



IT Automation

Collect/Compare function 【Practice】

※In this document, “IT Automation” will be written as “ITA”,

Table of contents

Introduction

- [\(1\) About this document](#)
- [\(2\) Operation environment](#)
- [\(3\) Scenario](#)

1. Scenario 1 【Collect function】 Collect target host OS Information

[Scenario 1 Overall diagram](#)

[1.1 Register target host](#)

[1.2 Register operation](#)

[1.3 Register movement](#)

[1.3.1 Header section and gather facts](#)

[1.4 Register Playbook](#)

[1.4.1 Directory for YAML files and collection](#)

[1.5 Movement-Playbook link](#)

[1.6 Register target host](#)

[1.7 Create Parameter sheet for registering collected values](#)

[1.8 Register Collected item value list](#)

[1.9 Register Collect interface information](#)

[1.10 Run operation](#)

[1.11 Confirm the collection results](#)

2. Scenario 2 【Compare function】

Compare the values and the expected values of the one collected in Scenario 1.

[Scenario 2 Overall diagram](#)

[2.1 Register Operation](#)

[2.2 Create parameter sheet for expected values](#)

[2.3 Register expected values](#)

[2.4 Register a Comparison](#)

[2.5 Run comparison](#)

[【Reference】 Compare details](#)

[【Reference】 \(1\) Register Comparison details](#)

Table of contents

[\[Reference\] \(2\) Register Compare details](#)

[\[Reference\] \(3\) Run Comparison](#)

3. Scenario 3 **【Collect function】** Collect the target host's SSL certificate file

[Scenario 3 Overall diagram](#)

[3.1 Register Target host](#)

[3.2 Register operation](#)

[3.3 Register Movement](#)

[3.4 Register Playbook](#)

[3.4.1 File collection directory](#)

[3.5 Movement-Playbook link](#)

[3.6 Register File name](#)

[3.7 Register File name](#)

[3.8 Register substitution value auto-registration settings](#)

[3.9 Create Parameter sheet for collect values](#)

[3.10 Register Collected item value list](#)

[3.11 Run operation](#)

[3.12 Confirm collection results](#)

4. Scenario 4 **【Compare function】**

Compare the file downloaded in scenario 3 with the same file from a different date.

[Scenario 4 Overall diagram](#)

[4.1 Register operation](#)

[4.2 Prepare SSL certificate with different contents](#)

[4.3 Register file name](#)

[4.4 Run operation](#)

[4.5 Confirm comparison results](#)

[4.6 Register Comparison definition](#)

[4.7 Run Comparison](#)

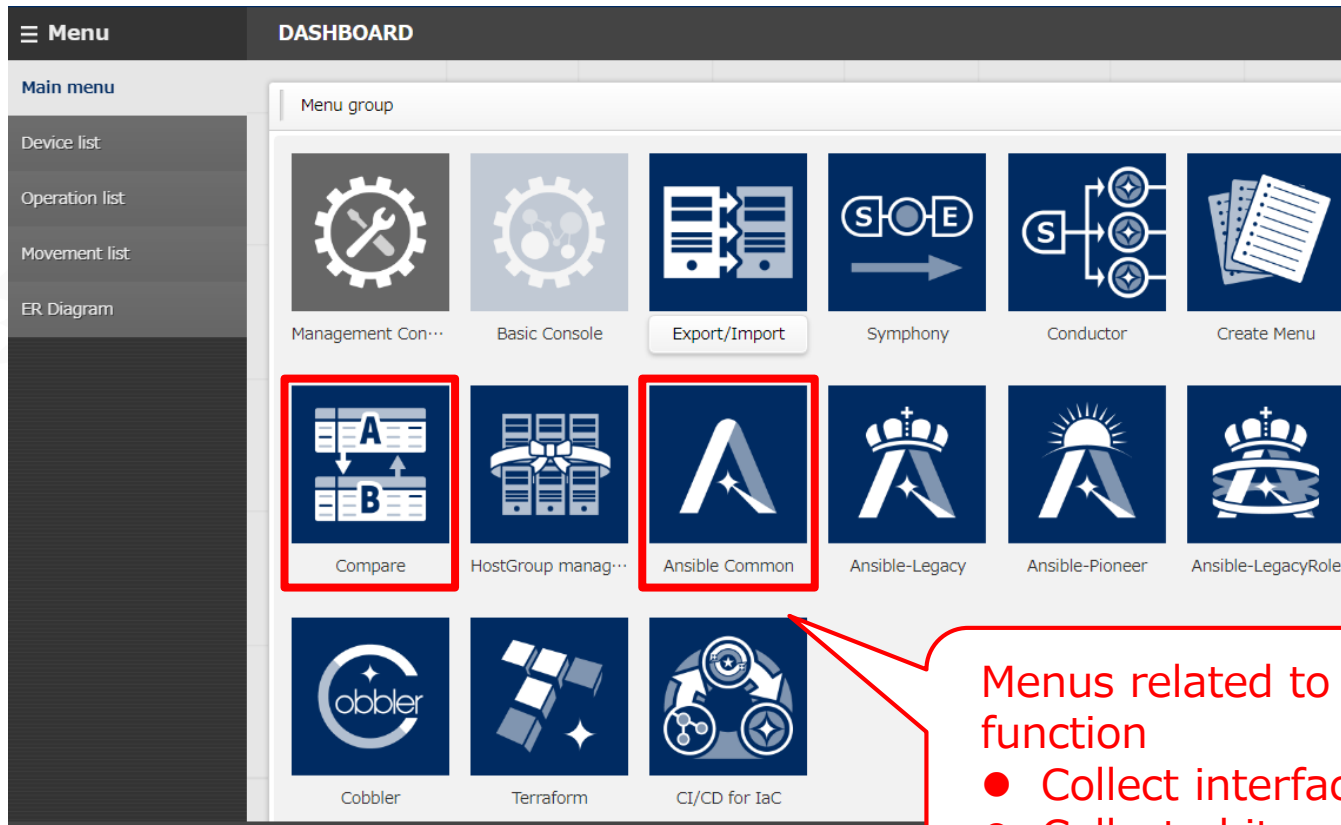
Introduction



(1) About this document

About this document

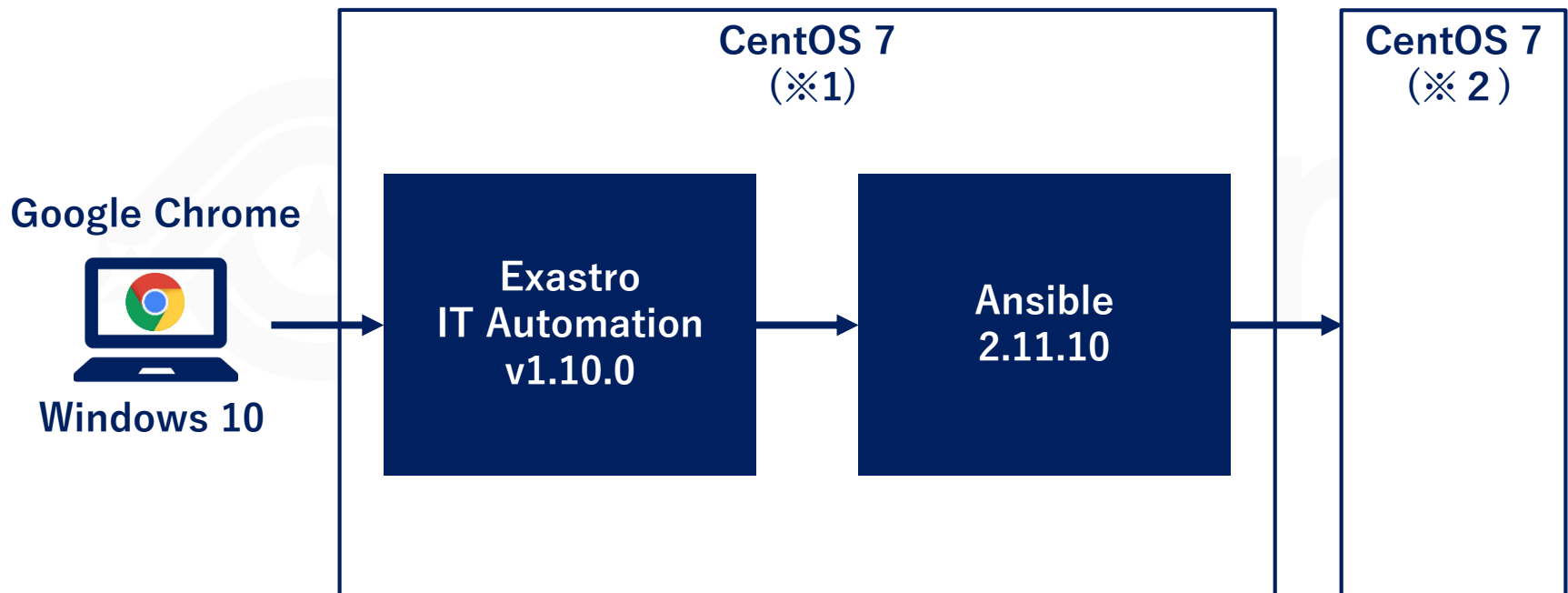
This document aims to teach the user about the Compare and Collect function by leading them through a hands-on scenario.



(2) Operation environment

Environment

The environment used in this document is as follows



※1 ITA can be installed on RHEL7 and RHEL8 type OS.

※2 You can use any OS as long as it is compatible with Ansible.

(3) Scenario

Comparing and collecting parameters and files

- Scenario 1 and 2 will be used to collect and compare parameters.
Scenario 3 and 4 will be used to collect and compare files.
- More specifically, we will be collecting/comparing the following information:
Parameter: OS Information
File: SSL Certificate

	Collect function	Compare function
Collecting/ Comparing parameters	Scenario 1 Collect the target host OS information	Scenario 2 Compare the values and the expected values of the one collected in Scenario 1.
Collecting/ Comparing files	Scenario 3 Collect the target host's SSL certificate file	Scenario 4 Compare the file downloaded in scenario 3 with the same file from a different date.

1. Scenario 1 【Collect function】 Collect target host OS Information

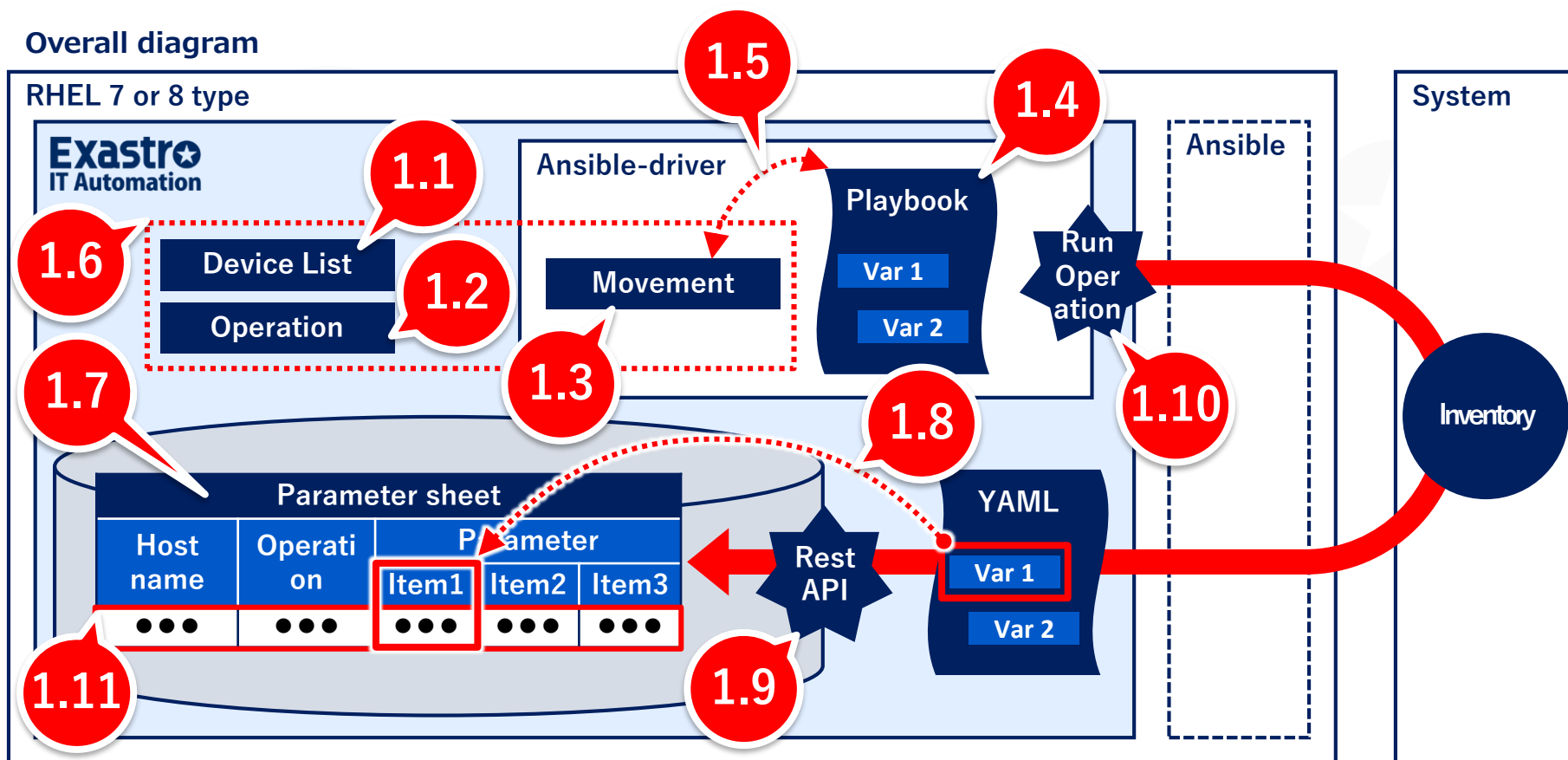


Scenario 1 Overall diagram

Scenario 1 workflow

- The numbers in the diagram below indicates the different chapters in this document.
- After configuring the different settings, we will start the operation and collect the inventory (OS info) , where it will be automatically registered to a parameter sheet.

Overall diagram



1.1 Register target host

Register the target host connection information

Go to "Device list" and start the registration

Menu : **Basic console > Device list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

The screenshot shows a registration form with several fields. Red boxes highlight the following fields: HW device type (SV), Host name (targethost), IP address (192.0.2.1), Login user ID (root), Management (radio button), Login password (masked with asterisks), and Authentication method (Password authentication). A blue dashed arrow points to the Register button.

HW device type	Host name (Free space)	IP address (Depends on your environment)	Login user ID (Depends on your environment)	Login password		Ansible dedicated information
				Management	Login password (Depends on your environment)	Dedicated information for Legacy/Role
SV	targethost	192.0.2.1	root	●	*****	Password authentication

1.2 Register operation

Register the operation we will use in Scenario 1.

In ITA, we call automated operation units for “Operations”.

Hereinafter, we will link all the necessary data to this operation.

Menu : **Basic Console > Operation list**

- ① Press “Register” -> “Start registration.
- ② Input the following information and press the “Register” button.

No.	Operation ID	Operation name*	Scheduled date for execution*
Auto-input	Auto-input	GatherFacts1	<input type="text" value="2021/04/22 17:09"/>

Operation name (Free space)	Scheduled date for execution (Free space)
GatherFacts1	2021/04/22 17:09

You can name the operation to whatever you want.

This item indicates the planned date and time for the operation. It is not a timer and will therefore not automatically run the operation after the specified time has passed.

1.3 Register movement

Register Movement in Ansible-Legacy

In ITA, the smallest automatic operation unit (A.K.A a “job”) is called a “Movement”.

After this, we will link a Playbook to it, making it a Movement that collects OS Information.

Menu : **Ansible-Legacy > Movement list**

- ① Press “Register” -> “Start registration.”
- ② Input the following information and press the “Register” button.

Movement ID	Movement Name *	Delay timer	Host specific format *	WinRM connection	Dedicated information for ansible
Auto-input	Gatherfacts		IP		<pre>- hosts: all remote_user: "{{ __loginuser__ }}" gather_facts: yes become: yes</pre>

Movement name (Free space)	Ansible user information	
	Host specific format	Header section
GatherFacts	IP	<pre>- hosts: all remote_user: "{{ __loginuser__ }}" gather_facts: yes become: yes</pre>

This activates “gather_facts”.
※ For more information, see [chapter 『1.3.1 Header section and gather facts』](#)

1.3.1 Header section and gather_facts

Activate gather_facts

The Ansible's Playbook Header section's "gather_facts" is deactivated by default when installing ITA.

- In this scenario, we will use "gather_facts" to collect the OS information, so we will need to input the following under the header section and activate it.
- If you don't need to change the default value, you can leave the header section blank.

Default

```
- hosts: all
  remote_user: "{{ __loginuser__ }}"
  gather_facts: no
  become: yes
```



Set gather_facts to "yes"

```
- hosts: all
  remote_user: "{{ __loginuser__ }}"
  gather_facts: yes
  become: yes
```

Input all the necessary lines to the header section and change this value.

1.4 Register Playbook (1/3)

Register Playbook for operations (1/2)

- The playbook that we will register in this guide contains a workflow which consists of "Generate YAML file containing gathered OS information under the Target host's /tmp/ directory" → Copy the generated YAML file to the ITA host server's collection directory".
- For more information regarding the YAML file and the Collect directory, please see Chapter [1.4.1 Directory for YAML files and collection](#).



1.4 Register Playbook (2/3)

Register Playbook for operations (1/2)

File name : GatherFacts.yml

```
- name: make yaml file
blockinfile:
  create: yes
  mode: 0644
  insertbefore: EOF
  marker: ""
  dest: "/tmp/gatherfacts.yml"
  content: |
    ansible_architecture      : {{ ansible_architecture }}
    ansible_bios_version      : {{ ansible_bios_version }}
    ansible_default_ipv4__address  : {{ ansible_default_ipv4.address }}
    ansible_default_ipv4__interface : {{ ansible_default_ipv4.interface }}
    ansible_default_ipv4__network  : {{ ansible_default_ipv4.network }}
    ansible_distribution       : {{ ansible_distribution }}
    ansible_distribution_file_path : {{ ansible_distribution_file_path }}
    ansible_distribution_file_variety : {{ ansible_distribution_file_variety }}
    ansible_distribution_major_version: {{ ansible_distribution_major_version }}
    ansible_distribution_release   : {{ ansible_distribution_release }}
    ansible_distribution_version  : {{ ansible_distribution_version }}
    ansible_machine              : {{ ansible_machine }}
    ansible_memtotal_mb           : {{ ansible_memtotal_mb }}
    ansible_nodename              : {{ ansible_nodename }}
    ansible_os_family             : {{ ansible_os_family }}
    ansible_pkg_mgr               : {{ ansible_pkg_mgr }}
    ansible_processor_cores       : {{ ansible_processor_cores }}

- name: copy the make yaml file to local
fetch:
  src: "/tmp/gatherfacts.yml"
  dest: "{{ __parameter_dir__ }}/{{ inventory_hostname }}"
  flat: yes
```

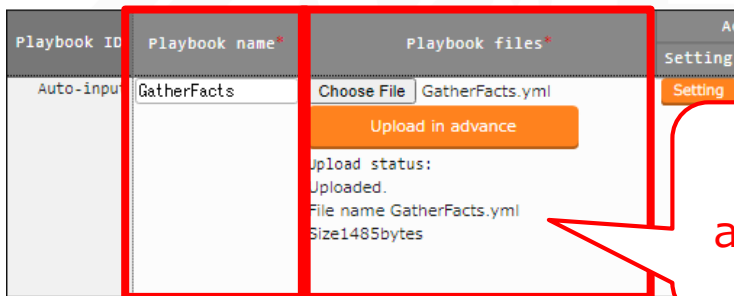
1.4 Register Playbook (3/3)

Register Playbook in Ansible-Legacy

Register the playbook we created in the last slide.

Menu: **Ansible-Legacy > Playbook file collection**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.



Make sure to press the "Register" button after you have pressed "Upload in advance".

Playbook file name (Free space)	Playbook file
GatherFacts	GatherFacts.yml

1.4.1 Directory for YAML files and collection (1/2)

Create directory for YAML files and collecting files.

- Since Collect result files in ITA is specified in a YAML file format, we need to create a YAML file.
- The generated YAML file will be stored in the Collection directory specified by the ITA reserved variables.

GatherFacts.yml
Line 7

```
dest: "{{ __parameter_dir__ }}/{{ inventory_hostname }}/"
```

This directory will be specified

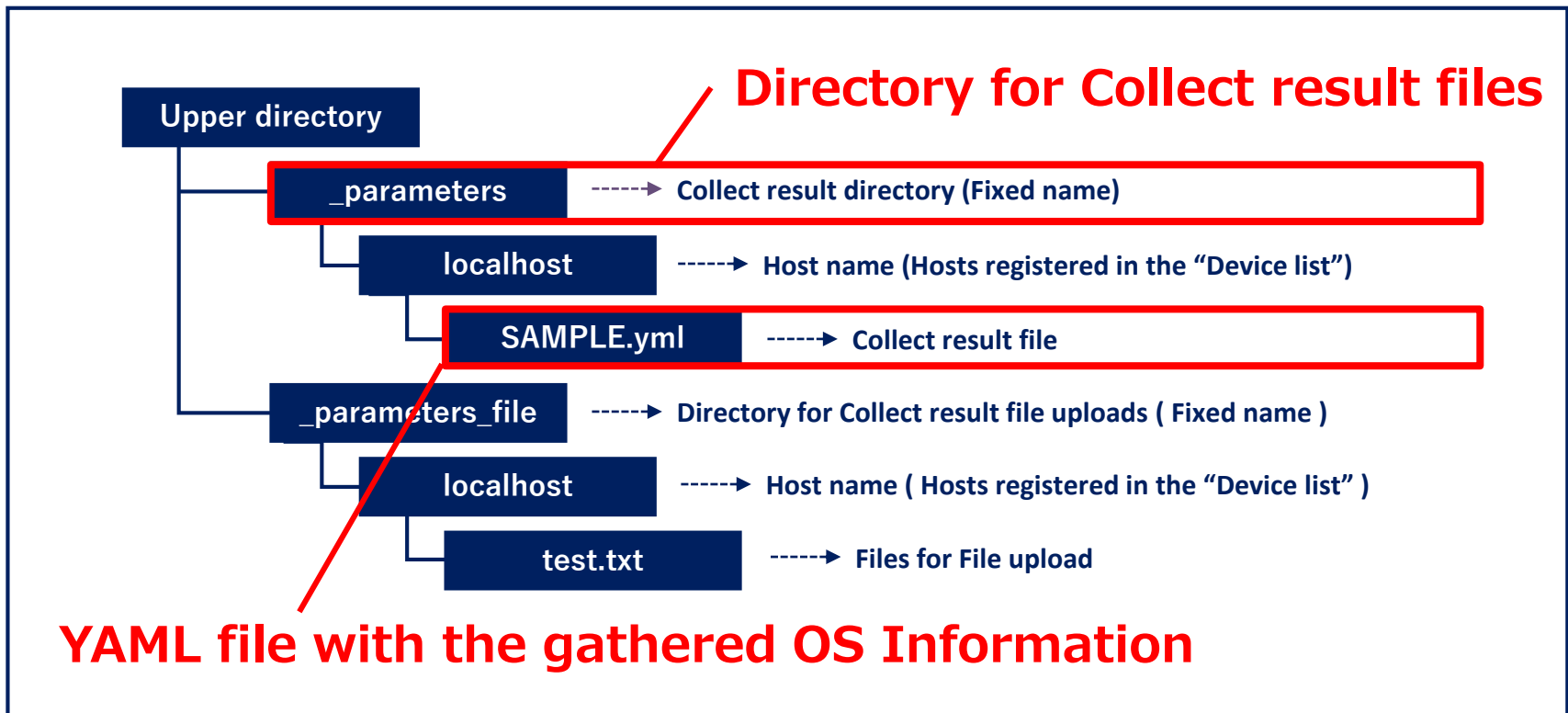
Path variables

	ITA reserved variable	Variable specified contents
Source file storage location	<code>__parameter_dir__</code>	"_parameters" path under the operation result directory
Collected file storage location	<code>__parameters_file_dir__</code>	"_parameters_file" path under the operation result directory

1.4.2 Directory for YAML files and collection (2/2)

The following figure displays the file hierarchy for the Collect file directory.

File hierarchy



1.5 Movement-Playbook link

Link Movement and Playbook

Link the previously registered Movement and Playbook.

Menu: **Ansible-Legacy > Movement-Playbook link**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

Associated item No.	Movement*	Playbook files*	Include order*
Auto-input	1:Gatherfacts	GatherFacts	1

Movement	Playbook file	Include order
GatherFacts	GatherFacts	1

The "Include order" specified the order in which the Playbook will be executed if there are multiple Playbooks linked to the Movement. In this scenario, we will only link 1 Playbook.

1.6 Register target host

Link Operation, Movement and the Target host.

Link the previously registered Operation, Movement and Target host.

Menu: **Ansible-Legacy > Target host**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

Item No.	Operation*	Movement*	Host*	Setting
Auto-input	1:GatherFacts1 ▾	1:Gatherfacts ▾	1:targethost ▾	Setting

Operation	Movement	Host
GatherFacts1	GatherFacts	targethost

1.7 Create Parameter sheet for registering collected values (1/4)

Create Parameter sheet that registers collected values.

Create a Menu called "Gathered Facts". This will be a parameter sheet where the collected values will be automatically registered to.

Menu: **Create Menu > Create/Define menu**

- ① Use the table on the next page and fill out the following fields/items.
- ② Press the "Create" button.

Fill out the following for each item

- Item name
- Input method
- Maximum number of bytes

1. Basic info

2. Target Menu group

3. Items

No	Host name	Operation name	Reference date and time	Scheduled date	Last run date	ansible_architecture	ansible_bios_version	address	interface	network	ansible_distribut
1		Operation	2020/01/01 00:00	2020/01/01 00:00		String	String	String	String	String	String
2		Operation	2020/01/01 00:00	2020/01/01 00:00		String	String	String	String	String	String
3		Operation	2020/01/01 00:00	2020/01/01 00:00		String	String	String	String	String	String

Create

1.7 Create Parameter sheet for registering collected values (2/4)

1. Basic information

Menu name (Free field)	Creation target	Display order
Gathered Facts	Parameter Sheet (Host/Operation)	1

2. Target Menu group

Input	Substitution value	Reference
Input (Default)	Substitution value (Default)	Reference (Default)

3. Item

Item name (Free field)	Input method	Maximum number of bytes (Free value)
ansible_architecture	String	128
ansible_bios_version	String	128
ansible_default_ipv4 > address (✖)	String	128
ansible_default_ipv4 > interface (✖)	String	128
ansible_default_ipv4 > network (✖)	String	128
ansible_distribution	String	128
ansible_distribution_file_path	String	128
ansible_distribution_file_variety	String	128
ansible_distribution_major_version	String	128
ansible_distribution_release	String	128

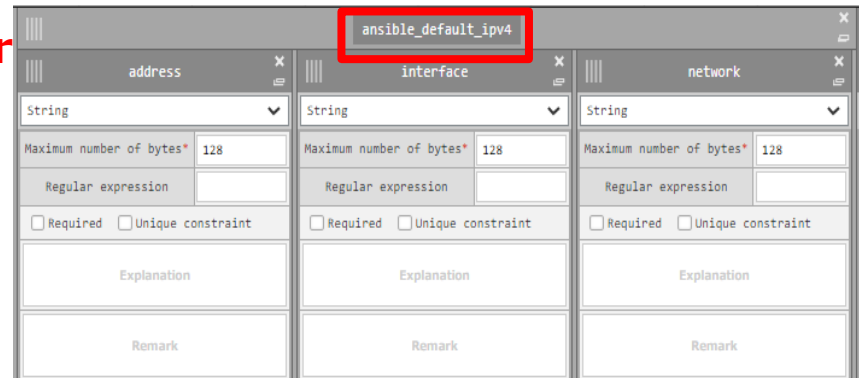
1.7 Create Parameter sheet for registering collected values (3/4)

Item name (Free field)	Input method	Maximum number of bytes (Free value)
ansible_distribution_version	String	128
ansible_machine	String	128
ansible_memtotal_mb	String	128
ansible_nodename	String	128
ansible_os_family	String	128
ansible_pkg_mgr	String	128
ansible_processor_cores	String	128

※ Change the following item names and group them together

ansible_default_ipv4 > address
ansible_default_ipv4 > interface
ansible_default_ipv4 > network

Create a column group called "ansible_default_ipv4 " and put the following columns in it. [address] , [interface] and [network]



1.7 Create Parameter sheet for registering collected values (4/4)



Created menu

The menu [Gathered Facts] has been created

You can check all the different items by pressing the "Register" button.

The screenshot shows the Exastro dashboard interface. The top navigation bar includes the Exastro logo and the word 'Input'. Below this is a 'Menu' button and a 'DASHBOARD' label. The main menu is displayed on the left, with 'Gathered Facts' highlighted in a red box. The main content area shows a 'Menu group' with three icons: 'Management Con...', 'Basic Console', and 'Export/Import'. Below these are three more icons: 'Input', 'Substitution value', and 'Reference', all of which are also highlighted with red boxes.

The 'Register' dialog box is shown in three instances, each with a red box highlighting the parameter list. The first instance shows parameters: ansible_architecture, ansible_bios_version, ansible_default_ipv4, address, interface, and network. The second instance shows: ansible_distribution, ansible_distribution_file_path, ansible_distribution_file_variety, and ansible_distribution_major_version. The third instance shows: ansible_machine, ansible_memtotal_mb, ansible_nodename, ansible_os_family, ansible_pkg_mgr, and ansible_processor_cores. A note at the bottom of the first instance states '* is a required item.'

1.8 Register Collected item value list (1/3)

Register Collected item value list

- Link the collect item's (FROM) YAML file name, variable name and the Parameter sheet's (TO) menu name and Item name.
- Use the table on the next page and register each variable and item as a single set.

Menu: **Ansible common > Collected item value list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

Collected item (FROM)

Parameter sheet (TO)

Collected items(FROM)				Parameter sheet(TO)		
ID	Perth format	PREFIX (file name)*	Variable name*	Member variables	Menu group:Menu	Item*
1	YAML	gatherfacts	ansible_architectu		2100011611:Substitution value:2:Gathered Facts	Parameter/ansible_architecture

1.8 Register Collected item value list (2/3)

Collected Item (FROM)			Parameter sheet (TO)	
Perth format	PREFIX (File name)	Variable name	Menu group :Menu	Item
YAML	gatherfacts	ansible_architecture	Substitution value: Gathered Facts	Parameter/ ansible_architecture
YAML	gatherfacts	ansible_bios_version	Substitution value: Gathered Facts	Parameter/ ansible_bios_version
YAML	gatherfacts	ansible_default_ipv4__address	Substitution value: Gathered Facts	Parameter/ ansible_default_ipv4/address
YAML	gatherfacts	ansible_default_ipv4__interface	Substitution value: Gathered Facts	Parameter/ ansible_default_ipv4/interface
YAML	gatherfacts	ansible_default_ipv4__network	Substitution value: Gathered Facts	Parameter/ ansible_default_ipv4/network
YAML	gatherfacts	ansible_distribution	Substitution value: Gathered Facts	Parameter/ ansible_distribution
YAML	gatherfacts	ansible_distribution_file_path	Substitution value: Gathered Facts	Parameter/ ansible_distribution_file_path
YAML	gatherfacts	ansible_distribution_file_variety	Substitution value: Gathered Facts	Parameter/ ansible_distribution_file_variety
YAML	gatherfacts	ansible_distribution_major_version	Substitution value: Gathered Facts	Parameter/ ansible_distribution_major_version
YAML	gatherfacts	ansible_distribution_release	Substitution value: Gathered Facts	Parameter/ ansible_distribution_release

1.8 Register Collected item value list (3/3)

Collected item (FROM)			Parameter sheet (TO)	
Perth format	PREFIX (file name)	Variable name	Menu group: Menu	Item
YAML	gatherfacts	ansible_distribution_version	Substitution value: Gathered Facts	Parameter/ Ansible_distribution_version
YAML	gatherfacts	ansible_machine	Substitution value: Gathered Facts	Parameter/ ansible_machine
YAML	gatherfacts	ansible_memtotal_mb	Substitution value: Gathered Facts	Parameter/ ansible_memtotal_mb
YAML	gatherfacts	ansible_nodename	Substitution value: Gathered Facts	Parameter/ ansible_nodename
YAML	gatherfacts	ansible_os_family	Substitution value: Gathered Facts	Parameter/ ansible_os_family
YAML	gatherfacts	ansible_pkg_mgr	Substitution value: Gathered Facts	Parameter/ ansible_pkg_mgr
YAML	gatherfacts	ansible_processor_cores	Substitution value: Gathered Facts	Parameter/ ansible_processor_cores

1.9 Register Collect interface information

Register Collect interface information

As REST API access is required when registering the collected values to parameter sheets in ITA, we will need to register a REST user that has execution permission.

Menu: **Ansible common > Collection interface information**

- ① Press the "Filter" button.
- ② Only 1 line will be displayed in the "List", so press the "update" button, fill in the information below and press the "register" button.

History	Update	ID	hostname	IP	REST user	REST password	REST method	protocol	port	Access permission
History	Update	1	localhost	127.0.0.1	administrator	*****	IP	http	80	Role to allow access



ID	hostname	IP	REST user	REST password	REST method	protocol	port
1	localhost	127.0.0.1	administrator	*****	IP	http	80

REST user	REST password
User with execute permission	The password of the user

1.10 Run operation (1/2)

Run the operation

Select Movement and Operation and execute them.

Menu: **Ansible-Legacy > Execution**

- ① Select the Movement we registered from Movement[list]
- ② Select the Operation we registered from Operation[list]
- ③ Press the “Execute” button

The screenshot displays the 'Ansible-Legacy > Execution' interface. It features two main panels: 'Movement [List]' and 'Operation [List]'. Both panels have a red border around their respective data tables. Below these panels is a summary section with 'Dry run' and 'Execute' buttons, and a table summarizing the selected movement and operation.

Movement [List]

Select	Movement ID	Movement Name	Orchestrator	Delay timer	Host specific format	WinRM connection	Dedicated information for ansible	Header section	Optional parameter	Access permission	Role to allow access	Remarks	Last update date/time	Last updated by
<input checked="" type="radio"/>	1	Gatherfacts	Ansible Legacy		IP		- hosts: all remote_user: "{{ __loginuser__ }}" gather_facts: yes become: yes						2021/08/31 18:19:41	System Administrator

Filter result count: 1

Operation [List]

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Access permission	Role to allow access	Remarks	Last update date/time	Last updated by
<input checked="" type="radio"/>	1	1	Gatherfacts1	2021/04/22 17:09					2021/08/31 18:10:22	System Administrator

Filter result count: 1

Movement ID 1
Movement Name Gatherfacts

Dry run **Execute**

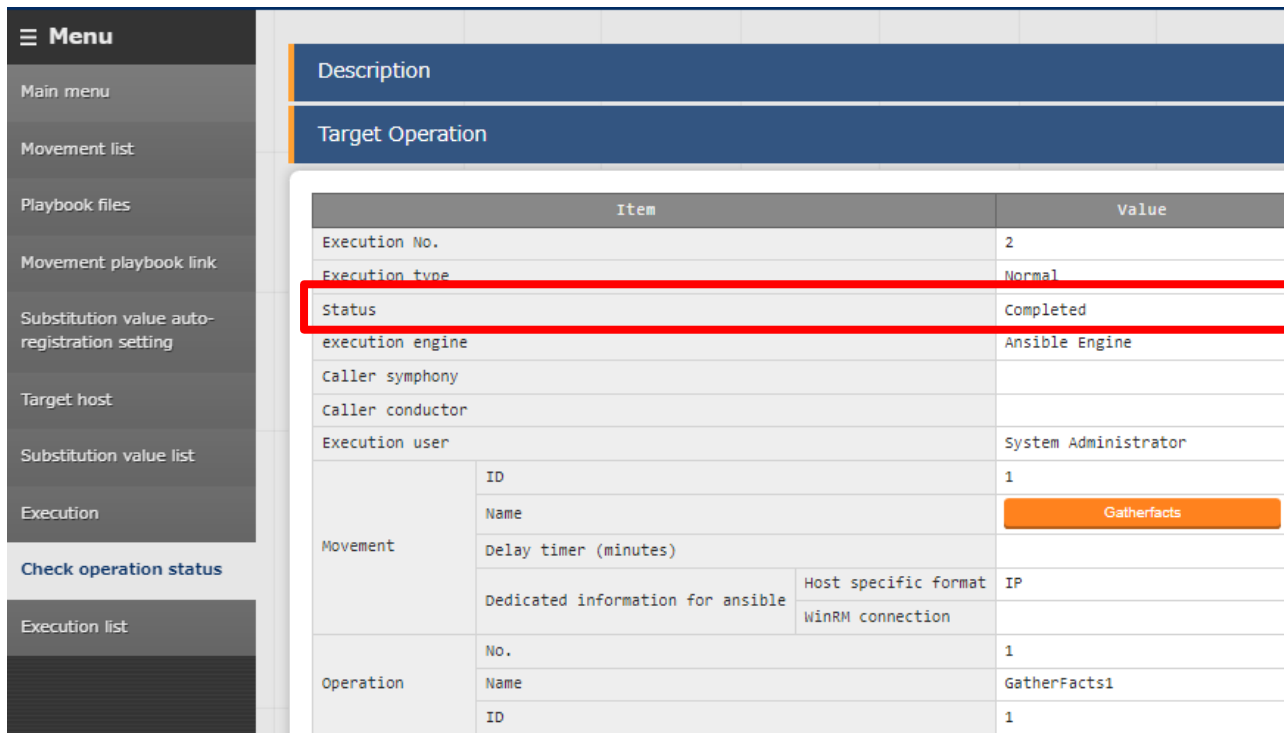
Movement[list]	Operation[list]
GatherFacts	GatherFacts1

1.10 Run operation (2/2)

Confirm the operation status

The operation ended successfully if the Status in the “Check operation status” menu says “Completed”

Menu: **Ansible-Legacy > Check operation status**



The screenshot shows the 'Check operation status' menu in the Ansible-Legacy interface. The menu is located on the left side of the screen, and the main content area displays a table of operation details. The 'Status' field is highlighted with a red box, indicating that the operation has been completed successfully.

Description		
Target Operation		
Item	Value	
Execution No.	2	
Execution type	Normal	
Status	Completed	
execution engine	Ansible Engine	
Caller symphony		
Caller conductor		
Execution user	System Administrator	
Movement	ID	1
	Name	Gatherfacts
	Delay timer (minutes)	
	Dedicated information for ansible	Host specific format IP
Operation	No.	1
	Name	GatherFacts1
	ID	1
	WinRM connection	

1.11 Confirm the collection results (1/2)

Confirm the collection results

Check if the collection succeeded/failed.

Menu: **Ansible-Legacy > Execution list**

① Press the "Filter" button.

② List > Collect status > "Status" can display the following:

- Collected : The data has been collected
- Collected (with notification) : Something went wrong when updating/registering
- Not target : Failed to collect
- Collection error : There is an error in the registered operation or the target host

History	Execution No. ⇅	Check execution status	Execution type ⇅	Status ⇅	execution engine ⇅	Collection status	
History	2	Check execution status	Normal	Completed	Ansible Engine	status ⇅	Collection log
						Collected	collectData_0000000002.log

1.11 Confirm the collection results (2/2)

Confirm the parameters

Check that the values has been registered to the parameter sheet.

Menu: **Input (or reference) > Gathered Facts**

- ① Press the "Filter" button.
- ② Check the list if all the items has values in them.

History	Duplicate	Update	Discard	No	Host name	Operation					Param	
History	Duplicate	Update	Discard		ID	Operation name	Reference date	Scheduled date for execution	Last execution date	ansible_architecture	ansible_bios_version	
				1	targethost	1 GatherFacts1	2021/09/01 13:28	2021/04/22 17:09	2021/09/01 13:28	x86_64	1.11.0-2.el7	

ansible_default_ipv4					ansible_distribution	ansible_distribution_file_path
ansible_default_ipv4 > address	ansible_default_ipv4 > interface	ansible_default_ipv4 > network				
192.0.2.1	eth0	192.0.2.0			CentOS	/etc/redhat-release

ansible_distribution_file_variety	ansible_distribution_major_version	ansible_distribution_release	ansible_distribution_version
RedHat	7	Core	7.8

ansible_machine	ansible_memtotal_mb	ansible_nodename	ansible_os_family	ansible_pkg_mgr	ansible_processor_cores
x86_64	1771	demo.localdomain	RedHat	yum	1

2. Scenario 2 【Compare function】

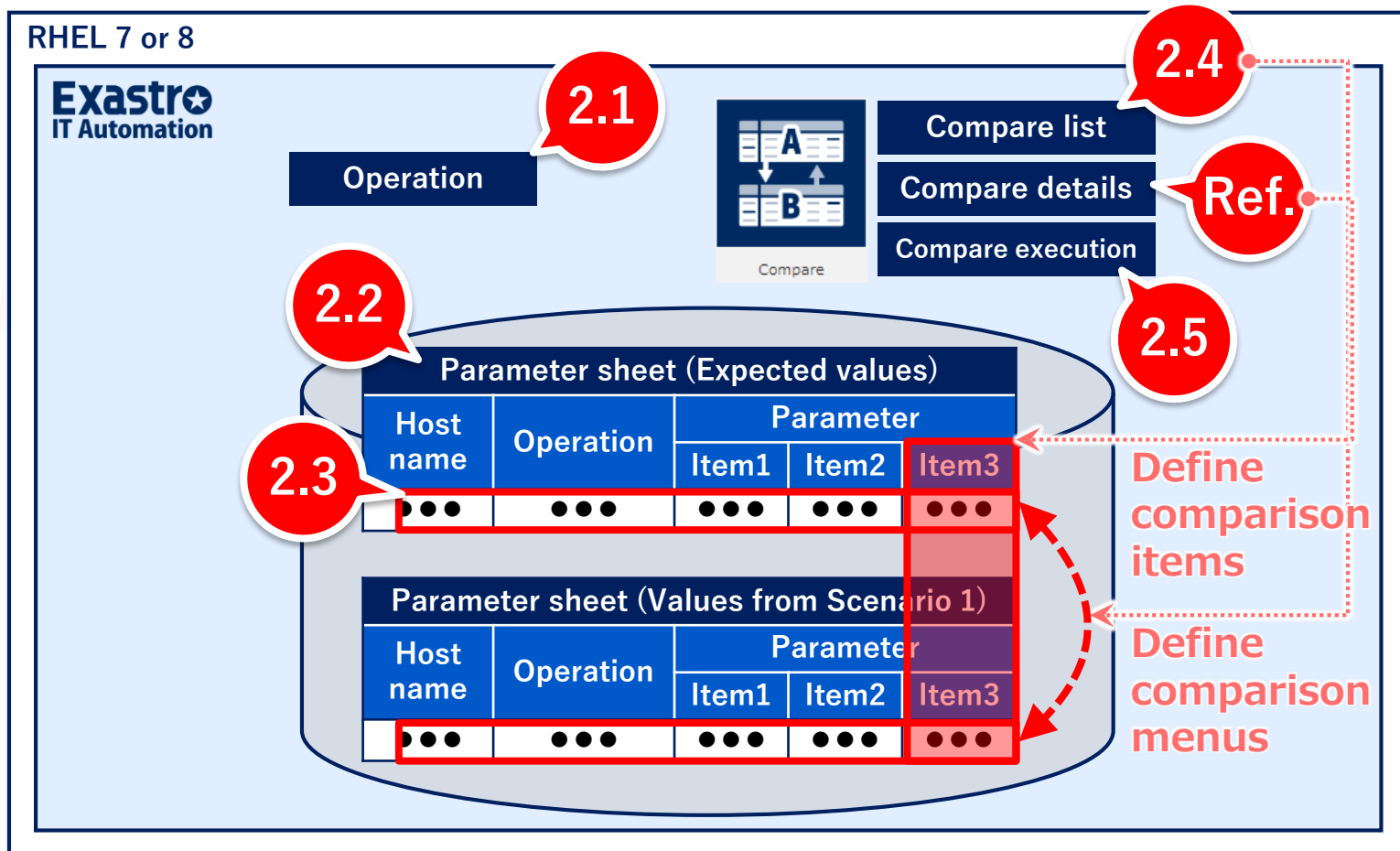
Compare the values and the expected values of the one collected in Scenario 1.

Scenario 2 Overall diagram

Scenario 2 workflow

- Register expected values to ITA and compare them to the values collected in Scenario 1.

Overall diagram



2.1 Register Operation

Register Operation

Register an operation that will compare the values.

Menu: **Basic Console > Operation list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

No.	Operation ID	Operation name*	Scheduled date for execution*	Access permission	
				Setting	Role to allow access
Auto-input	Auto-input	GatherFacts2	2021/10/01 09:25	Setting	

Operation name (Free space)	Scheduled date for execution (Free space)
GatherFacts2	2021/10/01 09:25

2.2 Create parameter sheet for expected values (1/3)

Create a parameter sheet for registering expected values.

Duplicate the parameter sheet we created in Scenario 1 and change the include order and menu name

Menu: **Create menu > Menu definition information**

- ① Press the “Filter” button and look for the “Gathered Facts” menu under “list”. After that, press the “Menu definition / creation” button.
- ② After the Menu definition screen appears, press the [Diversion new] button.
- ③ Only the “Menu name” and “Display” order will not be duplicated, so use table in the next slide to fill in the items.
- ④ Press the “Create” button.

History | 1 Gathered Facts | Menu definition - creation

ansible_machine	ansible_hosttotal_id	ansible_nodename	ansible_os_family
String	String	String	String
Regular number of bytes: 128	Regular number of bytes: 128	Regular number of bytes: 128	Regular number of bytes: 128
Regular expression	Regular expression	Regular expression	Regular expression
<input type="checkbox"/> Required <input type="checkbox"/> Unique constraint	<input type="checkbox"/> Required <input type="checkbox"/> Unique constraint	<input type="checkbox"/> Required <input type="checkbox"/> Unique constraint	<input type="checkbox"/> Required <input type="checkbox"/> Unique constraint
Explanation	Explanation	Explanation	Explanation
Remark	Remark	Remark	Remark

Menu creation information

Basic information

Menu name: Gathered Facts

Creation target: Parameter Sheet(Host/Operation)

Display order: 1

Create as hostgroup menu:

Create as vertical menu:

Last modified: 2021-08-31 19:10:31

Last updated by: System Administrator

Target menu group

Input: Input

Substitution value: Substitution value

Reference: Reference

Unique constraint(Multiple items)

Pattern:

Permission rule

Role:

Explanation

Remarks

List(Preview)

No.	Host name	Operation name	Reference site and time	Scheduled date	Last run date	ansible_architecture	ansible
1		Operation	2020/01/01 00:00	2020/01/01 00:00		String	String
2	192.168.0.1	Operation	2020/01/01 00:00	2020/01/01 00:00		String	String
3		Operation	2020/01/01 00:00	2020/01/01 00:00		String	String

Diversion new

2.2 Create parameter sheet for expected values (2/3)

The screenshot shows the 'Menu creation information' panel in Ansible Tower. The 'Menu name' is set to 'OS information' and the 'Display order' is set to '3'. A red circle with the number '3' highlights these fields. Below, a 'List(Preview)' table shows the menu item. A red circle with the number '4' highlights the 'Create' button.

No	Host name	Operation name	Reference date and time	Scheduled date	Last run date

Menu name (Free space)	Display order
OS information	3

The display order can be any number.

2.2 Create parameter sheet for expected values (3/3)



Created menu

No	Parameter					
	ansible_architecture	ansible_bios_version	ansible_default_ipv4	address	interface	network
Auto-input						

No	Parameter			
	ansible_distribution	ansible_distribution_file_path	ansible_distribution_file_variety	ansible_distribution_major_version

No	Parameter					
	ansible_machine	ansible_memtotal_mb	ansible_nodename	ansible_os_family	ansible_pkg_mgr	ansible_processor_cores

The "OS Information" menu has been created.

You can press "Start registration" to check if all the items are there.

2.3 Register expected values

Register expected values

We will now register the expected values to the “OS information” menu we created.

We want to make it so the values are different from the ones we collected in Scenario 1, so change the values in “ansible_default_ipv4__address” to something different.

Menu: **Input > OS information**

- ① Press “Register” -> “Start registration.”
- ② Input the following information and press the “Register” button.

No	Host name*	Operation	Parameter		
Auto-input	targethost	2021/10/01 09:25_2:GatherFacts2	ansible_architecture	ansible_bios_version	address
			x86_64	1.11.0-2.el7	193.0.2.2

The other items are shortened

Host name	Operation	Parameter/Item name	
		ansible_default_ipv4__address	Other items
targethost	GatherFacts2	Input a value different from the one collected in Scenario 1	Input the same values collected in scenario 1.

2.4 Register a Comparison

Select the two menus you want to compare

We will now define the comparison that will compare the values.

Menu: **Compare > Compare list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

No	Compare name *	Compare target menu 1 *	Compare target menu 2 *	Match all cases
Auto-input	OS info	2100011611:Substitution value:5:OS information	2100011611:Substitution value:2:Gathered Facts	<input checked="" type="radio"/>

Comparison definition name (Free field)	Compare target menu 1	Compare target menu 2	Match all cases
OS info	Substitution value:8:OS information	Substitution value:Gathered Facts	<input checked="" type="radio"/>

Here, we will compare all of the items, so select "●"

※ If you only want to compare select items, please see [\[reference\] Comparison details](#).

2.5 Run comparison (1/2)

Run the previously defined Comparison

We will now compare the values.

Menu: **Compare > Compare execution**

- ① Input/Select the following and press the “Compare” button.
- ② The comparison results will be displayed

Compare execution

Compare list: 1:OS info [5:OS information - 2:Gathered...] Base date 1: Base date 2: Target host: Choice

Output: ALL Difference Only

Compare

Comparison definition	Standard date 1	Standard date 2	Output
OS information-Gathered Facts	Blank	Blank	ALL

If you only want the comparison to output the items with differences, select “Difference Only”

2.5 Run comparison (2/2)



Comparison results

If the records contains an item with a difference, the "result" column will display "Difference".

Compare item number	Result	hostname	Menu name	No	Operation name	Base date	Parameter/ansible_architecture	Parameter/ansible_bios_version	Parameter/ansible_default_ipv4/address
1	Difference	targethost	OS information	1	GatherFacts2	2021/10/01 09:25	x86_64	1.11.0-2.el7	3.0.2.2
2	Difference	targethost	Gathered Facts	1	GatherFacts1	2021/09/01 13:28	x86_64	1.11.0-2.el7	2.168.141.12

Excel output

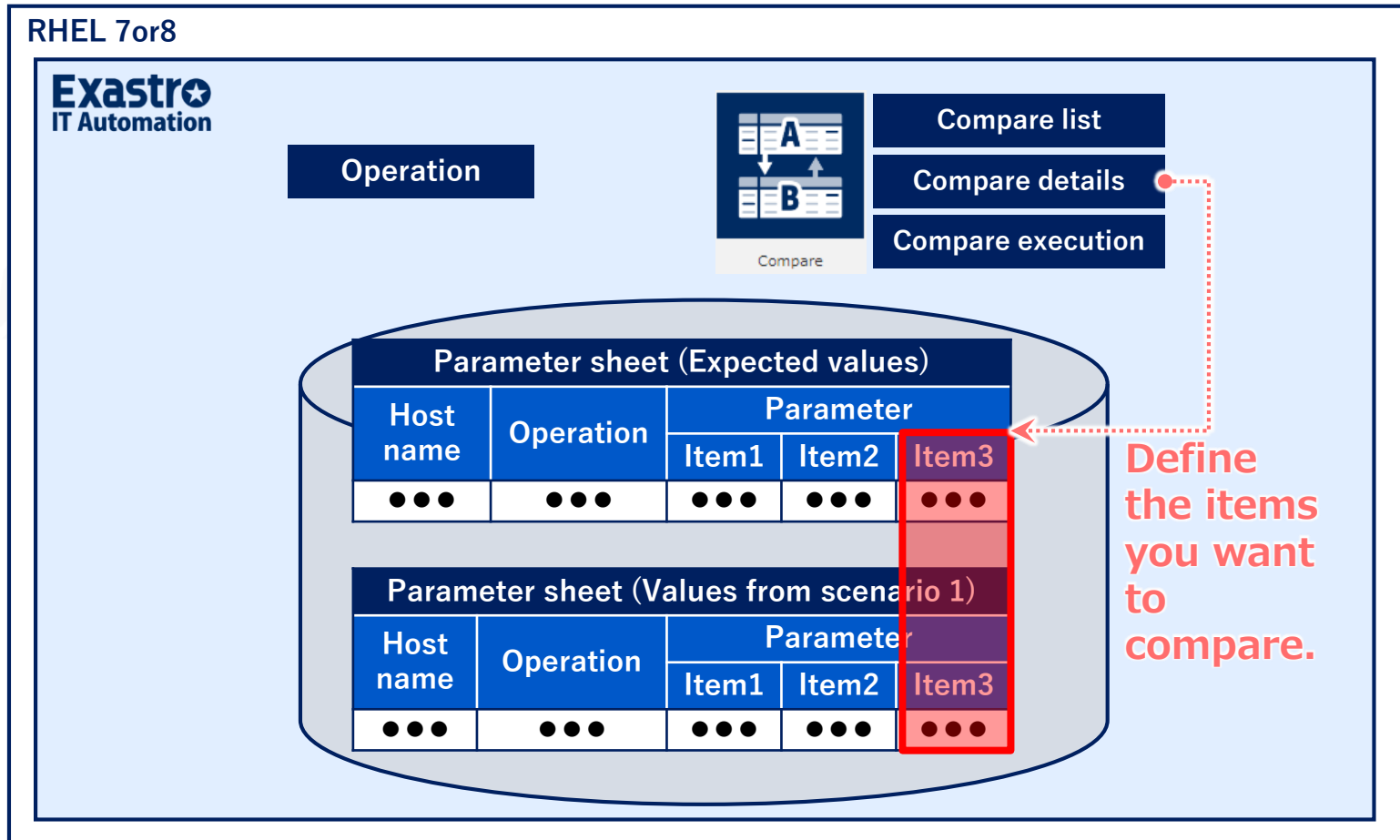
CSV output

The address, which has different values, will be displayed in red.

Compare single parameter sheet items.

- You can use the “Compare details” menu if you want to compare single items in a certain parameter sheet.

Diagram



【Reference】 (1) Register Comparison details

Select the 2 menus you want to compare

Select the menus you want to compare. Since we are only comparing select items, make sure that Match all cases is set to "OFF".

Menu: **Compare** > **Compare list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

No	Compare name*	Compare target menu 1*	Compare target menu 2*	Match all cases
Auto-input	IP address	2100011611:Substitution value:5:OS information ▼	2100011611:Substitution value:2:Gathered Facts ▼	▼

Compare name (Free)	Compare target menu 1	Compare target menu 2	Match all cases
IP address	Substitution value:8:OS information	Substitution value:Gathered Facts	-

Make sure that this item is blank.

[Reference] (2) Register Compare details

Select the items you want to compare

Select the items you want to compare from the menus in the Compare details menu.

Menu: **Compare > Compare details**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

The screenshot shows a form with the following fields:

- Compare name***: A dropdown menu with the selected value "ip address [5:OS information-2:Gathered Facts]".
- Display item name***: A text input field containing "IP address".
- Target column 1**: A dropdown menu with the selected value "2100011611:Substitution value:5:OS information:20:Parameter/ansible_default_ipv4/address".
- Target column 2**: A dropdown menu with the selected value "2100011611:Substitution value:2:Gathered Facts:3:Parameter/ansible_default_ipv4/address".
- Display order**: A text input field containing "1".

Comparison definition name	Display item name	Target column 1	Target column 2	Display order
IP address [Gathered Facts-OS information]	IP address	Substitution value: OS information:Parameter/ansible_default_ipv4/address	Substitution value: Gathered Facts:Parameter/ansible_default_ipv4/address	1

Run the comparison

Now that you've configured the Comparison definition details, we can now run the comparison.

Menu: **Compare > Run Comparison**

- ① Input/Select the following and press the "Compare" button.
- ② The comparison results will be displayed

Compare execution

Compare list: 2:ip address [5:OS information - 2:Gathe... Base date 1: Base date 2: Target host: Choice

Output: ALL Difference Only

Compare

Comparison definition	Base date 1	Base date 2	Output
IP Address [OS information-Gathered Facts]	Blank	blank	ALL

【Reference】 (3) Run Comparison (2/2)



Comparison results

Compare result

Compare item number	Result	Hostname	Menu name	No	Operation name	Base date	IP address
1	Difference	targethost	OS information	1	GatherFacts2	2021/10/01 09:25	193.0.2.2
2	Difference	targethost	Gathered Facts	1	GatherFacts1	2021/09/01 13:28	192.168.41.12

Excel output

CSV output

Only the specified item will be displayed.

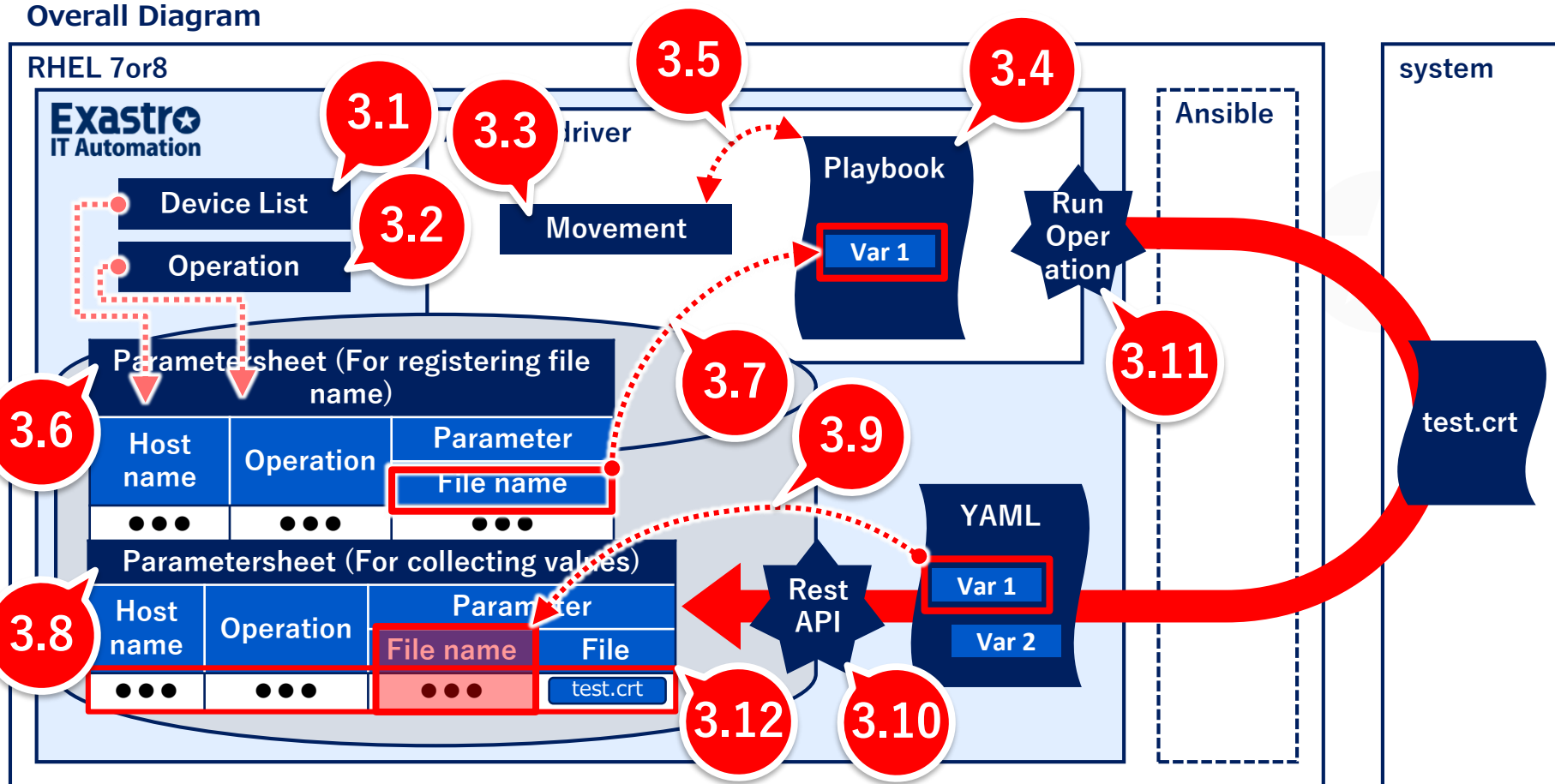
3. Scenario 3 【Collect function】 Collect the target host's SSL certificate file

Scenario 3 Overall diagram

Scenario 3 workflow

- While the contents are more or less the same as Scenario 1, in this scenario, we will collect a file.
- The file collected from the Parameter sheet will be downloadable.

Overall Diagram



3.1 Register Target host

Register target host connection information

- You can skip this step if you are using the same host you used in Scenario 1.

Menu: **Basic Console**> **Device List**

- Press "Register" -> "Start registration."
- Input the following information and press the "Register" button.

The screenshot shows a registration form with several sections. Red boxes highlight the following fields: 'HW device type' (SV), 'Host name' (targethost), 'IP address' (192.0.2.1), 'Login user ID' (root), 'Management' (radio button), 'Login password' (masked with asterisks), and 'Authentication method' (Password authentication). A blue dashed arrow points from the 'Login password' field to the 'Authentication method' dropdown.

HW device type	Host name (Free space)	IP address (Depends on your environment)	Login user ID (Depends on your environment)	Login password		Ansible dedicated information
				Management	Login password (Depends on your environment)	Dedicated information for Legacy/Role
						Authentication method
SV	targethost	192.0.2.1	root	●	*****	Password Authentication

3.2 Register operation

Register operation

Register the operation we will use in this scenario.

Menu: **Basic Console**> **Operation list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

No.	Operation ID	Operation name*	Scheduled date for execution*	Acc
3	3	<input type="text" value="getSSL1"/>	<input type="text" value="2021/04/23 17:10"/>	Setting <input type="button" value="Setting"/>

Operation name (Free space)	Scheduled date for execution (Free space)
getSSL1	2021/04/23 17:10

3.3 Register Movement

Register Movement in Ansible-Legacy

After this, we will link a Playbook to it, making it Movement that collects the SSL certificate.

Menu: **Ansible-Legacy > Movement List**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

Movement ID	Movement Name*	Delay timer	Dedicated information for ansible		
			Host specific format*	WinRM connection	
Auto-input	getSSL		IP		

Movement name (Free space)	Ansible User information
	Host specific format
getSSL	IP

3.4 Register Playbook (1/3)

Register Playbook for operations (1/2)

- The playbook that we will register in this guide contains a workflow which consists of "Generate YAML file which will gather the SSL certificate under the Target host's /tmp/ directory" → Copy the SSL certificate to the ITA host server's collection directory".
- For more information regarding the File collection directory, please see chapter [3.4.1 File collection directory](#)
- Make sure to create and put the SSL certificate file (test.crt) under the target host's /etc/pki/tls/certs/ path in advance.

3.4 Register Playbook (2/3)

Register Playbook for operations (2/2)

```
- name: make yaml file
blockinfile:
  create: yes
  mode: 0644
  insertbefore: EOF
  marker: ""
  dest: "/tmp/getSSL.yml"
  content: |
    SSL_file_name      : {{ VAR_ssl_name }}
    SSL_file           : {{ VAR_ssl_name }}

- name: copy the make yaml file to local
fetch:
  src: "/tmp/getSSL.yml"
  dest: "{{ __parameter_dir__ }}/{{ inventory_hostname }}"
  flat: yes

- name: get SSL file
fetch:
  src: "/etc/pki/tls/certs/{{ VAR_ssl_name }}"
  dest: "{{ __parameters_file_dir__ }}/{{ inventory_hostname }}"
  flat: yes
```

File name : getSSL.yml

3.4 Register Playbook (3/3)

Register Playbook in Ansible-Legacy

Register the Playbook we created in the last slide.

Menu: **Ansible-Legacy** > **Playbook file**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

Playbook ID	Playbook name*	Playbook files*	Access permission	
			Setting	Role to allow access
Auto-input	getSSL	<input type="button" value="Choose File"/> getSSL.yml <input type="button" value="Upload in advance"/> Upload status: Uploaded. File name getSSL.yml Size514bytes	<input type="button" value="Setting"/>	

Playbook file name (Free space)	Playbook file
getSSL	getSSL.yml

3.4.1 File collection directory (1/2)

The collected files will be stored in a file collection directory.

- The collected file will be stored in the File Collection directory specified by the ITA reserved variables.

```
dest: "{{ __parameters_file_dir__ }}"{{ inventory_hostname }}/"
```

getSSL.yml
2 lines from
the bottom

This Directory will be specified

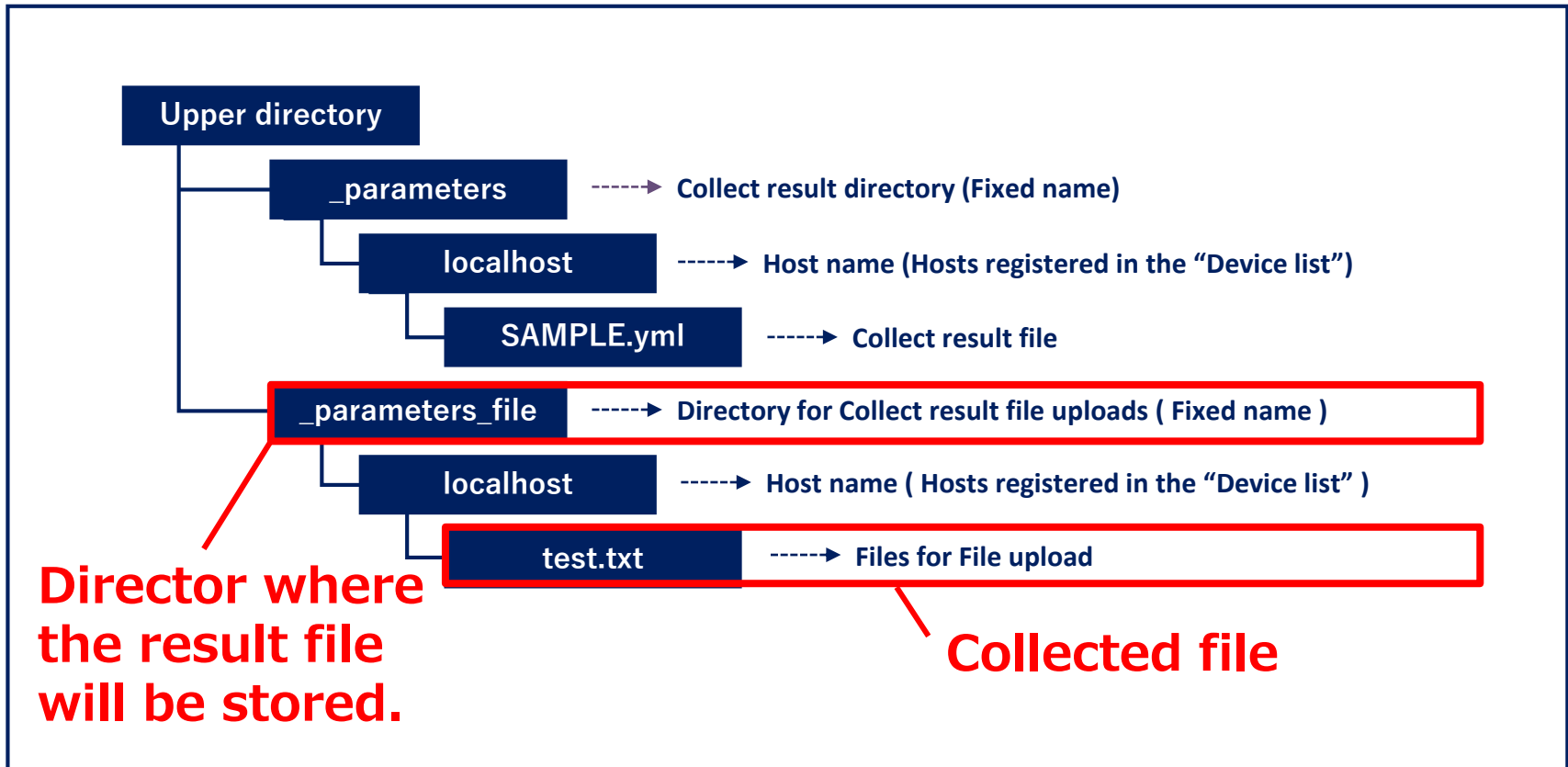
Path variables

	ITA reserved variable	Variable specified contents
Source file storage location	__parameter_dir__	"_parameters" path under the operation result directory
Collected file storage location	__parameters_file_dir__	"_parameters_file" path under the operation result directory

3.4.1 File collection directory (2/2)

The following figure displays the file hierarchy for the Collect file directory.

File hierarchy



3.5 Movement-Playbook link

Link Movement and Playbook

Link the previously registered Movement and Playbook.

Menu: **Ansible-Legacy > Movement-Playbook link**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

Associated item No.	Movement*	Playbook files*	Include order*
Auto-input	2:getSSL	getSSL	1

Movement	Playbook file	Include order
getSSL	getSSL	1

3.6 Register File name (1/3)

Create menu for variable registration

Create a Parameter sheet that we can use to register the File name (test.crt)

Menu: **Create Menu** > **Create/Define menu**

- ① Use the table below to fill out the following items.
- ② Press the "Create" button

1. Basic Information

Menu name (Free space)	Creation target	Display order
SSL certificate name	Parameter sheet (Host/Operation)	4

2. Target Menu group

Input	Substitution value	Reference
Input (Default)	Substitution value (Default)	Reference (Default)

3. Item

Item name (Free space)	Input method	Maximum number of bytes (Free value)
File name	String	128

3.6 Register File name (2/3)



Created menu

The menu [SSL certificate name] has been created

You can check all the different items by pressing the "Register" button.

EXASTRO IT Au

Menu

Main menu

- Gathered Facts
- OS Information
- SSL certificate name

Input

Substitution value

Reference

No	Host name*	Operation	Parameter
		Operation*	File name
Auto-input	<input type="text"/>	<input type="text"/>	<input type="text"/>

3.6 Register File name (3/3)

Register File name

Register the file name (test.crt) to the parameter sheet you created.

Menu: **Input > SSL certificate name**

- ① Press "Register" -> "Start registration.
- ② Input the following information and press the "Register" button.

No	Host name*	Operation	Parameter
		Operation*	File name
Auto-input	targethost ▼	2021/04/23 17:10_3:getSSL1 ▼	test.crt

Host name	Operation	Parameter
		File name
targethost	getSSL1	test.crt

3.7 Register substitution value auto-registration settings

Register Substitute value auto-registration settings

- Link the name of the file we will collect (Specific values) with the variables inside the Playbook.name ([File name : test.crt] [Variable name : VAR_ssl_name])
- By registering the file name to a different parameter sheet (already done in chapter [3.6 Register file name](#)) and linking the playbook variable to the parameter sheet's item name in the substitution value auto-registration setting menu, the system can automatically set the variable's specific values.

Menu: **Ansible-Legacy** > **Substitution value auto-registration setting**

- ① Press "Register" -> "Start registration.
- ② Input the following information and press the "Register" button.

Item No.	Menu group:Menu	Item	Registration method	Movement
Auto-input	2100011611:Substitution value:8:SSL certificate name	Parameter/File name	Value type	2:getSSL

Value variable
Variable name
1:VAR_ssl_name

Parameter sheet (From)		Registration method	IaC variable (To)	
Menu group : Menu	Item		Movement	Value variable
				Variable name
Substitution value : SSL certificate name	Parameter/File name	Value-type	getSSL	VAR_ssl_name

3.8 Create Parameter sheet for collect values (1/3)

Create a Parameter sheet that registers collected values.

- Create a menu called “SSL certificate”
- Inside the menu, create 2 items and name them “File name” and “File”. The “File” item will later allow us to download the collected file.

Menu: **Create menu > Create/Define Menu**

- ① Use the table below to fill out the following fields.
- ② Press the “Create” button.

1. Basic information

Menu name (Free space)	Creation target	Display order
SSL certificate	Parameter sheet (Host/Operation)	2

2. Target Menu group

Input	Substitution value	Reference
Input (Default)	Substitution value (Default)	Reference (Default)

3.8 Create Parameter sheet for collect values (2/3)

3.Items

Item name (Free space)	Input method	Maximum number of bytes (Free value)
File name	String	128
File	File upload	1000000

Fill out the following for the items

Item name — File name

Input method — String

Maximum number of bytes — 128

Item name — File

Input method — File upload

Maximum number of bytes — 1000000

3.8 Create Parameter sheet for collect values (3/3)



Created menu

The menu "SSL certificate" has been created.

You can check all the different items by pressing the "Register" button.

Input
DASHBOARD

Menu group

Management Con... Basic Console Export/Import

Input Substitution value Reference

Register

No	Host name	Operation	Parameter								
Auto-input	<input type="text"/>	<input type="text"/>	<table border="1"><thead><tr><th>File name</th><th>File</th></tr></thead><tbody><tr><td><input type="text"/></td><td><input type="button" value="Choose File"/> No file chosen</td></tr><tr><td></td><td><input type="button" value="Upload in advance"/></td></tr><tr><td></td><td>Upload status:</td></tr></tbody></table>	File name	File	<input type="text"/>	<input type="button" value="Choose File"/> No file chosen		<input type="button" value="Upload in advance"/>		Upload status:
File name	File										
<input type="text"/>	<input type="button" value="Choose File"/> No file chosen										
	<input type="button" value="Upload in advance"/>										
	Upload status:										

* is a required item.

3.9 Register Collected item value list

Register Collected item value list

- Configure the parameter sheet so that the collected items will automatically be registered to the parameter sheet.
- Link the collect item's (FROM) YAML file name, variable name and the Parameter sheet's (TO) menu name and Item name. Do this for both the "File name" and "File" items.

Menu: **Ansible common > Collected item value list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

Collected item (FROM)

Parameter sheet (TO)

Collected items(FROM)			Parameter sheet(TO)			
ID	Perth format	PREFIX (file name)	Variable name	Member variables	Menu group:Menu	Item
Auto-input	YAML	getSSL	SSL_file_name		2100011611:Substitution value:11:SSL certificate	Parameter/File name

Collected item (FROM)			Parameter sheet (TO)	
Parse format	PREFIX (File name)	Variable name	Menu group :Menu	Item
YAML	getSSL	SSL_file_name	Substitution value:SSL certificate	Parameter/File name
YAML	getSSL	SSL_file	Substitution value:SSL certificate	Parameter/File

3.10 Register Collected interface information

Register Collect interface information

- As REST API access is required when registering the collected values to parameter sheets in ITA, we will need to register a REST user that has execution permission.
- If you are going to use the same Rest user you created in Scenario 1, you can skip this step.

Menu: **Ansible common > Collect interface information**

- ① Press the “Filter” button
- ② Only 1 line will be displayed in the “list”, so press the “update” button, fill in the information below and press the “register” button

History	Update	ID	hostname	IP	REST user	REST password	REST method	protocol	port	Access permission
										Role to allow access
History	Update	1	localhost	127.0.0.1	administrator	*****	IP	http	80	



ID	hostname	IP	REST user	REST password	REST method	protocol	port
1	localhost	127.0.0.1	administrator	*****	IP		

REST user	REST password
User with execute permission	The password of the user

3.11 Run operation (1/2)

Run operation

Select Movement and Operation and execute them.

Menu: **Ansible-Legacy > Execution**

- ① Select the Movement we registered from Movement[list]
- ② Select the Operation we registered from Operation[list]
- ③ Press the “Execute” button

The screenshot shows the 'Ansible-Legacy > Execution' interface. It consists of several panels:

- Movement [List]**: A table with columns: Select, Movement ID, Movement Name, Orchestrator, Delay timer, Host specific format, WinRM connection, Header section, Optional parameter, Access permis, Role to allow a, Last update date/time, Last updated by. The row for '2 getSSL' is highlighted with a red box.
- Operation [Filter]**: A button to open the operation list.
- Operation [List]**: A table with columns: Select, No., Operation ID, Operation name, Scheduled date for execution, Last execution date, Access permission, Role to allow access, Remarks, Last update date/time, Last updated by. The row for '3 getSSL1' is highlighted with a red box.
- Summary**: Shows 'Movement ID 2' and 'Movement Name getSSL'.
- Buttons**: 'Dry run' and 'Execute' buttons. The 'Execute' button is highlighted with a red box.

Below the screenshot, two boxes illustrate the mapping between the Movement and Operation names:

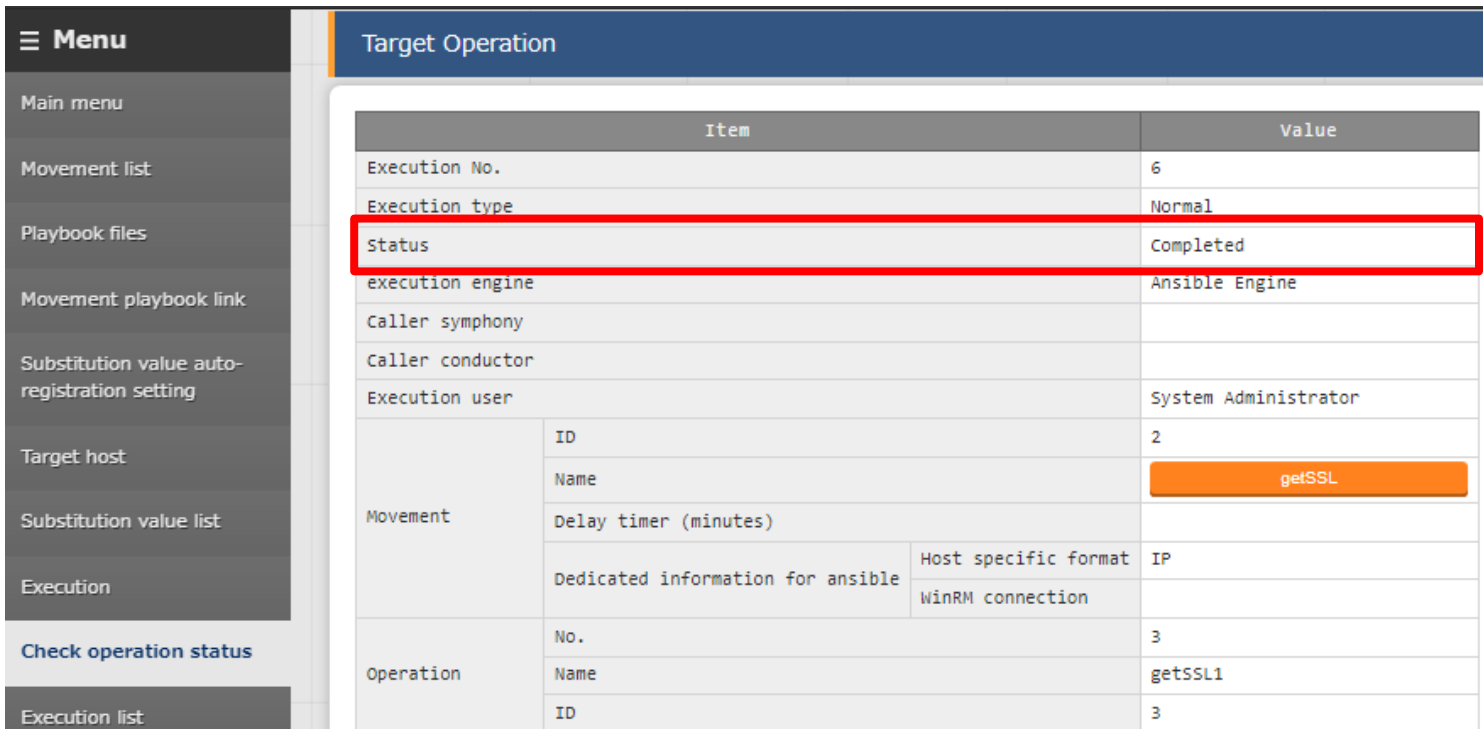
Movement [List]	Operation [List]
getSSL	getSSL1

3.11 Run operation (2/2)

Confirm the operation status

The operation ended successfully if the Status in the “Check operation status” menu says “Completed”

Menu: **Ansible-Legacy > Check operation status**



The screenshot shows the 'Check operation status' menu in the Ansible-Legacy interface. The left sidebar contains a 'Menu' with options like 'Main menu', 'Movement list', 'Playbook files', 'Movement playbook link', 'Substitution value auto-registration setting', 'Target host', 'Substitution value list', 'Execution', 'Check operation status', and 'Execution list'. The 'Check operation status' option is selected. The main content area displays a table with the following data:

Item		Value
Execution No.		6
Execution type		Normal
Status		Completed
execution engine		Ansible Engine
Caller symphony		
Caller conductor		
Execution user		System Administrator
Movement	ID	2
	Name	getSSL
	Delay timer (minutes)	
	Dedicated information for ansible	Host specific format IP winRM connection
Operation	No.	3
	Name	getSSL1
	ID	3



3.12 Confirm collection results (1/2)

Confirm the collection results

Check if the collection succeeded/failed.

Menu: **Ansible-Legacy > Execution list**

① Press the "Filter" button.

② List > Collect status > "Status" can display the following:

- Collected : The data has been collected
- Collected (with notification) : Something went wrong when updating/registering
- Not target : Failed to collect
- Collection error : There is an error in the registered operation or the target host

History	Execution No.	Check execution status	Execution type	Status
History	6	Check execution status	Normal	Completed

Collection status	
status	Collection log
collected	CollectData 000000006.log

3.12 Confirm collection results (2/2)

Confirm the parameters

Check that the values has been registered to the parameter sheet. You can also download the file.

Menu: **Input (or Substitution value) > SSL Certificate**

- ① Press the "Filter" button.
- ② Check the list if all the items has values in them.

					Operation					Parameter		
History	Duplicate	Update	Discard	No	Host name	ID	Operation name	Reference date	Scheduled date for execution	Last execution date	File name	File
History	Duplicate	Update	Discard		1 targethost	3	getSSL1	2021/09/02 09:38	2021/04/23 17:10	2021/09/02 09:38	test.crt	test.crt

4. Scenario 4 【Compare function】

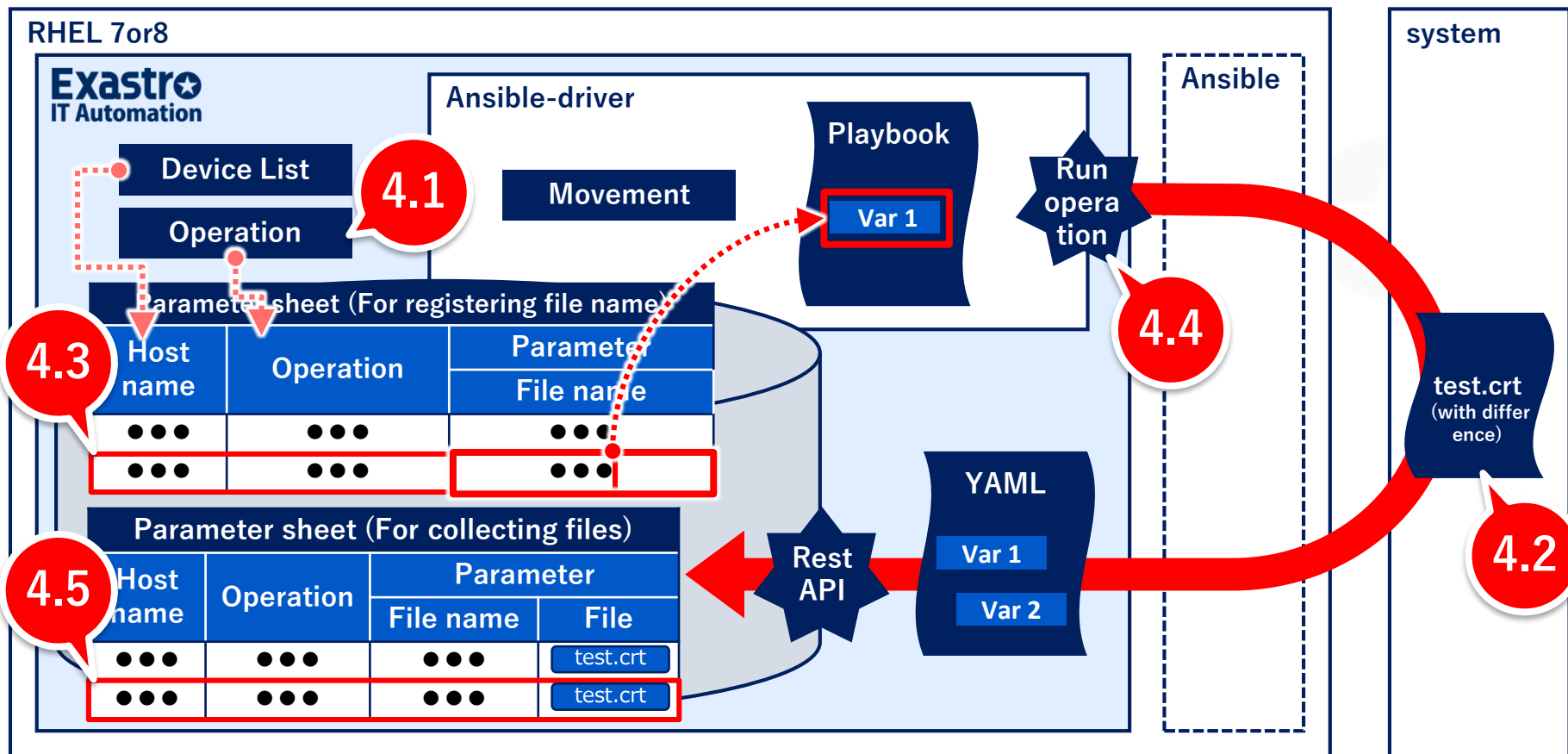
Compare the file downloaded in scenario 3 with the same file from a different date.

Scenario 4 Overall diagram (1/2)

Scenario 4 workflow

- Collect a SSL certification file with a different “base date” from the one we collected in Scenario 3 and compare the files.

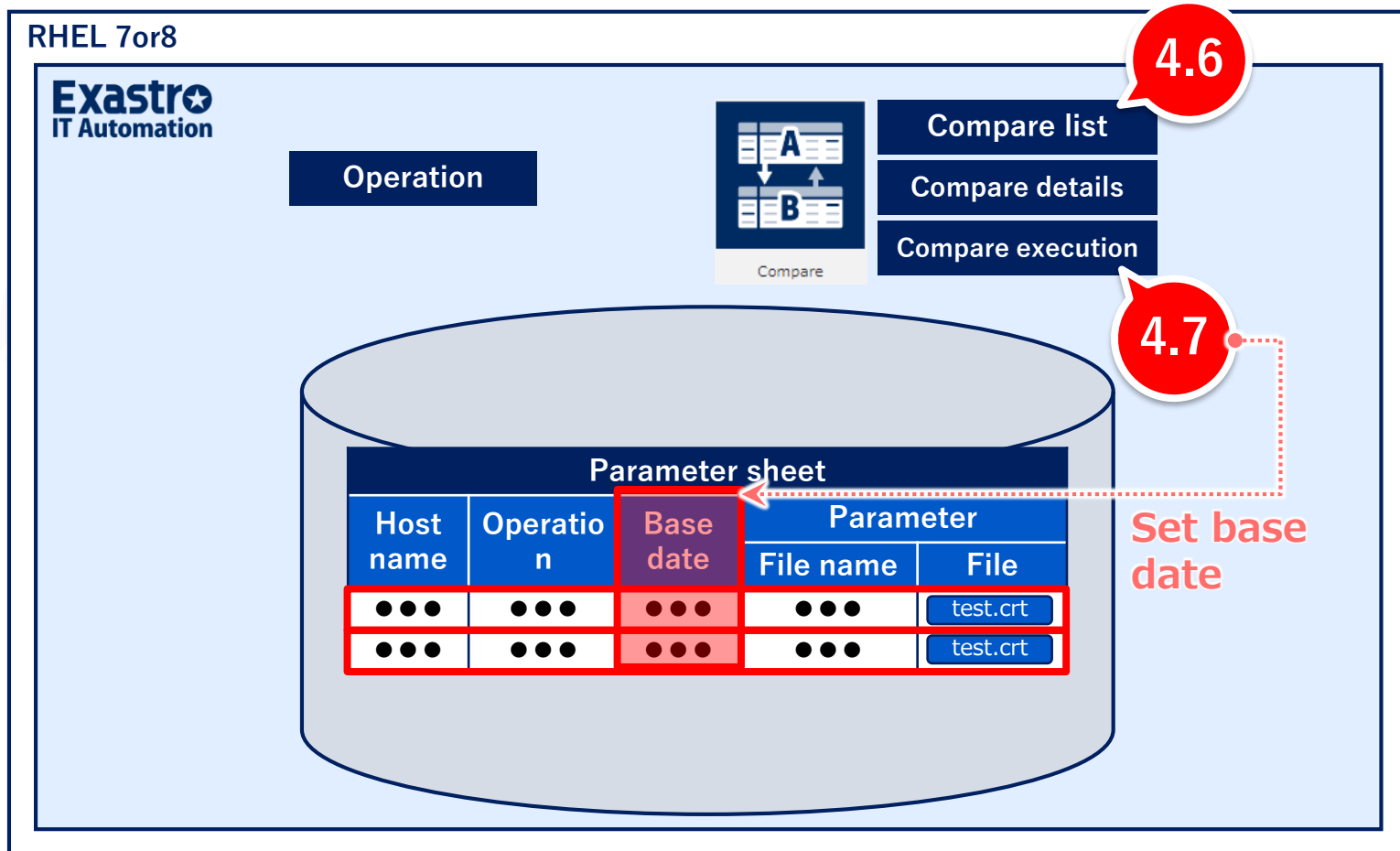
Overall diagram



Scenario 4 Overall diagram (2/2)

- As we are comparing a file within the same menu, but with different date values, we will change the "Standard date".

Diagram (Compare function)



4.1 Register operation

Register operation

Register an operation for comparing

Menu: **Basic Console**> **Operation list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

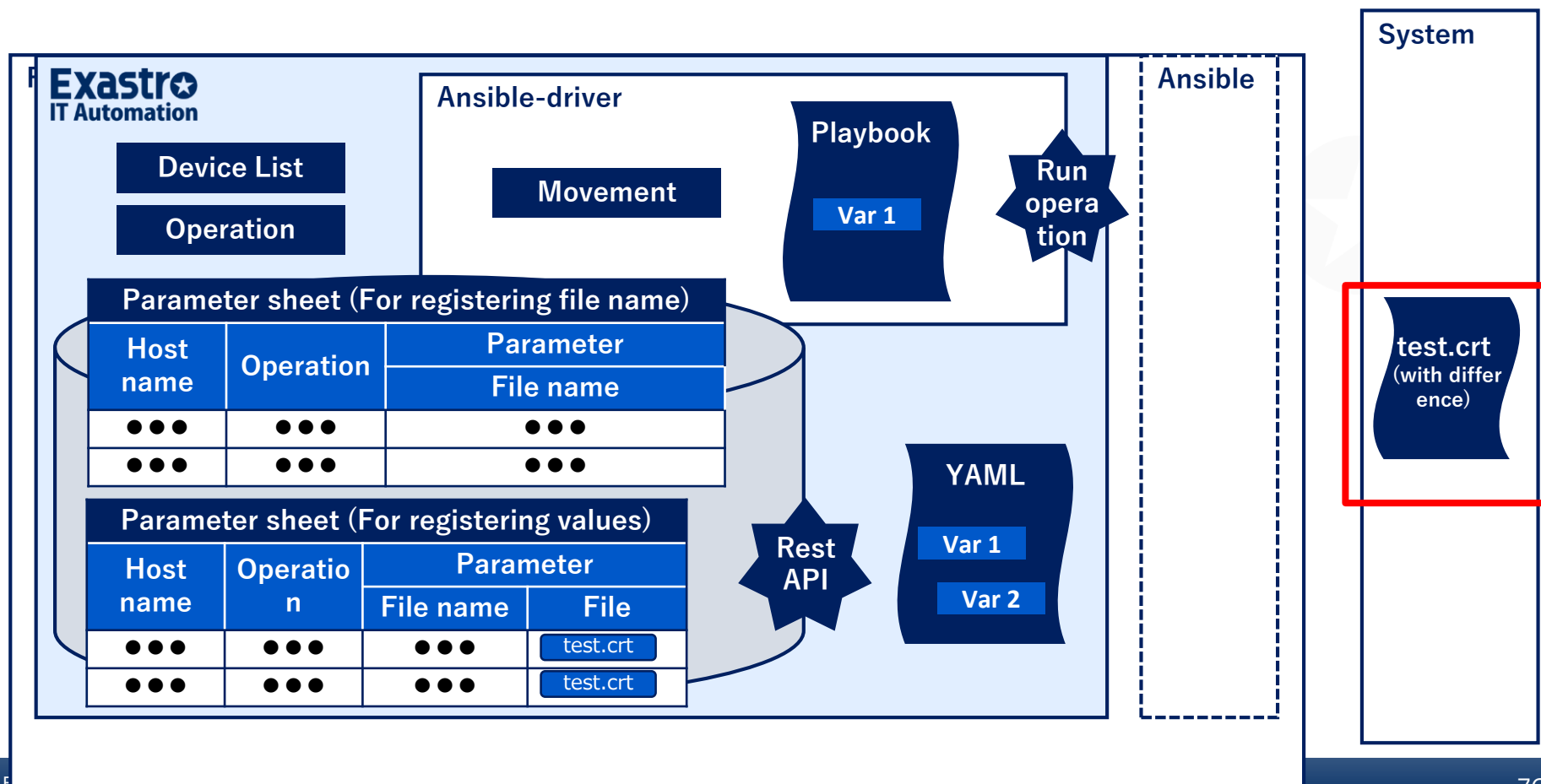
No.	Operation ID	Operation name*	Scheduled date for execution*	Access permission	
				Setting	Role to allow access
Auto-input	Auto-input	getSSL2	2021/04/28 12:19	Setting	

Operation name (Free space)	Scheduled date for execution (Free space)
getSSL2	2021/04/28 12:19

4.2 Prepare SSL certificate with different contents

Prepare an SSL certificate with a difference

- In this scenario, we want to check if something is different from the certificate we collected in Scenario 3, so prepare an SSL certificate different to that one.
- In order to do so, we will change a part of the contents of the SSL certificate (test.crt) that is in the Target server's /etc/pki/tls/certs/ directory.test.crt)



4.3 Register file name

Register file name

- Register a new Record in the menu we created in Scenario 3, "SSL certificate name".
- The contents should be the same as the one we created in Scenario 3, but change the operation to the one we created for comparing.

Menu: **Input > SSL certificate name**

- ① Press "Register" -> "Start registration.
- ② Input the following information and press the "Register" button.

No	Host name*	Operation	Parameter
		Operation*	File name
Auto-input	targethost	2021/04/28 12:19_4:getSSL2	test.crt

Host name	Operation	Parameter
		File name
targethost	getSSL2	test.crt

4.4 Run operation

Run operation

- Collect the SSL certificate with different contents from the one collected in scenario 3.
- Select the same Movement from Scenario 3, "getSSL". The operation should be the new one we created for comparing, "getSSL2".

Menu: **Ansible-Legacy > Execution**

- ① Select the registered Movement from Movement[List]
- ② Select the newly registered operation from Operation[List]
- ③ Press the "Execute" button.

The screenshot shows the 'Ansible-Legacy Execution' interface. It consists of several panels:

- Movement [List]**: A table with columns: Select, Movement ID, Movement Name, Orchestrator, Delay timer, Host specific format, MINM connection, Header section, Optional parameters, Access permission, Remarks, Last update date/time, Last updated by. Row 2 (ID: 2, Name: getSSL) is highlighted with a red box.
- Operation [List]**: A table with columns: Select, No., Operation ID, Operation name, Scheduled date for execution, Last execution date, Access permission, Remarks, Last update date/time, Last updated by. Row 4 (ID: 4, Name: getSSL2) is highlighted with a red box.
- Buttons**: At the bottom left, there are 'Dry run' and 'Execute' buttons. The 'Execute' button is highlighted with a red box.

Movement[List]	Operation[List]
getSSL	getSSL2

4.5 Confirm comparison results

Confirm Parameter sheet

- Check that the “getSSL2” has been collected to the Parameter sheet.
- [Check the Standard date/time \(We will need it in 4.7 Run Comparison\)](#)

Menu: **Input (or Reference) > SSL certificate**

- ① Press the “Filter” button.
- ② Check the updated list that the items has values in them.

List/Update

				No		Host name		Operation		Parameter	
History	Duplicate	Update	Discard	ID	Operation name	Reference date	Scheduled date for execution	File name	File		
History	Duplicate	Update	Discard	1	targethost	3 getSSL1	021/09/02 09:38	2021/04/23 17:10	test.crt	test.crt	
History	Duplicate	Update	Discard	2	targethost	4 getSSL2	021/09/02 13:13	2021/04/28 12:19	test.crt	test.crt	

4.6 Register Comparison definition

Select the 2 menu you want to compare

As we will compare the same menu, but with different values, choose the same menu for both of the items.

Menu: **Compare** > **Compare list**

- ① Press "Register" -> "Start registration."
- ② Input the following information and press the "Register" button.

No	Compare name *	Compare target menu 1 *	Compare target menu 2 *	Match all cases
Auto-input	SSL certificate	2100011611:Substitution value:11:SSL certificate	2100011611:Substitution value:11:SSL certificate	<input checked="" type="checkbox"/>

Comparison definition name (Free space)	Compare target menu 1	Compare target menu 2	Match all cases
SSL certificate	Substitution value:SSL certificate	Substitution value:SSL certificate	<input checked="" type="checkbox"/>

4.7 Run Comparison (1/3)

Run the comparison

- Select the “SSL Certificate” Comparison definition and input the standard dates.
- The Standard dates should be the most recent for both of them
- Please see the next page for more information regarding standard dates.

Menu: **Compare > Compare execution**

- ① Input the following information and press the “Compare” button.
- ② The Comparison result should be displayed

Compare execution

Compare list: 3:SSL certificate [11:SSL certificate - 11:...] Base date 1: 2021/09/02 12:00 Base date 2: 2021/09/02 13:30 target host: Choice

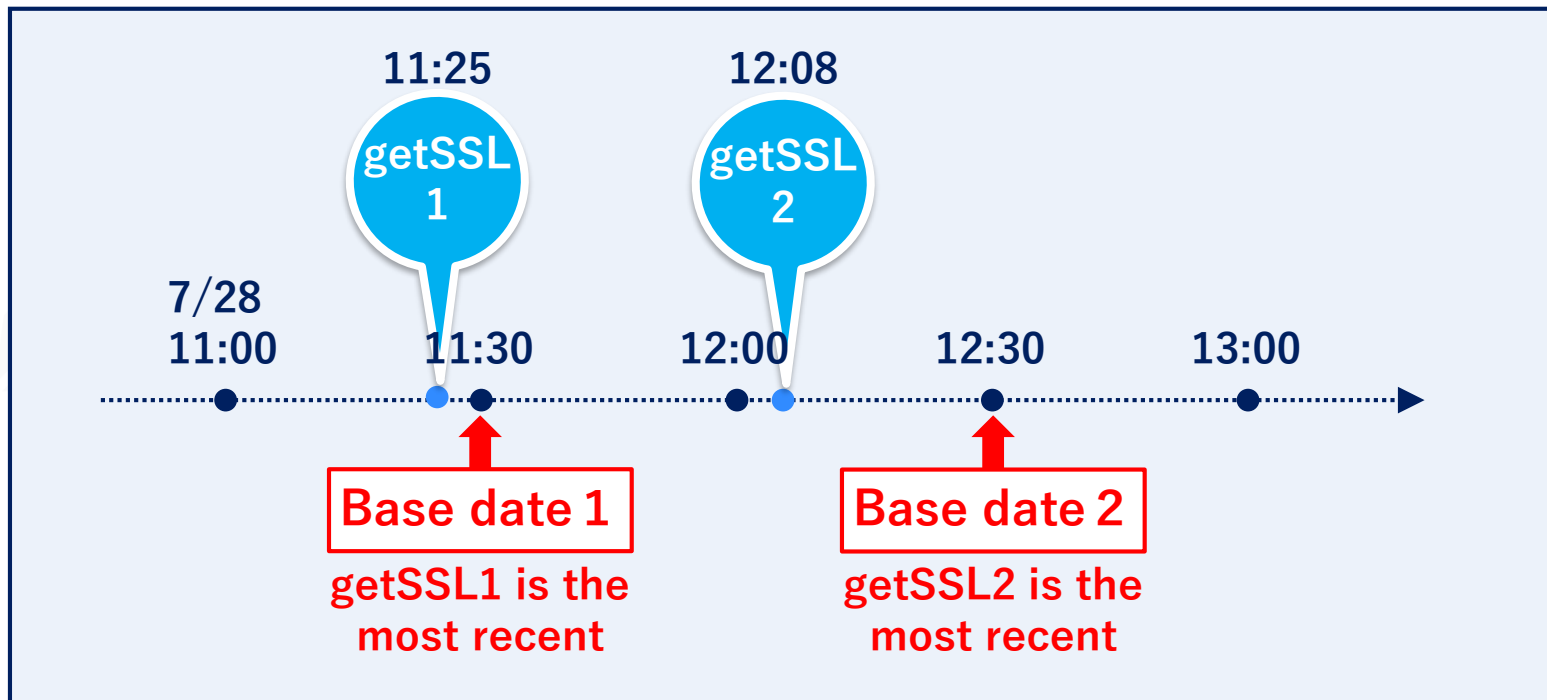
Output: ALL Difference Only

Compare

Comparison definition	Base date 1	Base date 2	Output contents
SSL certificate	2021/7/28 11:30	2021/7/28 12:30	ALL

4.7 Run comparison (2/3)

- The Standard dates are displayed below.



Set the base date depending on when the files were collected.

4.7 Run comparison (3/3)



Compare results

Compare result

Compare item number	Result	Hostname	Menu name	No	Operation name	Base date	Parameter/File name	Parameter/File
1	Difference	targethost	SSL certificate 1	1	getSSL1	2021/09/02 09:38	test.crt	test.crt
2	Difference	targethost	SSL certificate 2	2	getSSL2	2021/09/02 13:13	test.crt	test.crt

Excel output

CSV output

Items that are different (different content) are displayed in red text.



Exastro