Inventory file additional option

site.yml The file calls the whole information of playbook and host information and executes Ansible.

Ansible-Legacy	Playbook files	Playbook file
Ansible-Legacy	Movement details	Include order
Ansible-Legacy	Movement details	gather_facts

## 8.2 Result data created during Ansible execution

The result of executing [Input data] with ansible is saved as [Result data] in ZIP format. [Result data] can be downloaded in ZIP format from "5.3.12 Check operation status".

### 8.2.1 Legacy/LegacyRole List of files saved in result data

Table 8.2.1-1 Legacy/LegacyRole List of files where result data is saved

File name	Record content	In Ansible Engine	In Ansible Tower
result.txt	Record the execution checking of Ansible	<input type="radio"/>	
xxx.pid	A file that records the process ID of the Ansible-playbook command. xxx: Process ID of the Ansible-playbook command	<input type="radio"/>	
error.log	Error output destination file when ITA outputs an error message due to some error when executing work, or when the Ansible-playbook command outputs an error message due to some error. The contents displayed in the error log of work execution confirmation.	<input type="radio"/>	<input type="radio"/>
exec.log	A log file that is a partial processing of the execution log output by Ansible-playbook. Contents displayed in the execution log of work execution confirmation.	<input type="radio"/>	<input type="radio"/>
exec.log.org	Execution log output by ansible-playbook	<input type="radio"/>	<input type="radio"/>
Ita_<mode name>_executions_jobtpl_<execution no.>_<group no.>	This is a divided execution log file. For the naming convention of the file name, please refer to ⑥ Execution log display of 5.3.12 Check operation status.		<input type="radio"/>
forced.txt	Record file in emergency stop	<input type="radio"/>	
user_files	The directory where the file is recorded when some kind of file is output to the ITA original variable 「__workflowdir__」 in the executed playbook.	<input type="radio"/>	<input type="radio"/>

## 8.2.2 List of files saved in Pioneer result data

Table 8.2.1-2 Pioneer list of files where result data is saved

File name	Record content	In Ansible Engine	In Ansible Tower
result.txt	Record the execution result of Ansible	<input type="radio"/>	
xxx.pid	A file that records the process ID of the Ansible-playbbok command. xxx: Process ID of the Ansible-playbbok command	<input type="radio"/>	
error.log	Error output destination file when ITA outputs an error message due to some error when executing work, or when the Ansible-playbbok command outputs an error message due to some error. The contents displayed in the error log of work execution confirmation.	<input type="radio"/>	<input type="radio"/>
exec.log	A log file that is a partial processing of the execution log output by Ansible-playbbok. Contents displayed in the execution log of work execution confirmation.	<input type="radio"/>	<input type="radio"/>
exec.log.org	Execution log output by ansible-playbook	<input type="radio"/>	<input type="radio"/>
Ita_<mode name>_executions_jobtpl_<execution no.>_<group no.>	This is a divided execution log file. For the naming convention of the file name, please refer to ⑥ Execution log display of 5.3.12 Check operation status.		<input type="radio"/>
forced.txt	Record file in emergency stop	<input type="radio"/>	
user_files	The directory where the file is recorded when some kind of file is output to the ITA original variable <code>__workflowdir__</code> in the executed playbook.	<input type="radio"/>	<input type="radio"/>

(1) Legacy-Role

Table 9.2.1-3 Legacy-Role list of files where result data is saved

File name	Record content	In Ansible Engine	In Ansible Tower
result.txt	Record the execution result of Ansible	<input type="radio"/>	
xxx.pid	A file that records the process ID of the Ansible-playbook command. xxx: Process ID of the Ansible-playbook command	<input type="radio"/>	
error.log	Error output destination file when ITA outputs an error message due to some error when executing work, or when the Ansible-playbook command outputs an error message due to some error. The contents displayed in the error log of work execution confirmation.	<input type="radio"/>	<input type="radio"/>
exec.log	A log file that is a partial processing of the execution log output by Ansible-playbook. Contents displayed in the execution log of work execution confirmation.	<input type="radio"/>	<input type="radio"/>
exec.log.org	Execution log output by ansible-playbook	<input type="radio"/>	<input type="radio"/>
Ita_<mode name>_executions_jobtpl_<execution no.>_<group no.>	This is a divided execution log file. For the naming convention of the file name, please refer to ⑥ Execution log display of 5.3.12 Check operation status.		<input type="radio"/>
forced.txt	Record file in emergency stop	<input type="radio"/>	
user_files	The directory where the file is recorded when some kind of file is output to the ITA original variable 「__workflowdir__」 in the executed playbook.	<input type="radio"/>	<input type="radio"/>