IT Automation

Collect/Compare function【Practice】

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

Exastro IT Automation Version 1.8.0

Exastro developer

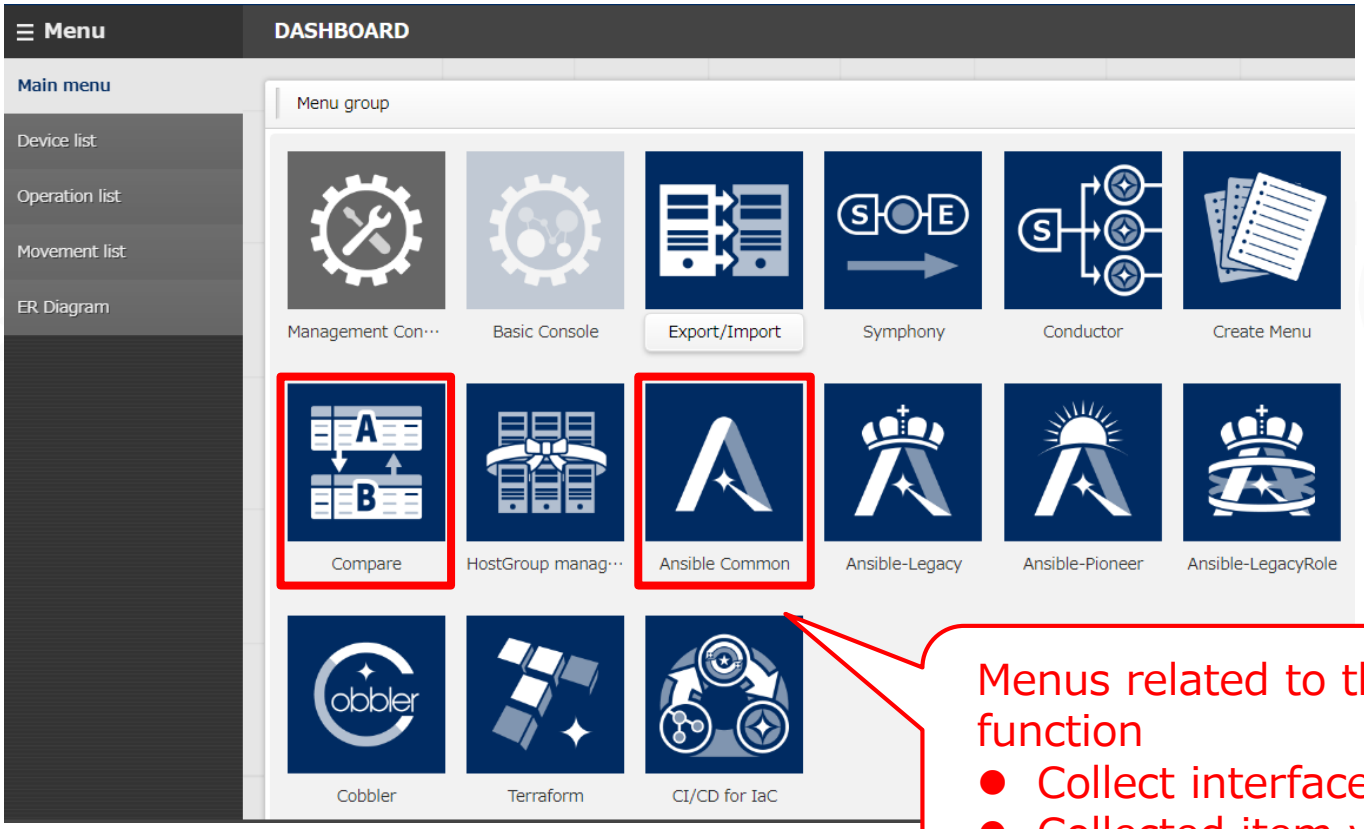
Table of contents

Table of contents

Introduction

About this document

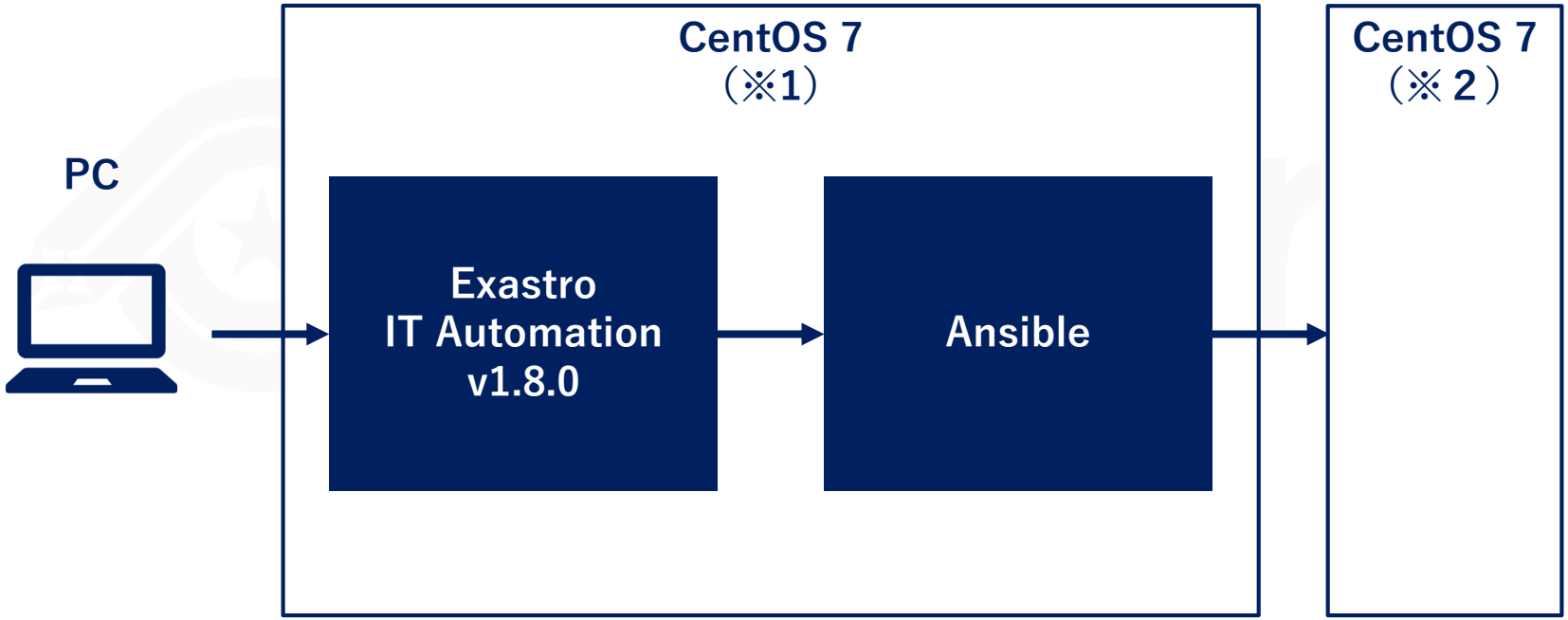
This document aims to teach the user about the Compare and Collect function by leading them through a hands-on scenario.



(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

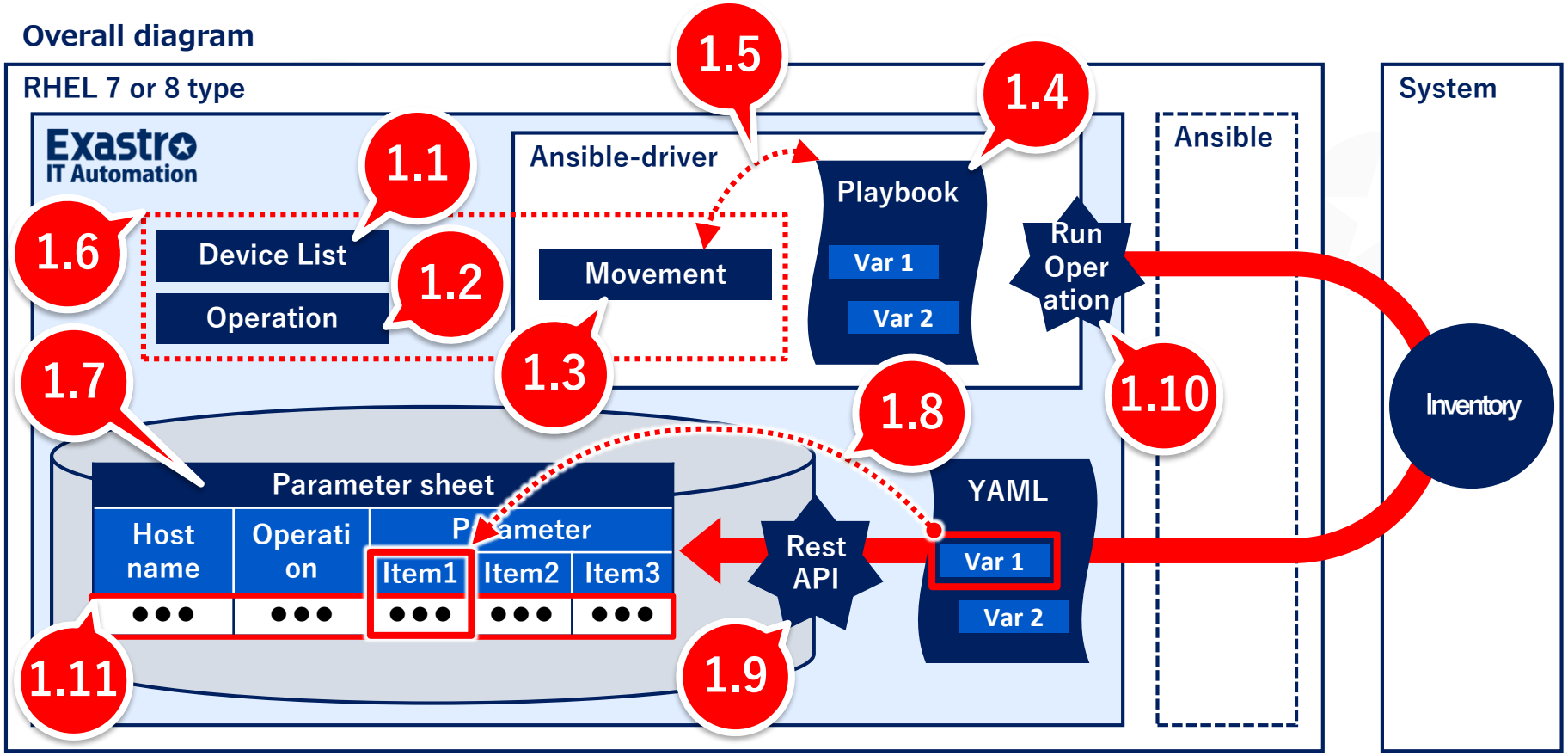
	Collect function	Compare function
Collecting/ Comparing parameters	**Scenario 1** Collect the target host OS information	**Scenario 2** Compare the values and the expected values of the one collected in Scenario 1.
Collecting/ Comparing files	**Scenario 3** Collect the target host's SSL certificate file	**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.



1.1 Register target host

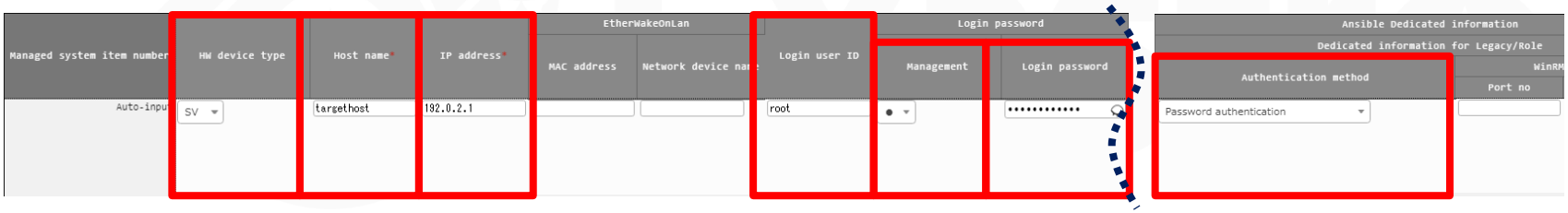
Register the target host connection information

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.



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

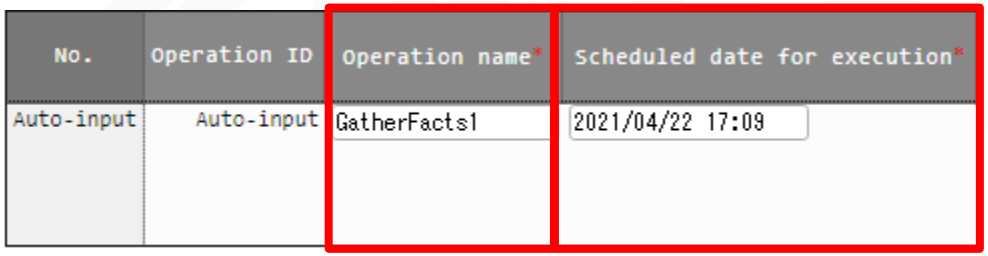
Register the operation we will use in Scenario 1.

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.



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.

Register Movement in Ansible-Legacy

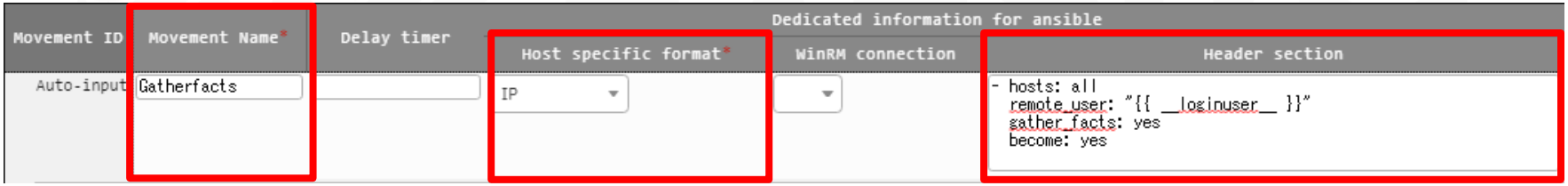
In ITA, the smallest automatic operation unit(A.K.A a “job”) is called a “Movement”.

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

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（1/2）

Create Playbooks

- We will create a Playbook that contains a YAML file that lists the gathered OS information.
- For more information regarding the YAML file and the directory where the files will be collected, please refer to chapter “1.4.1 Directory for YAML files and collection”

File name : GatherFacts.yml

```
- name: make yaml file
  blockinfile:
    create: yes
    mode: 644
    insertbefore: EOF
    marker: ""
    dest: "{{ __parameter_dir__ }}/{{ inventory_hostname }}/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 }}
  delegate_to: 127.0.0.1
```

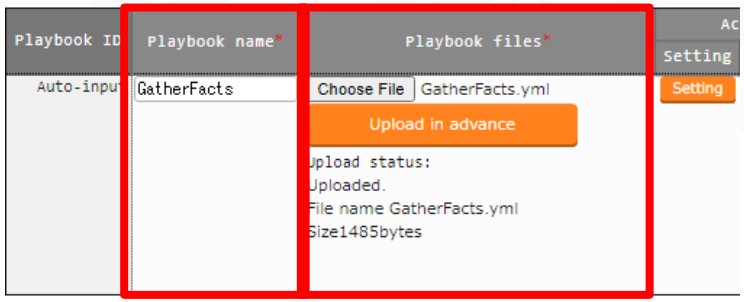
Register Playbook in Ansible-Legacy

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.



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 }}/gatherfacts.yml"
```

This directory will be specified

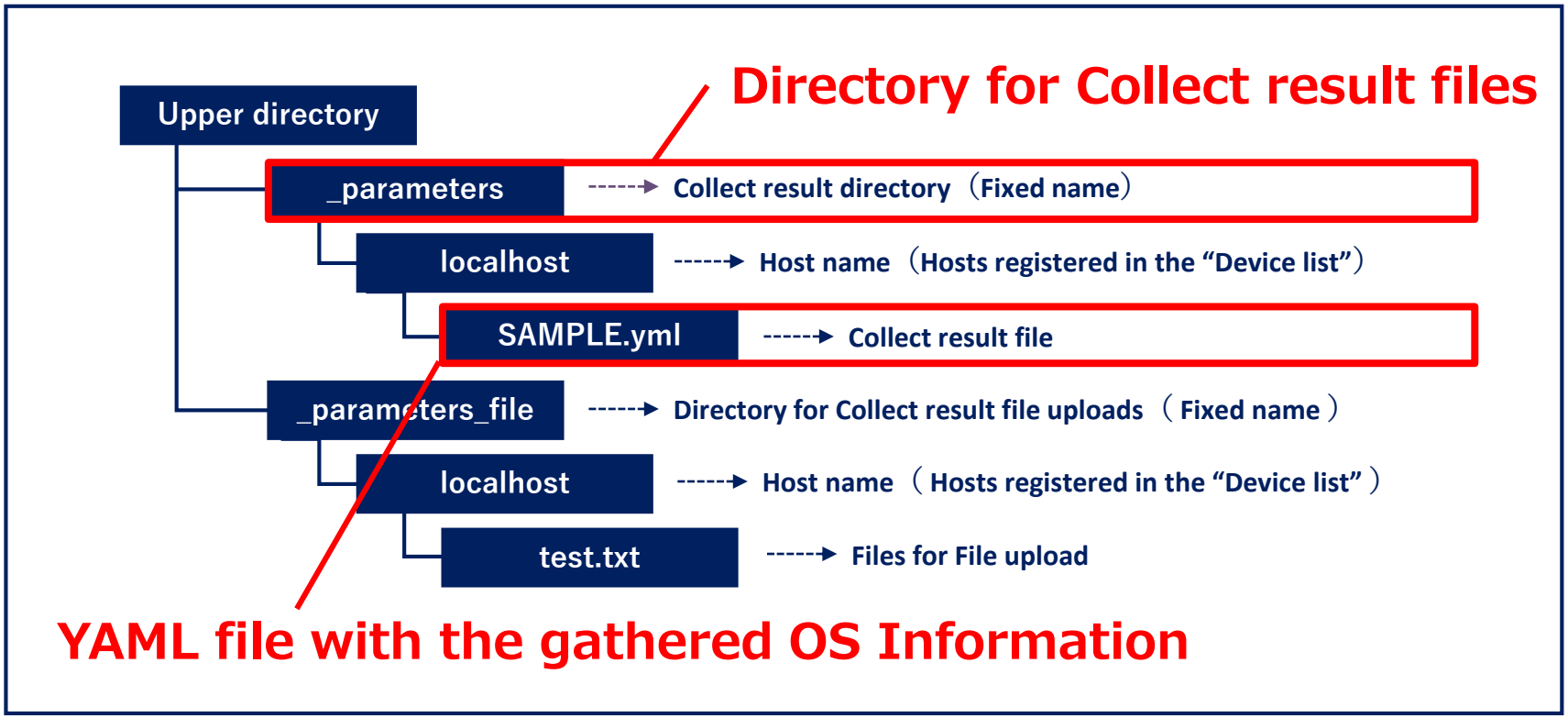
Path variables

	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

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



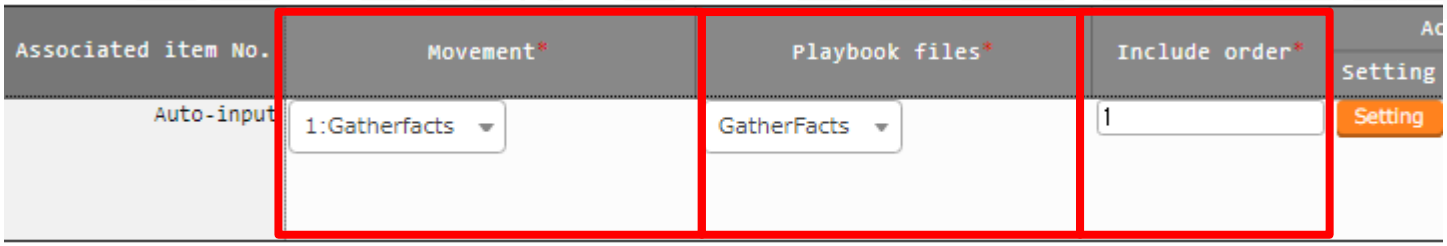
Link Movement and Playbook

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.



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.

Operation	Movement	Host
GatherFacts1	GatherFacts	targethost

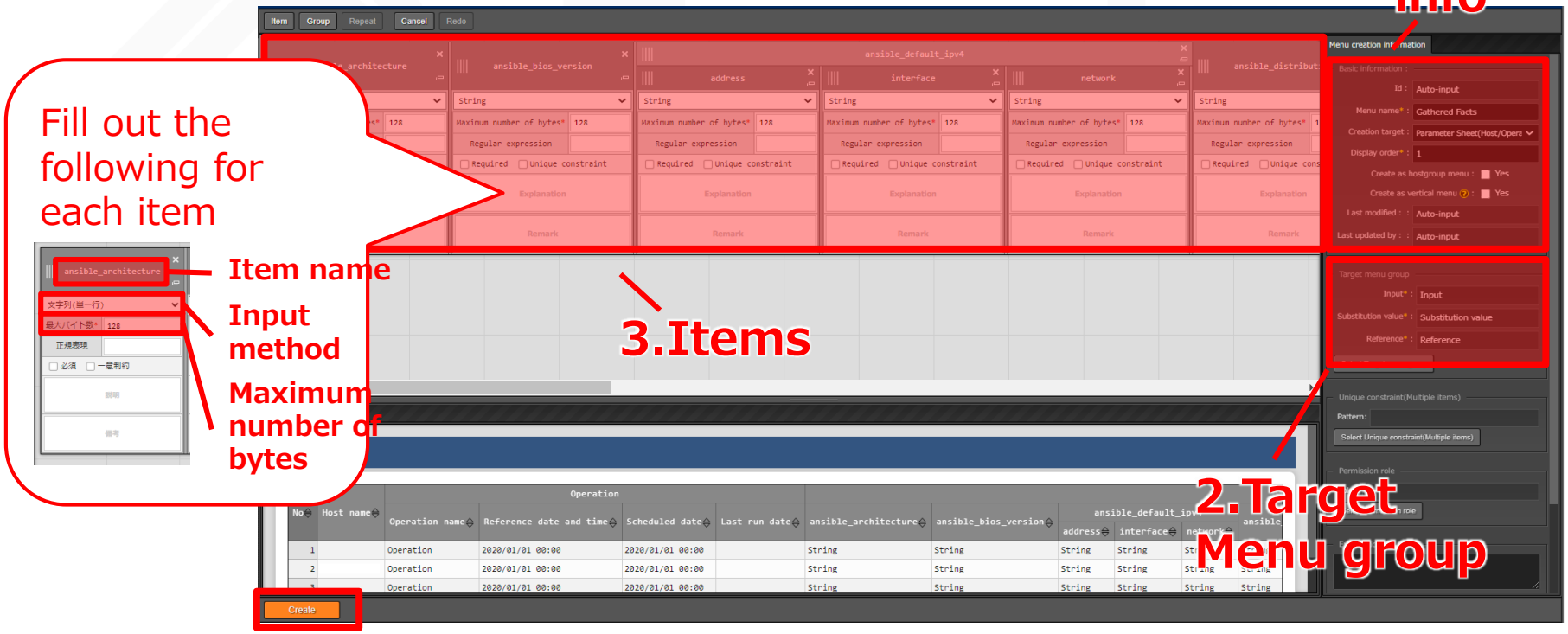
Create Parameter sheet that registers collected values.

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.



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

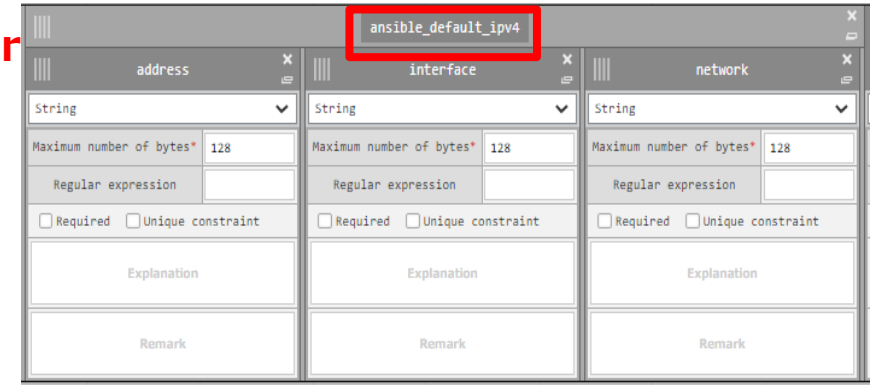
Item name（Free field）	Input method	Maximum number of bytes（Free value）
ansible_architecture	String	128
ansible_bios_version	String	128
ansible_default_ipv4 ＞ address（※）	String	128
ansible_default_ipv4 ＞ interface（※）	String	128
ansible_default_ipv4 ＞ network（※）	String	128
ansible_distribution	String	128
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	最大バイト数（任意の値）
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



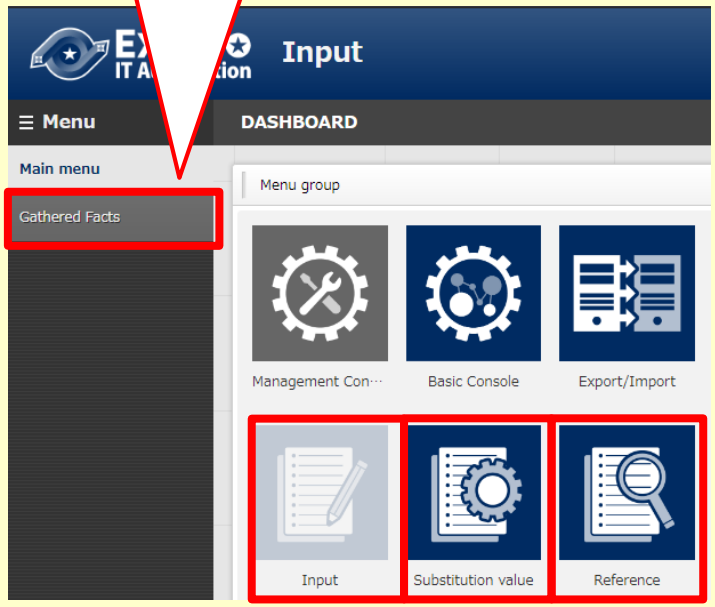
Create a column group called “ansible_default_ipv4 “ and put the following columns in it. [address] , [interface] and [network]

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.



1.8 Register Collected item value list（1/3）

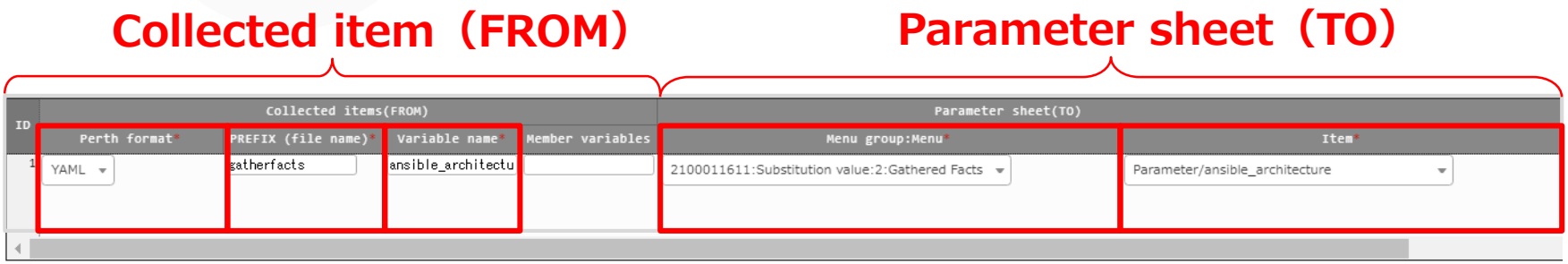
Register Collected item value list

- 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.



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_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

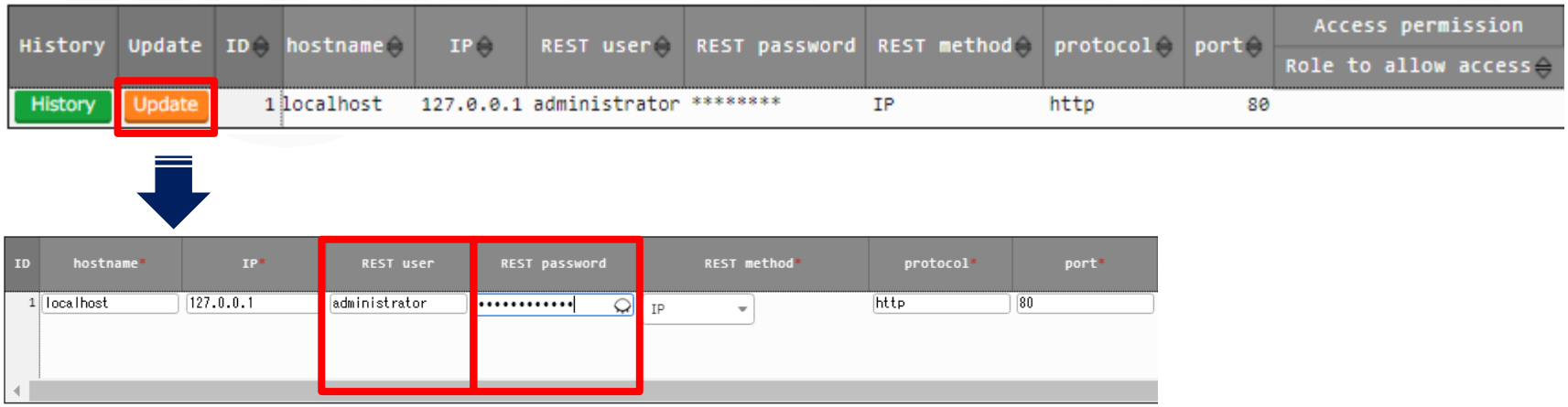
Register Collect interface information

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.



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

1.10 Run operation（1/2）

Run the operation

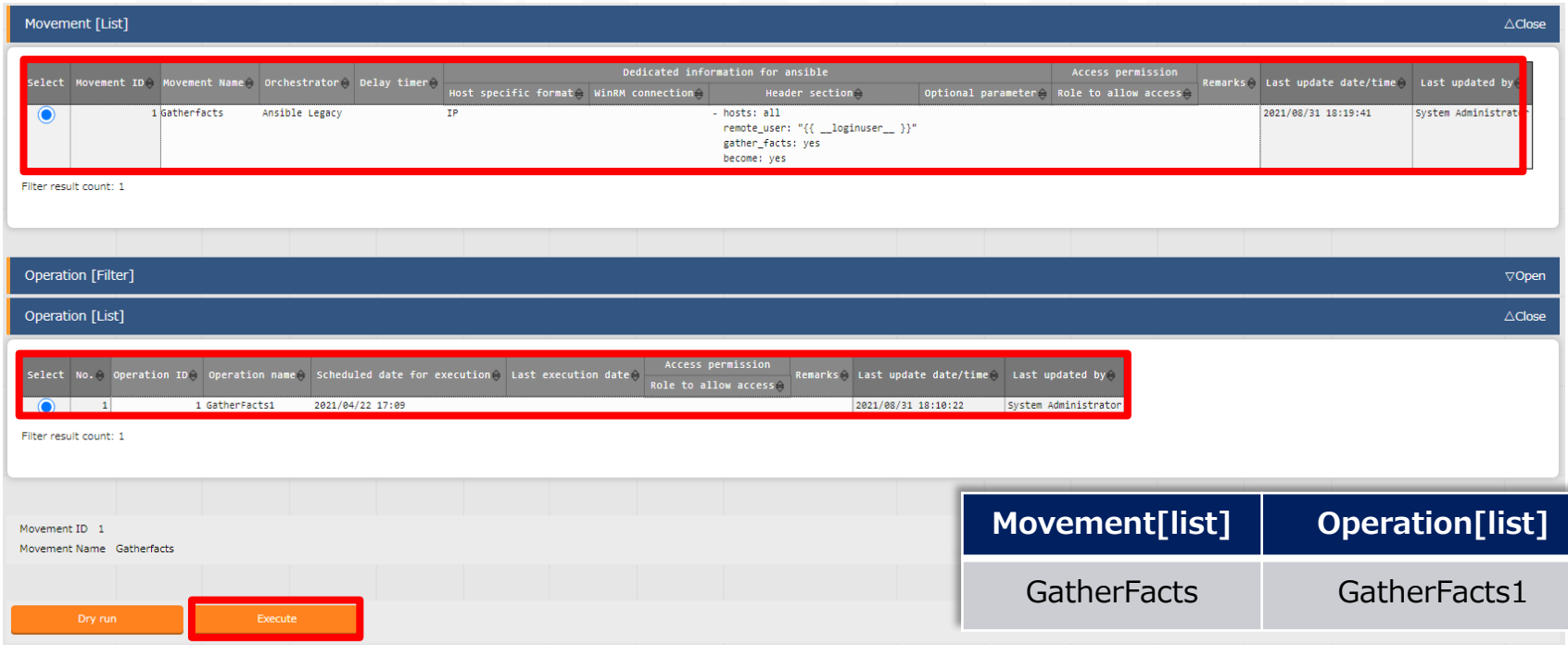
Select Movement and Operation and execute them.

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



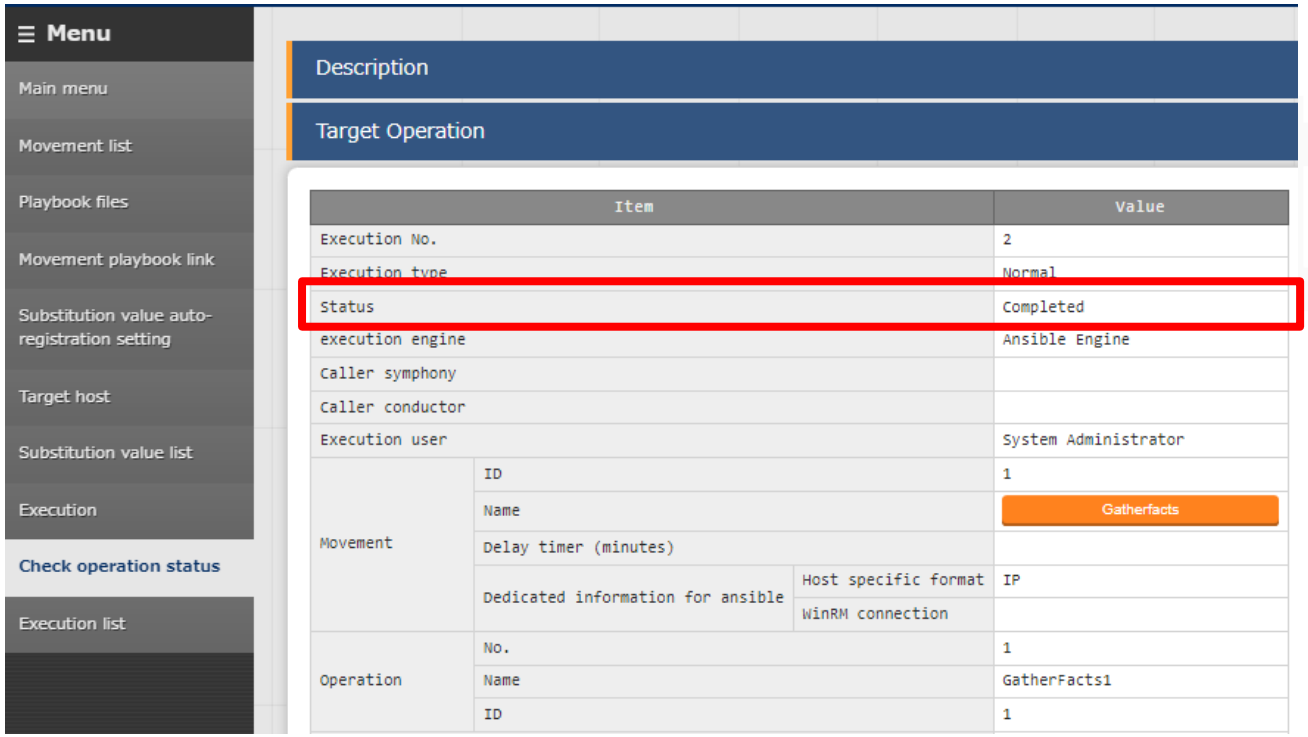
Movement[list]	Operation[list]
GatherFacts	GatherFacts1

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**



1.11 Confirm the collection results（1/2）

Confirm the collection results

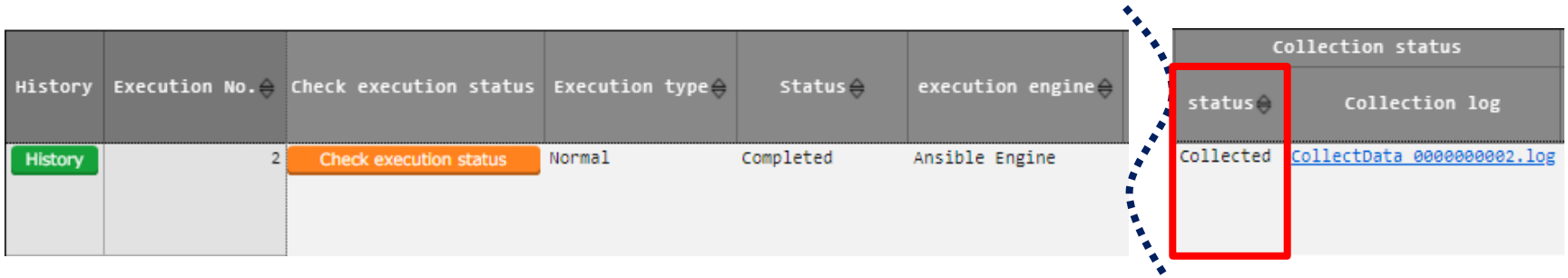
Check if the collection succeeded/failed.

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	
						status	Collection log
History	2	Check execution status	Normal	Completed	Ansible Engine	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	
						ID	Operation name	Reference date	Scheduled date for execution	Last execution date	ansible_architecture	ansible_bios_version
History	Duplicate	Update	Discard	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 > networ		
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

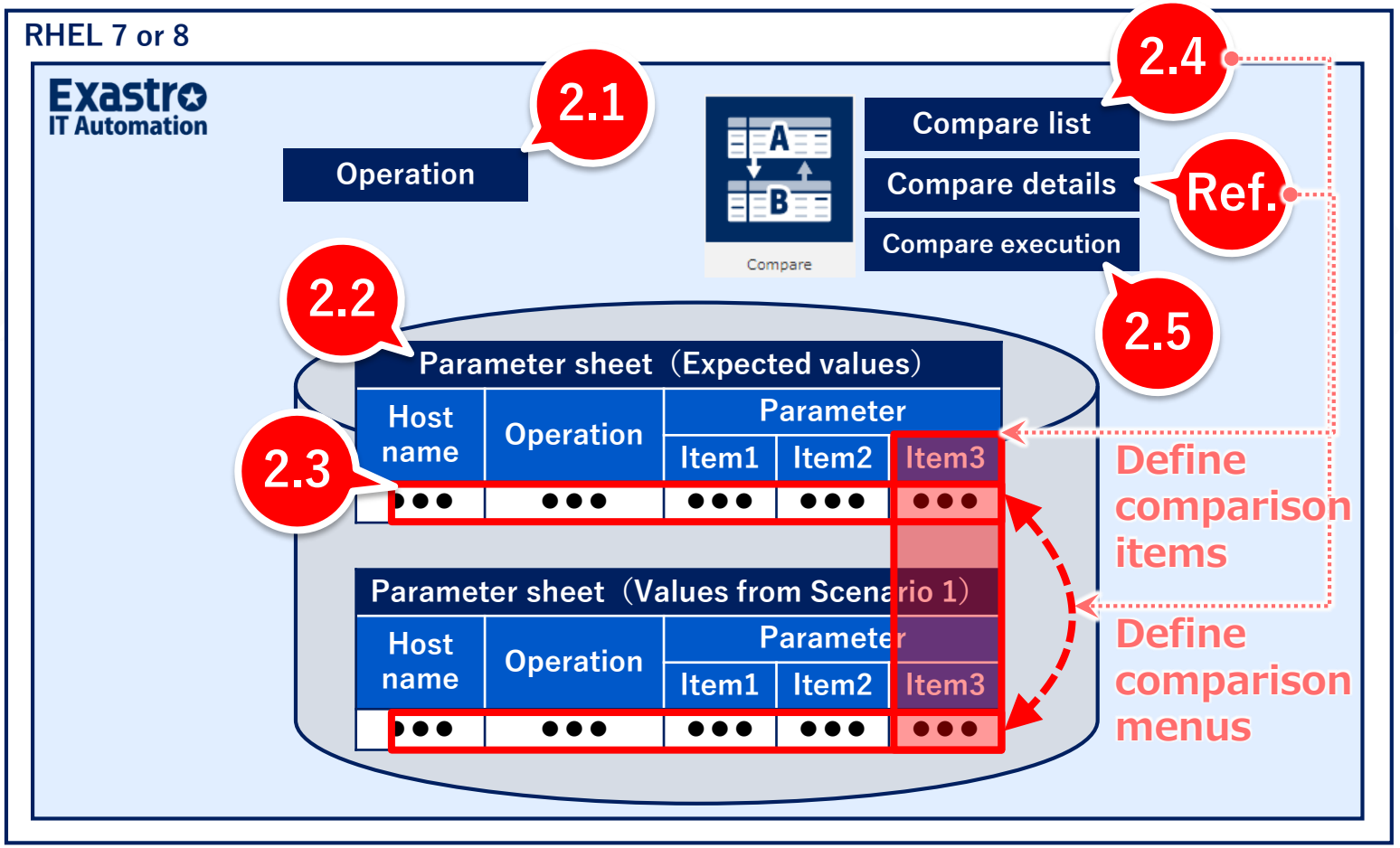
2. Scenario 2 【Compare function】

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

Scenario 2 workflow

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

Overall diagram



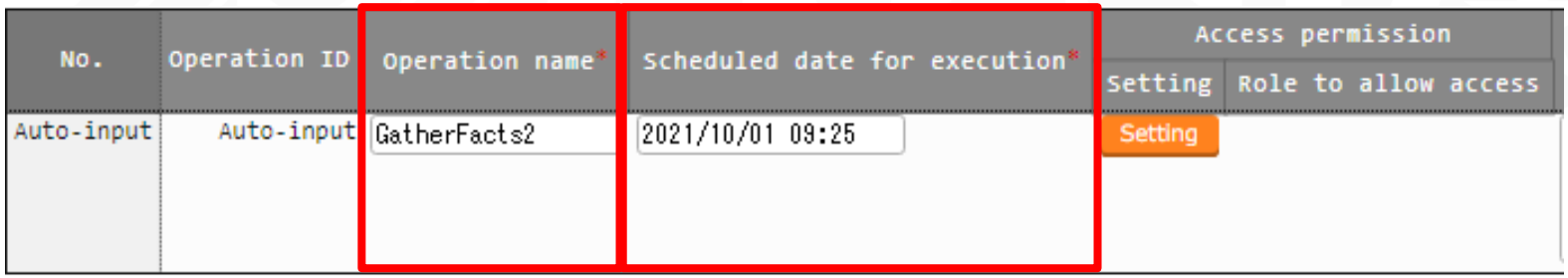
Register Operation

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.



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

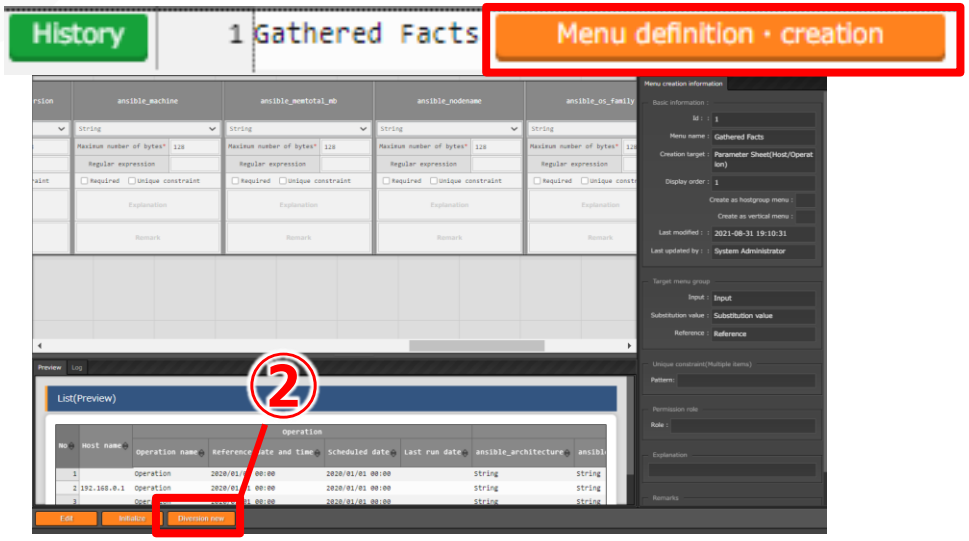
Menu: **Create menu ＞Menu definition information**

① Press the “Filter” button and look for the “Gathered Facts” menu under “list”. After that, press the “Menu definition / creation” button.

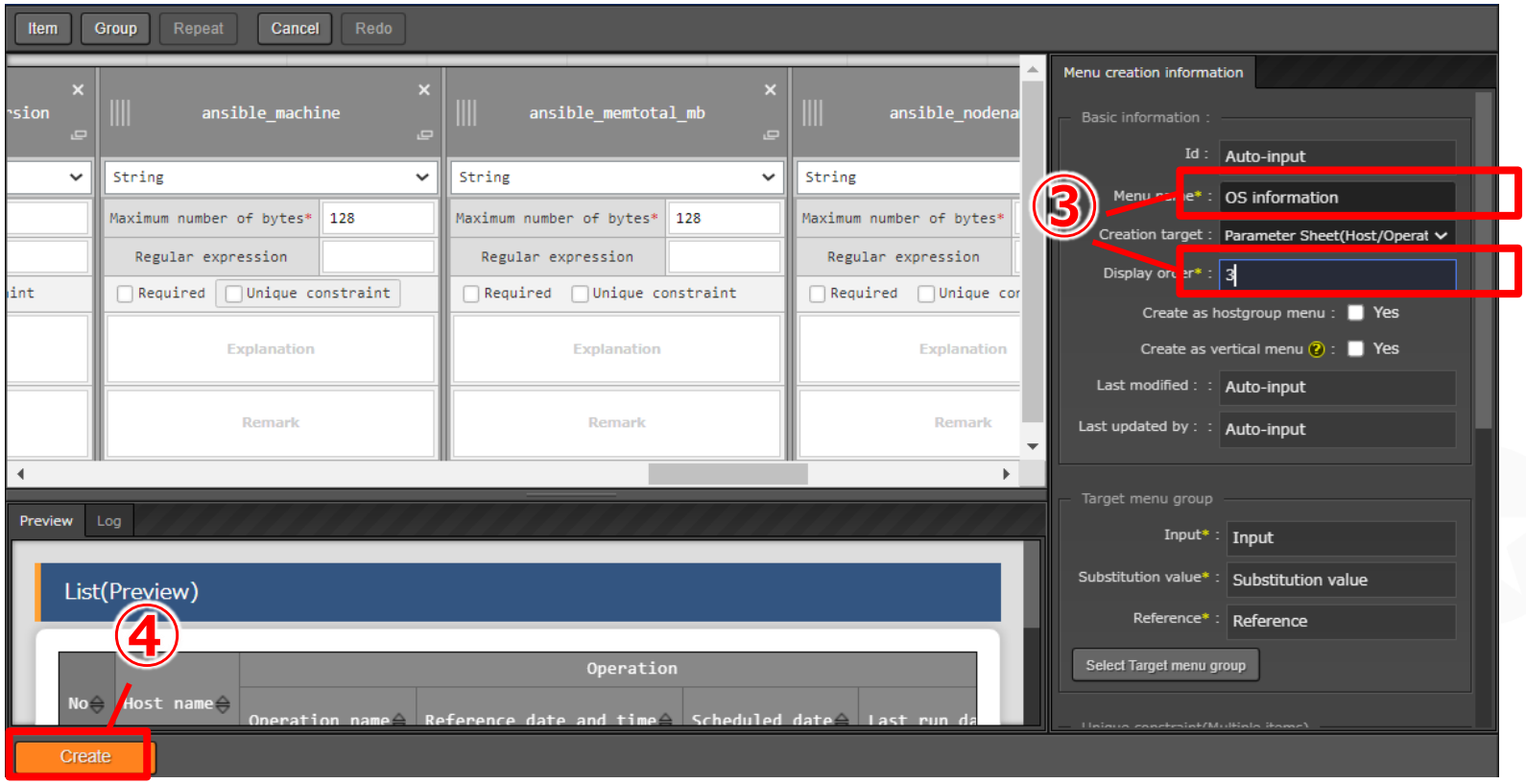
② After the Menu definition screen appears, press the ［Diversion new］button.

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.



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

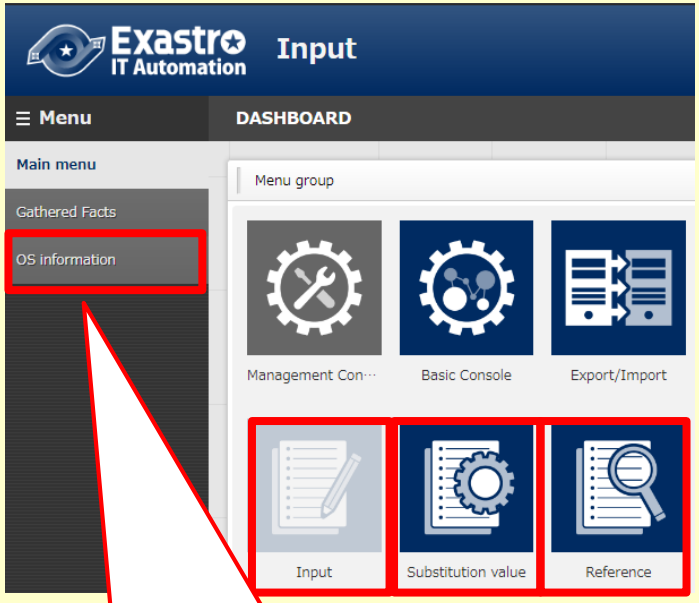


Menu name (Free space)	Display order
OS information	3

The display order can be any number.

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

Created menu



The “OS Information” menu has been created.

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

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.

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.

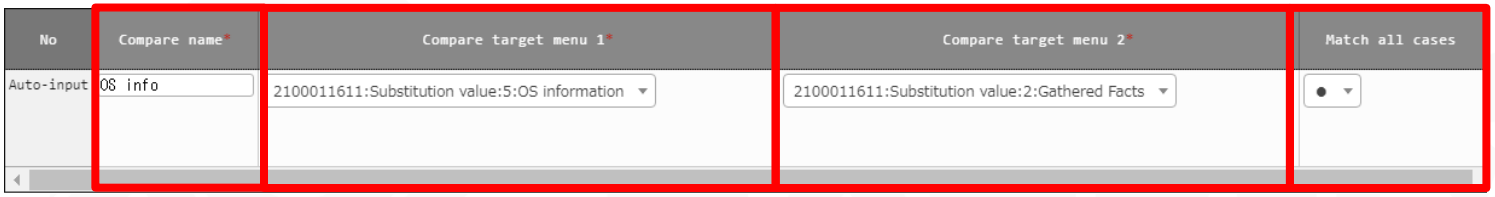
Select the two menus you want to compare

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.



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	●

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

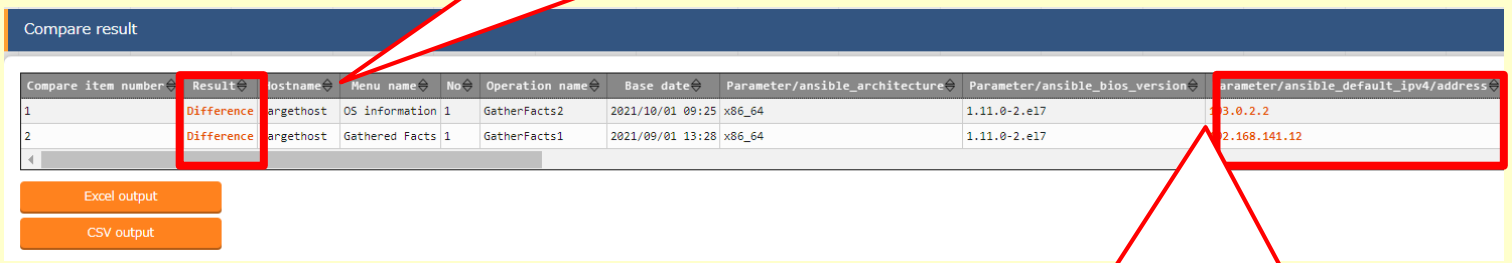
Comparison definition	Standard date 1	Standard date 2	Output
OS information-Gathered Facts	Blank	Blank	ALL

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”.

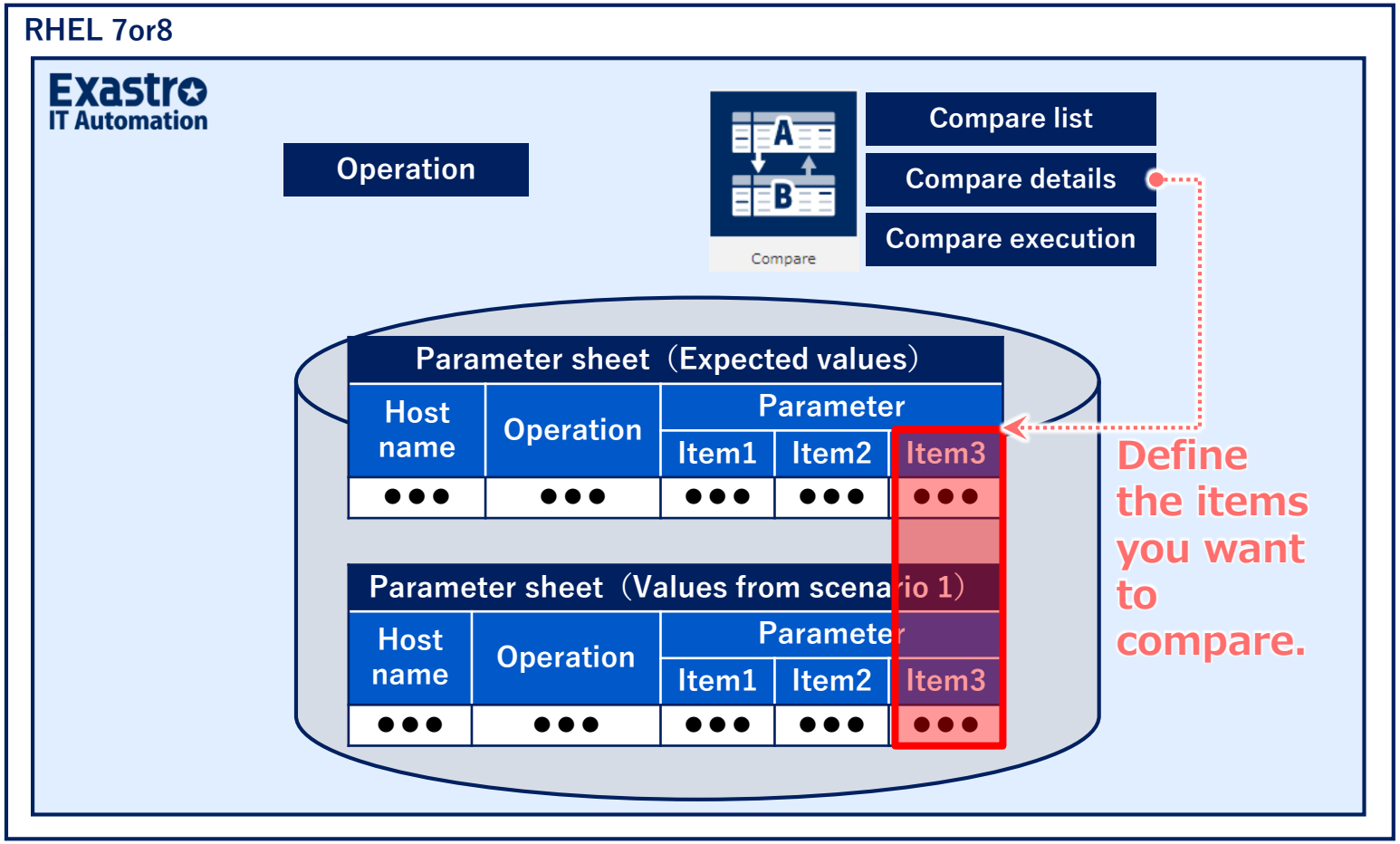


The address, which has different values, will be displayed in red.

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



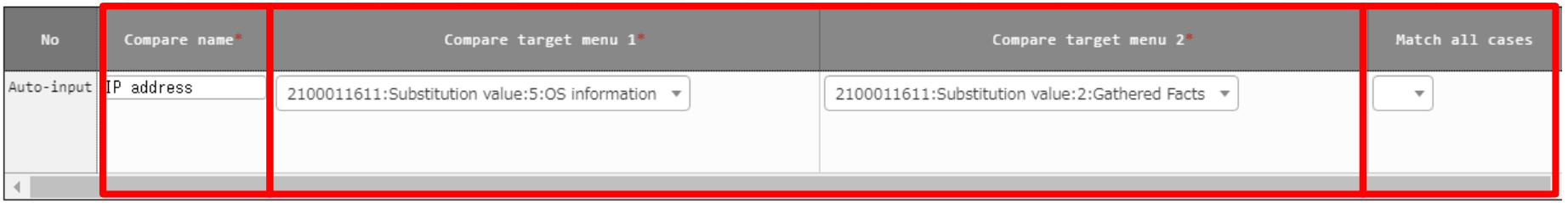
Select the 2 menus you want to compare

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.



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

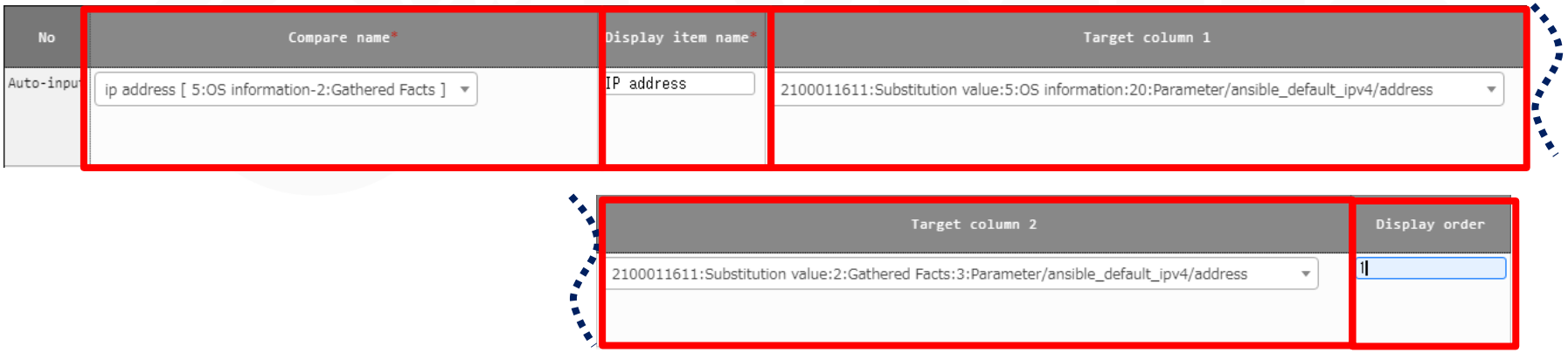
Select the items you want to compare

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.



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

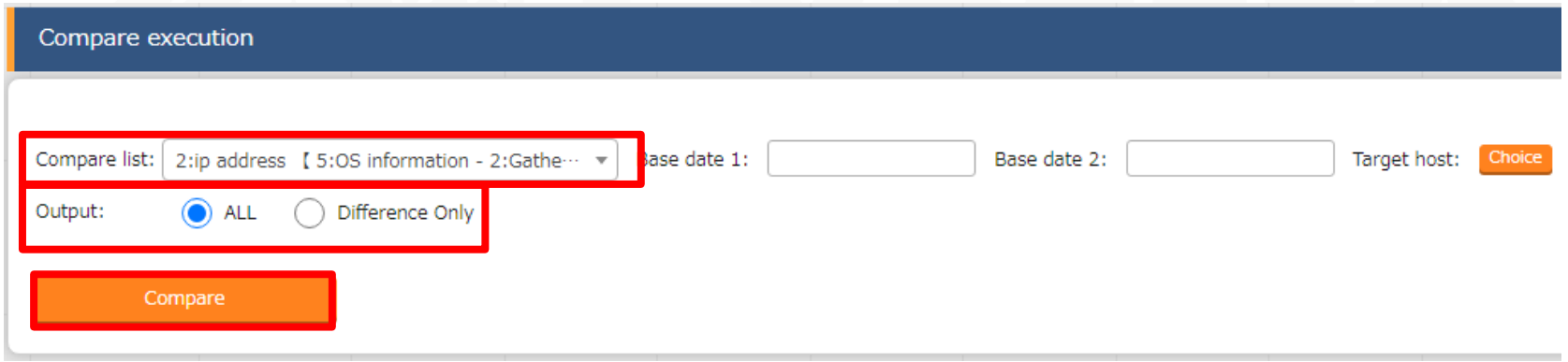
Run the comparison

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.

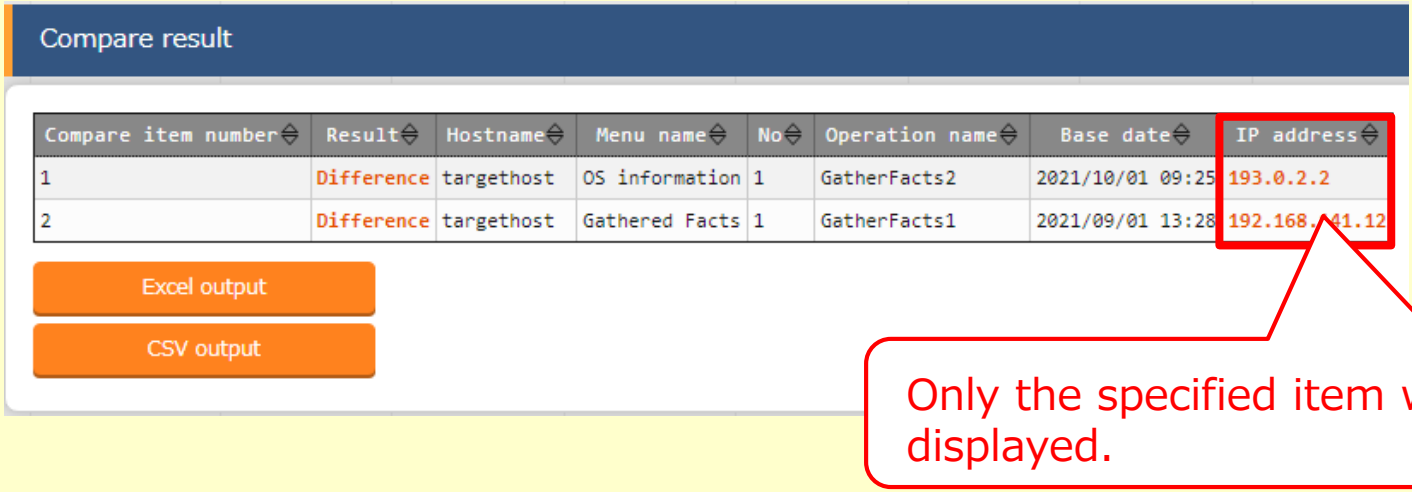
② The comparison results will be displayed



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



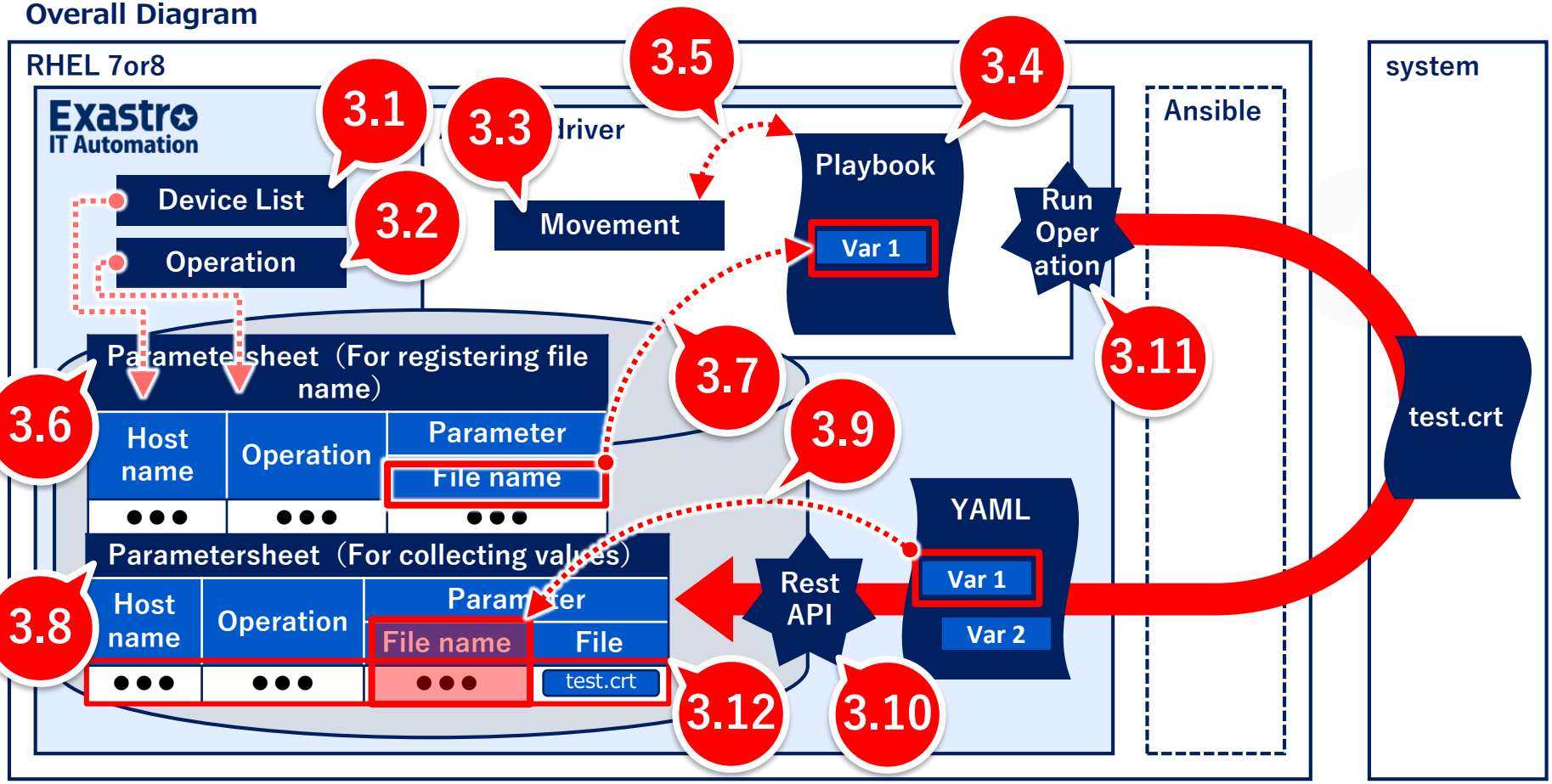
3. Scenario 3 【Collect function】
Collect the target host’s SSL certificate file

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



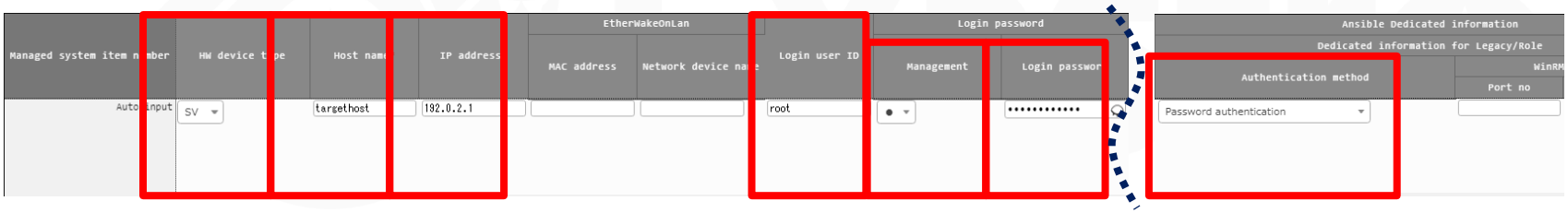
Register target host connection information

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

Menu: **Basic Console> Device List**

① Press “Register” -> “Start registration.

② Input the following information and press 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	Authentication method
SV	targethost	192.0.2.1	root	●	********		Password Authentication

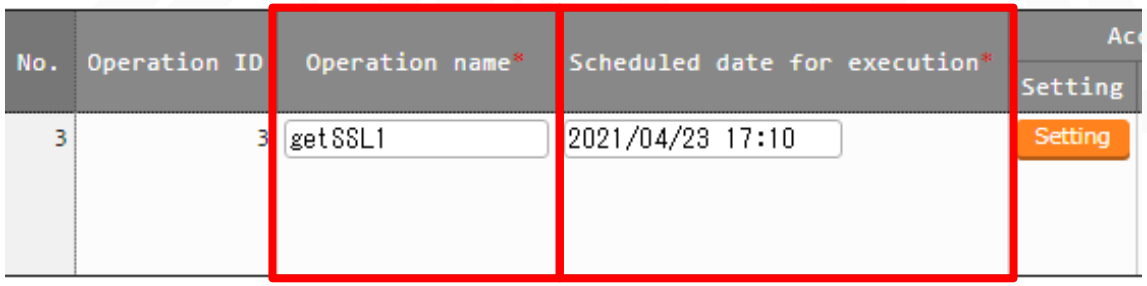
Register operation

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.



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

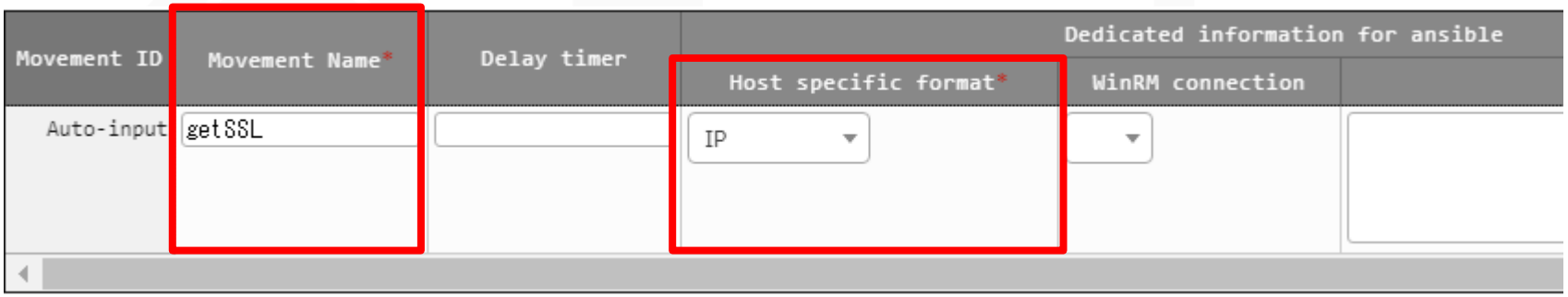
Register Movement in Ansible-Legacy

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

Create Playbook

- This Playbook creates an YAML file that will collect the SSL certificate file and then copies said file to the Collect directory.
- For more information regarding the directory where the file will be moved to, please refer to “1.4.1 Directory for YAML files and collection”

File name : getSSL.yml

```
- name: make yaml file
  blockinfile:
    create: yes
    mode: 644
    insertbefore: EOF
    marker: ""
    dest: "{{ __parameter_dir__ }}/{{ inventory_hostname }}/getSSL.yml"
    content: |
      SSL_file_name                 : {{ VAR_ssl_name }}
      SSL_file                      : {{ VAR_ssl_name }}
  delegate_to: 127.0.0.1

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

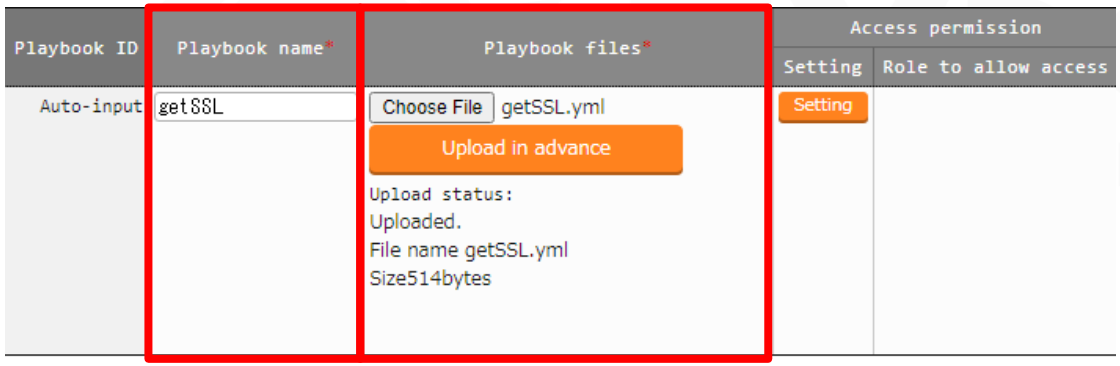
Register Playbook in Ansible-Legacy

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 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

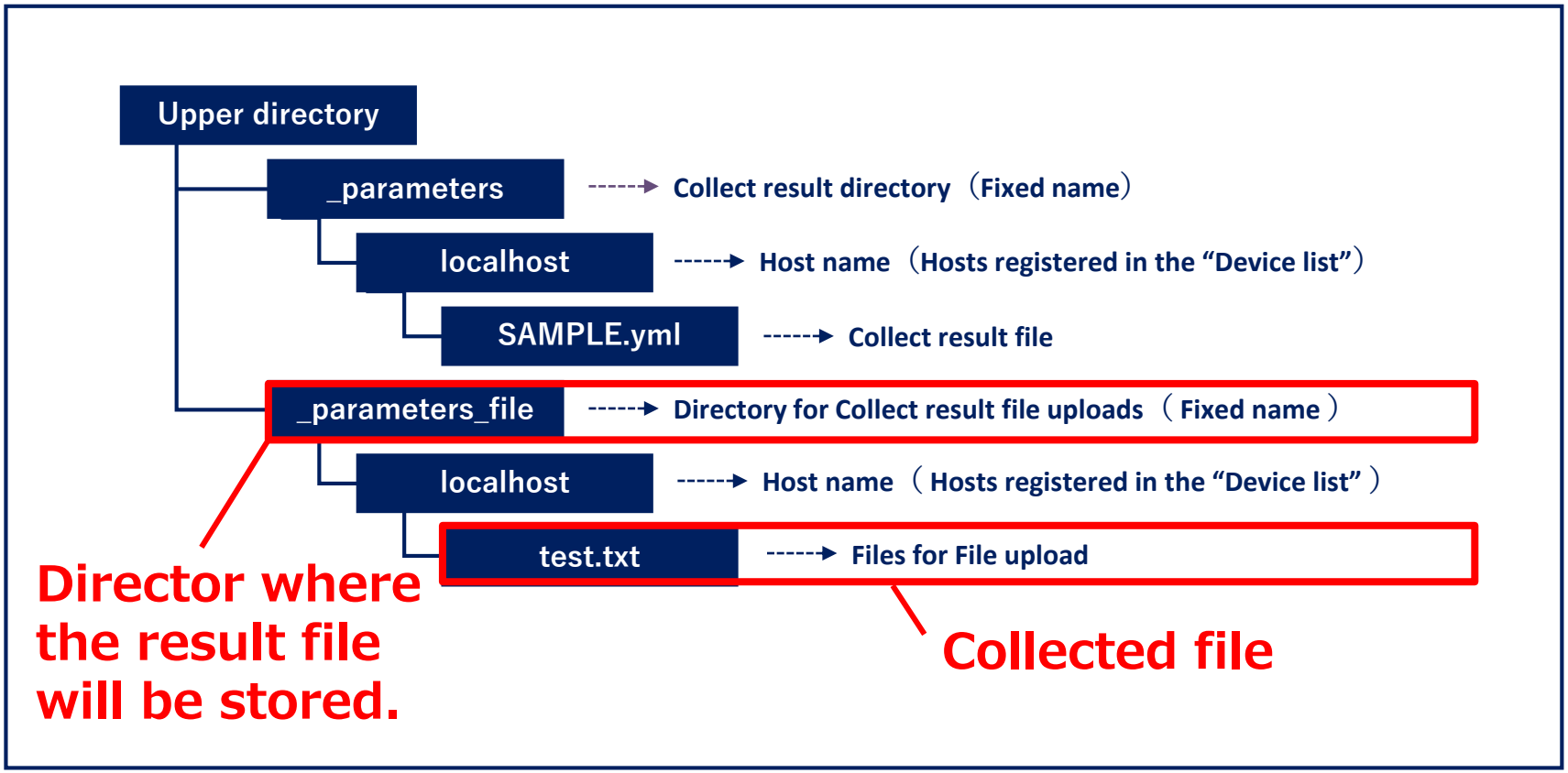
Path variables

	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



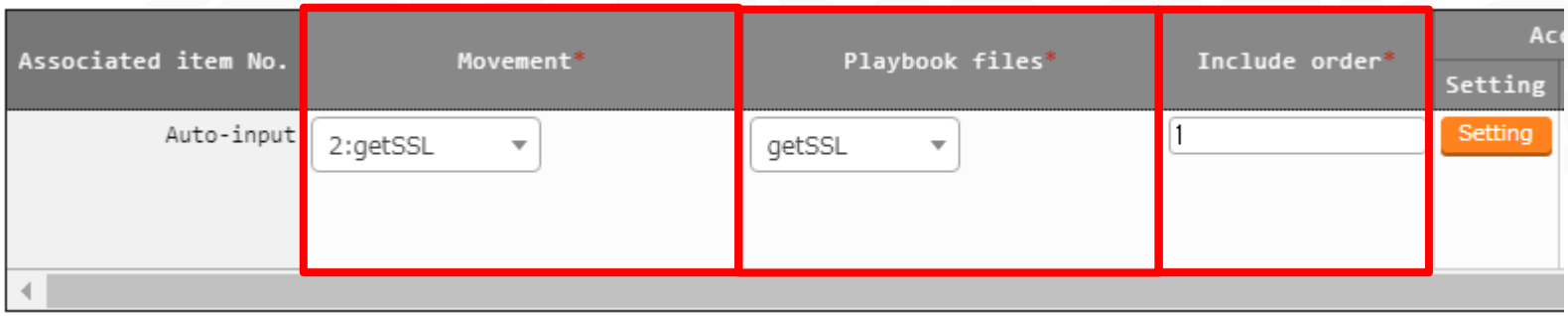
Link Movement and Playbook

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.



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）

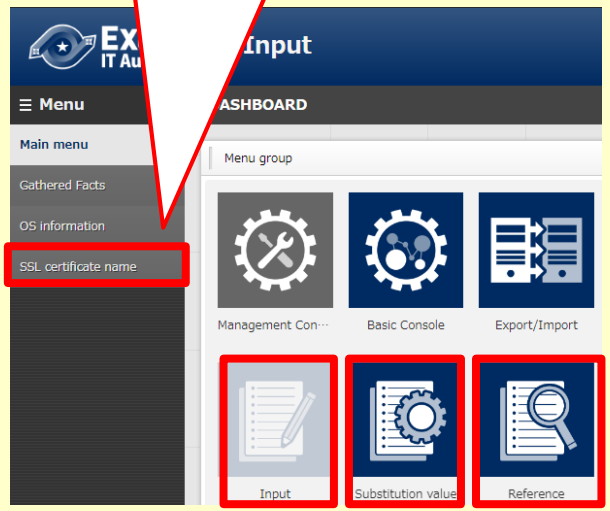
3.Item

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

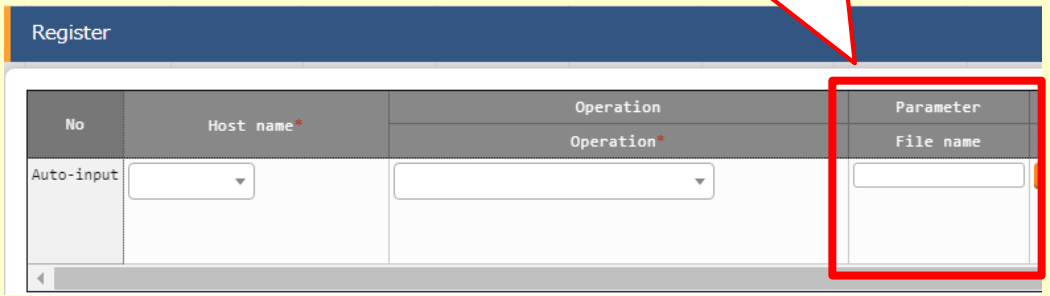
3.6 Register File name (2/3)

Created menu

The menu [SSL certificate name] has been created



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



3.6 Register File name（3/3）

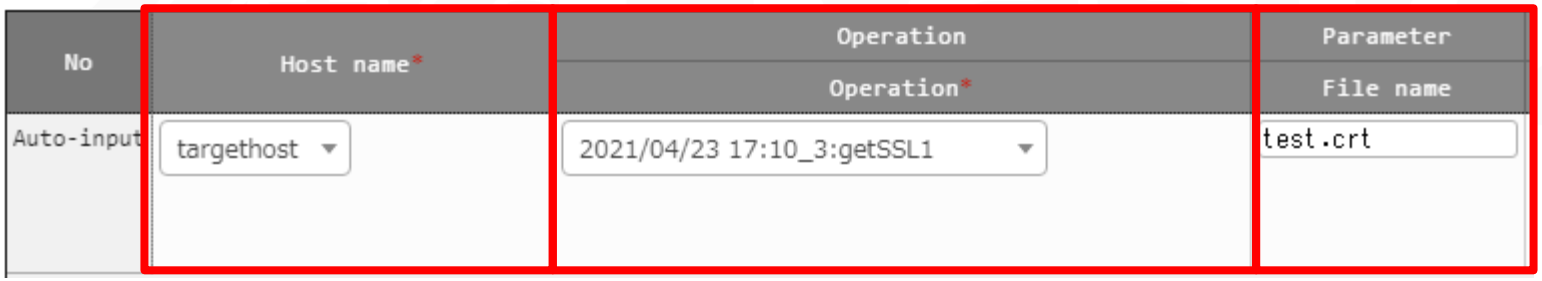
Register File name

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.



Host name	Operation	Parameter
		File name
targethost	getSSL1	test.crt

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.

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

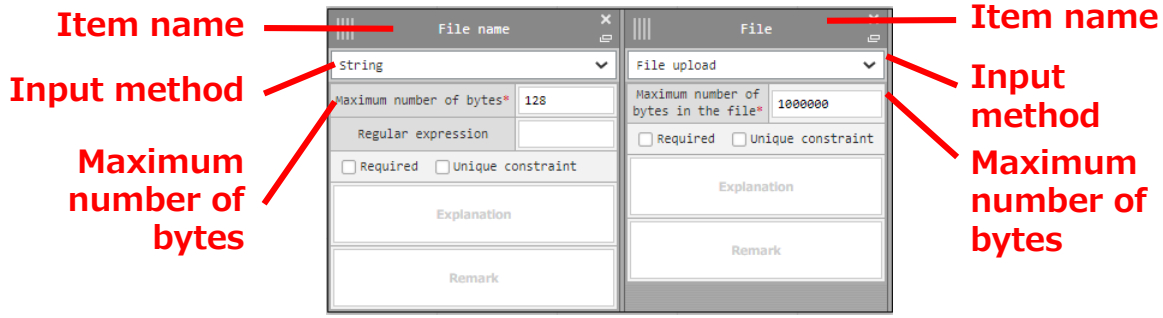
Input	Substitution value	Reference
Input（Default）	Substitution value（Default）	Reference（Default）

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

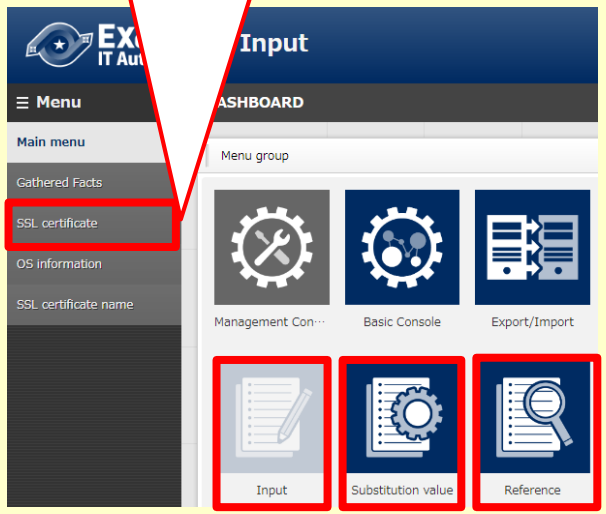


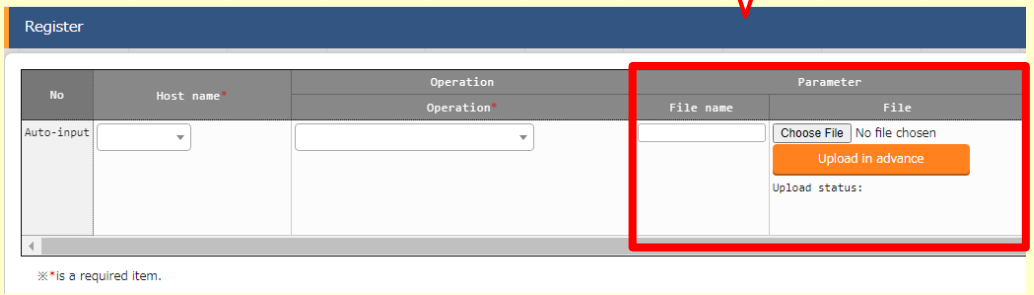
3.8 Create Parameter sheet for collect values (3/3)

Created menu

The menu “SSL certificate” has been created.

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





Register Collected item value list

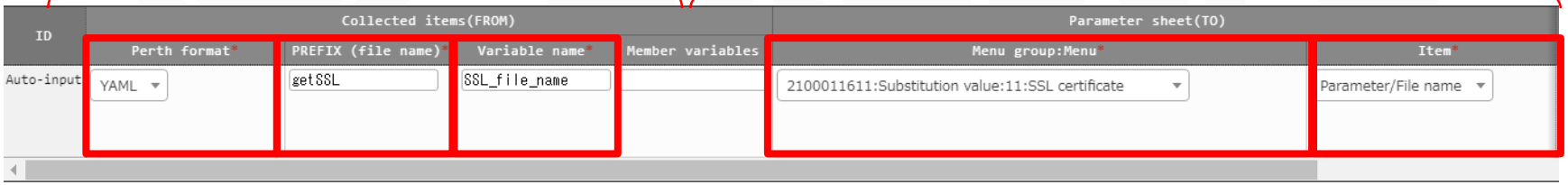
- Configure the parameter sheet so that the collected items will automatically be registered to the parameter sheet.
- Link the collect item's (FROM) YAML file name, variable name and the Parameter sheet's (TO) menu name and Item name. Do this for both the "File name" and "File" items.

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 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

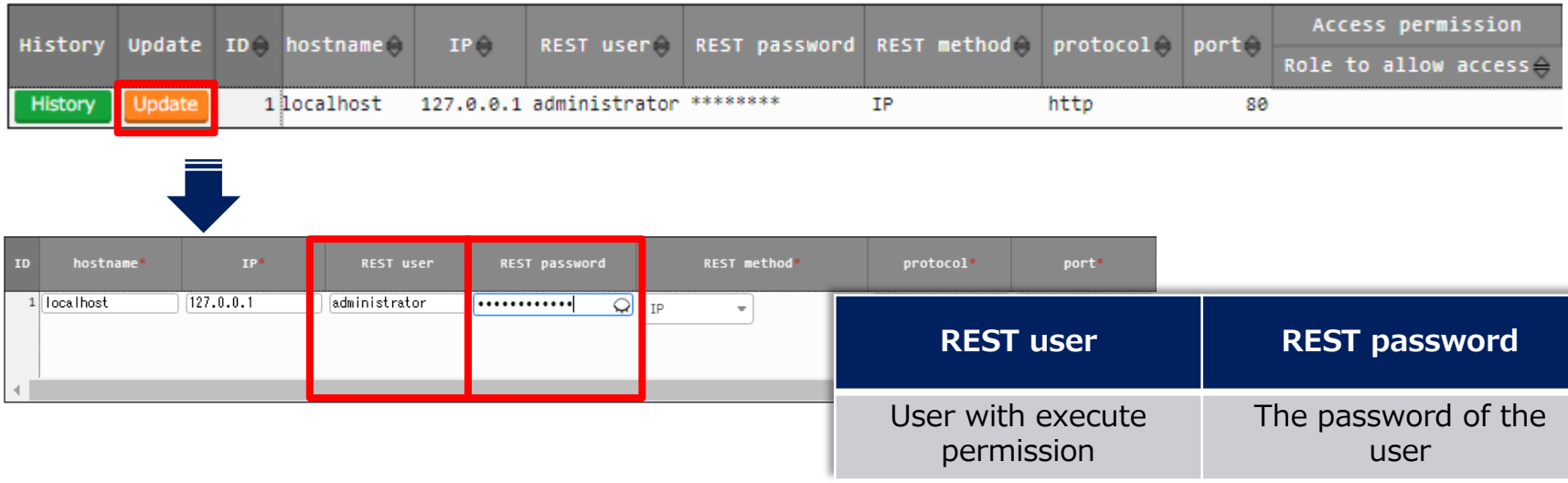
Register Collect interface information

- 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



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

3.11 Run operation（1/2）

Run operation

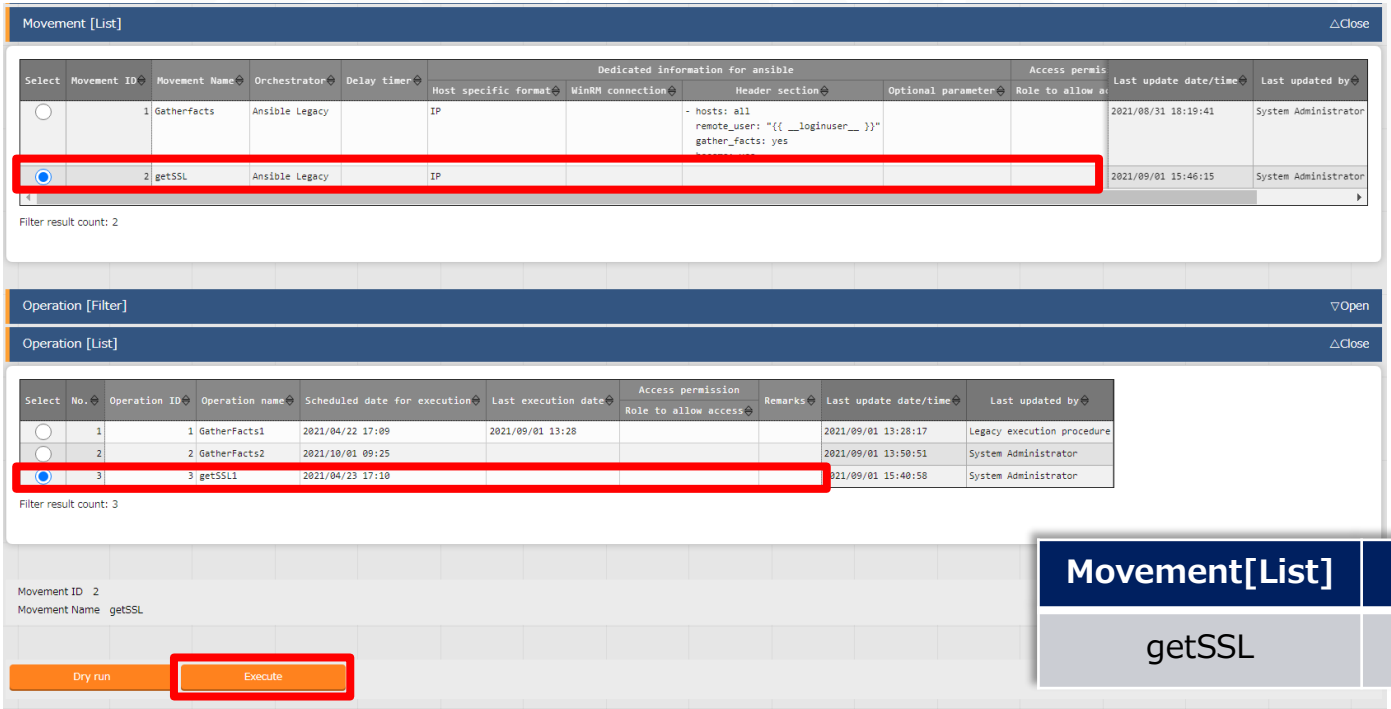
Select Movement and Operation and execute them.

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

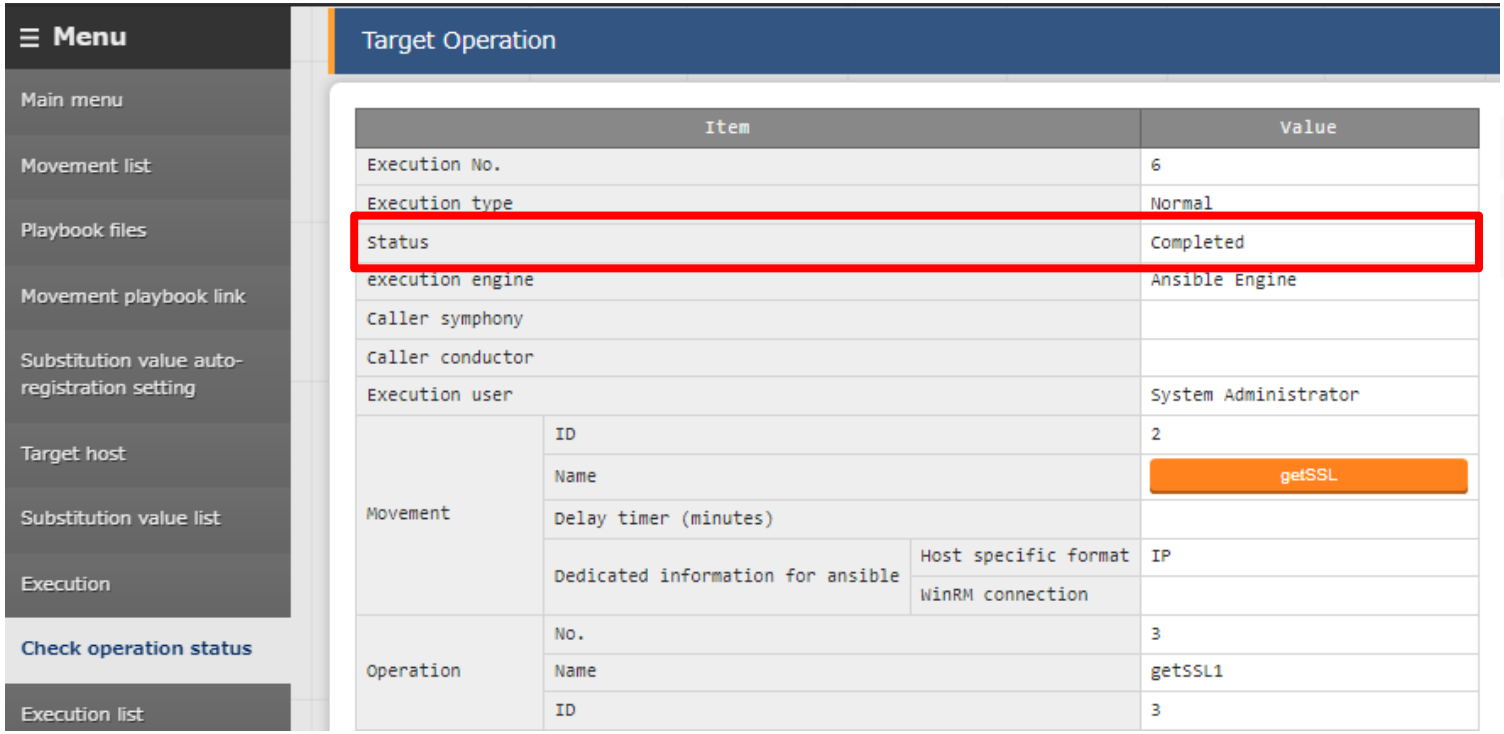


Movement[List]	Operation [List]
getSSL	getSSL1

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**



Confirm the collection results

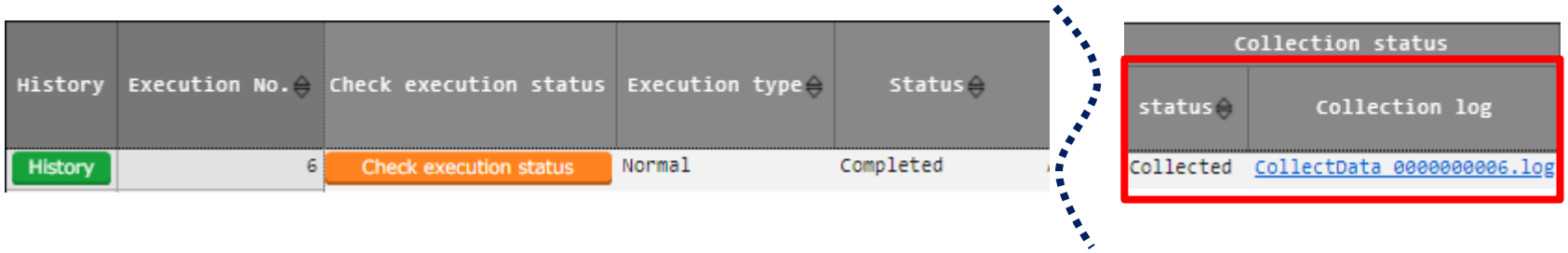
Check if the collection succeeded/failed.

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



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.

History	Duplicate	Update	Discard	No	Host name	Operation					Parameter	
						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】

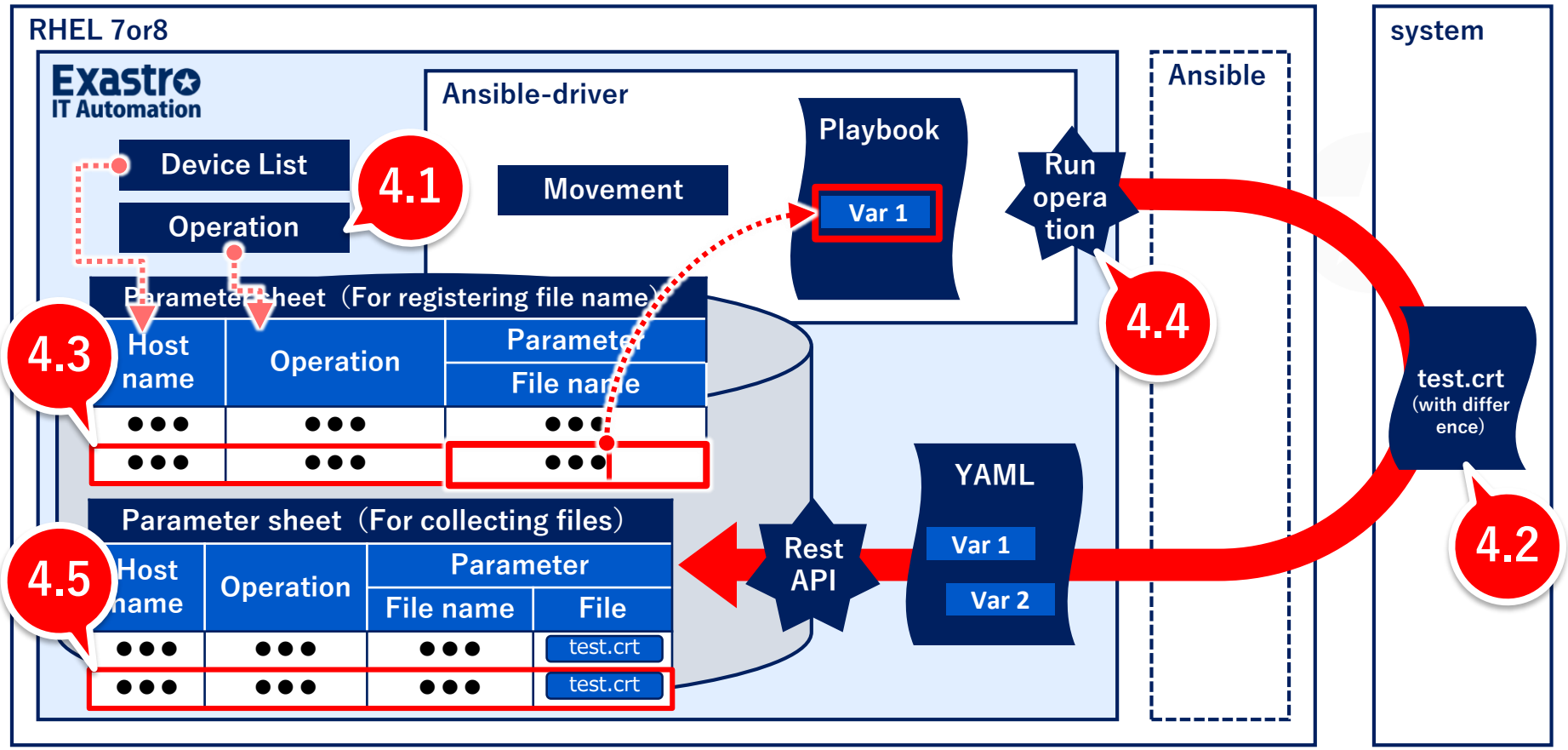
Compare the file downloaded in scenario 3 with the same file from a different date.

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.

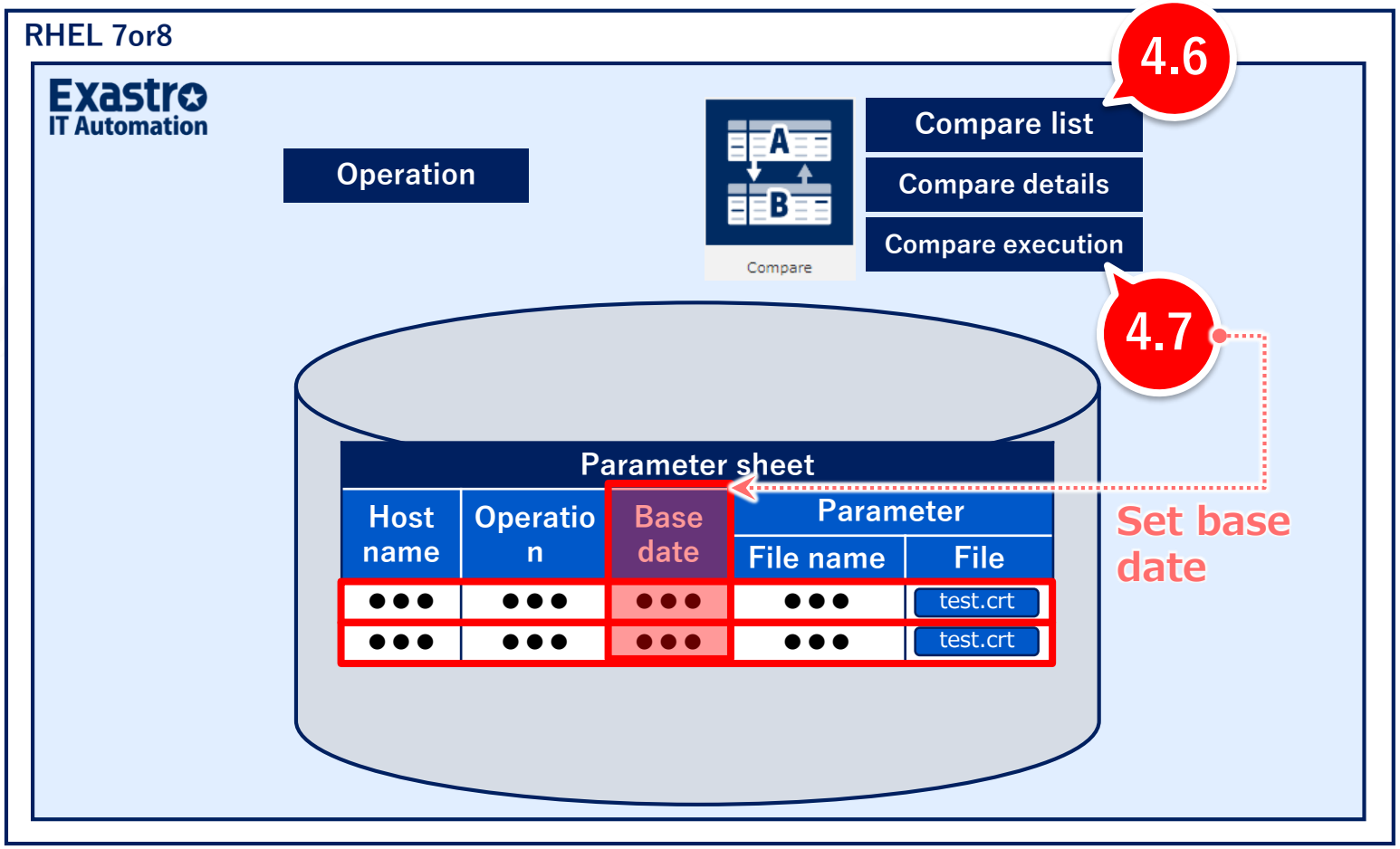
Overall diagram



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）



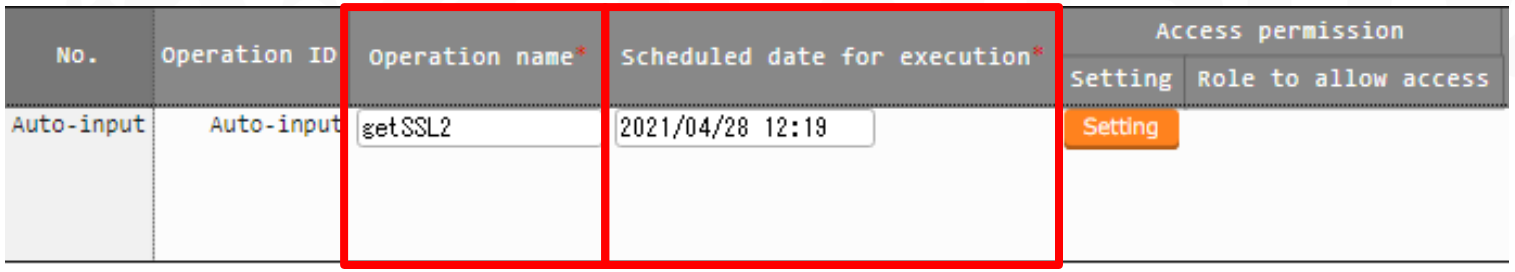
Register operation

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.

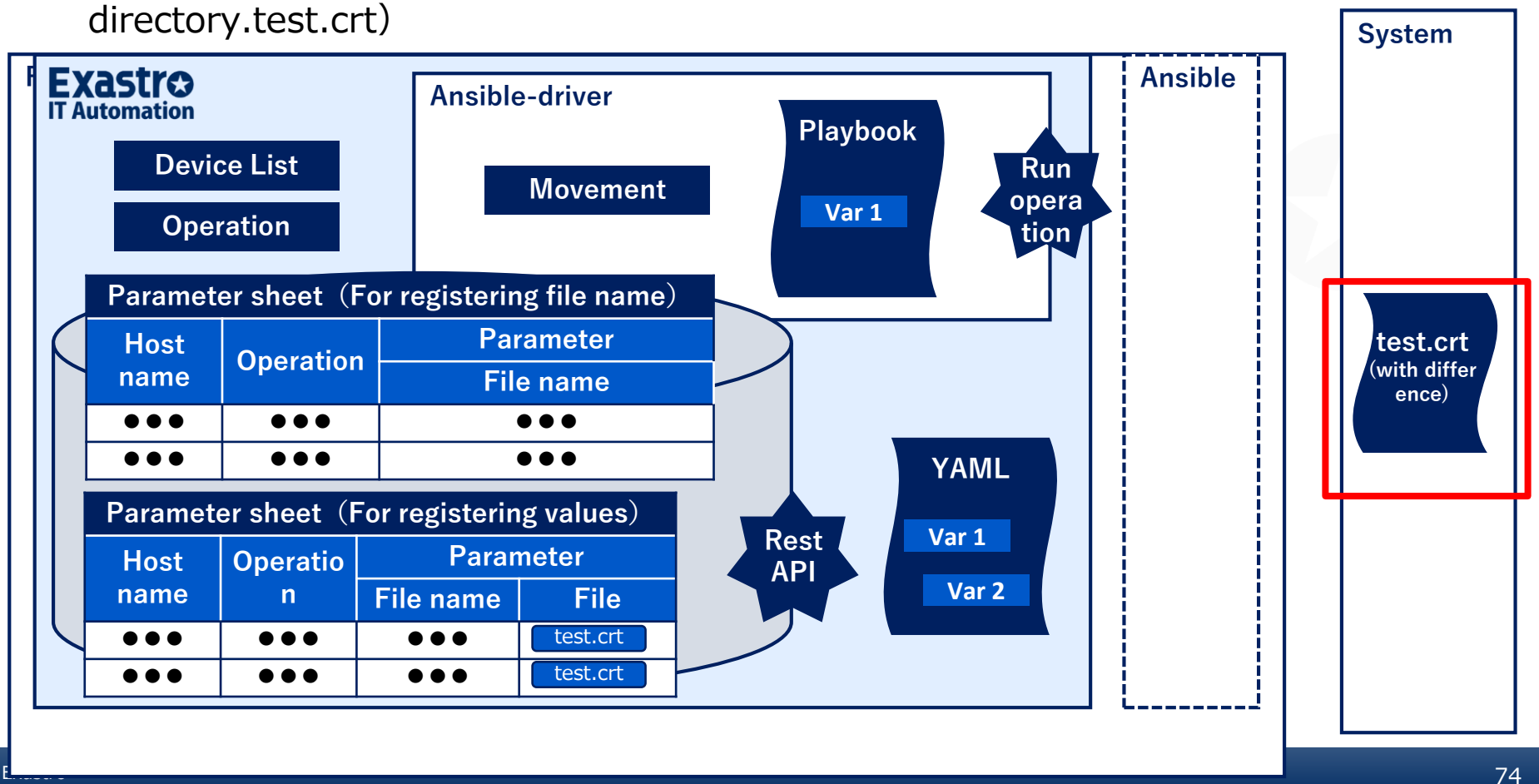


Operation name (Free space)	Scheduled date for execution (Free space)
getSSL2	2021/04/28 12:19

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)



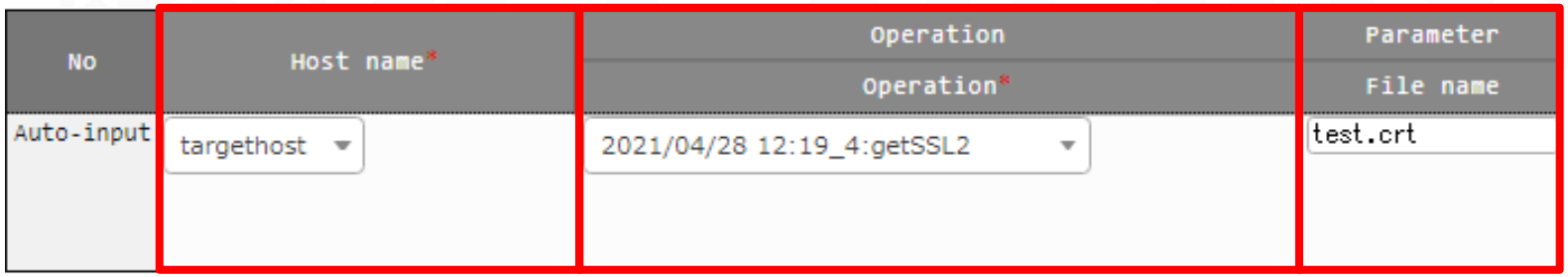
Register file name

- 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.



Host name	Operation	Parameter
		File name
targethost	getSSL2	test.crt

Run operation

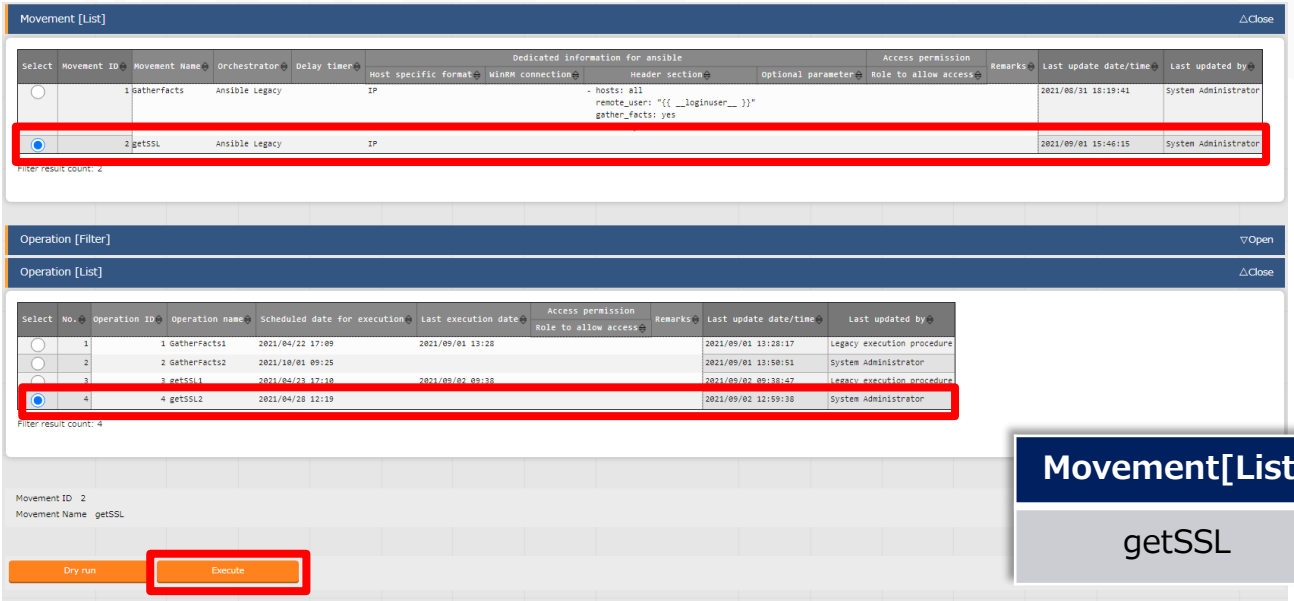
- Collect the SSL certificate with different contents from the one collected in scenario 3.
- Select the same Movement from Scenario 3, “getSSL”. The operation should be the new one we created for comparing, “getSSL2”.

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.



Movement[List]	Operation[List]
getSSL	getSSL2

4.5 Confirm comparison results

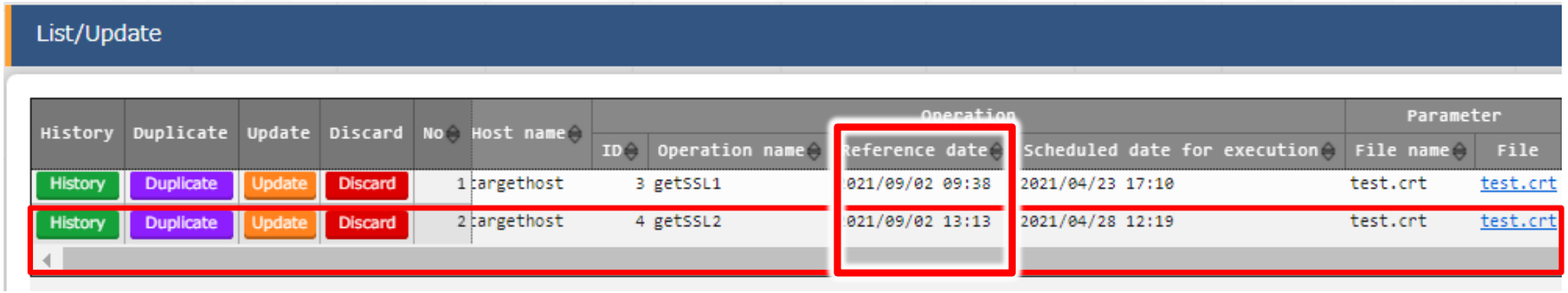
Confirm Parameter sheet

- 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.



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.

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	●

4.7 Run Comparison（1/3）

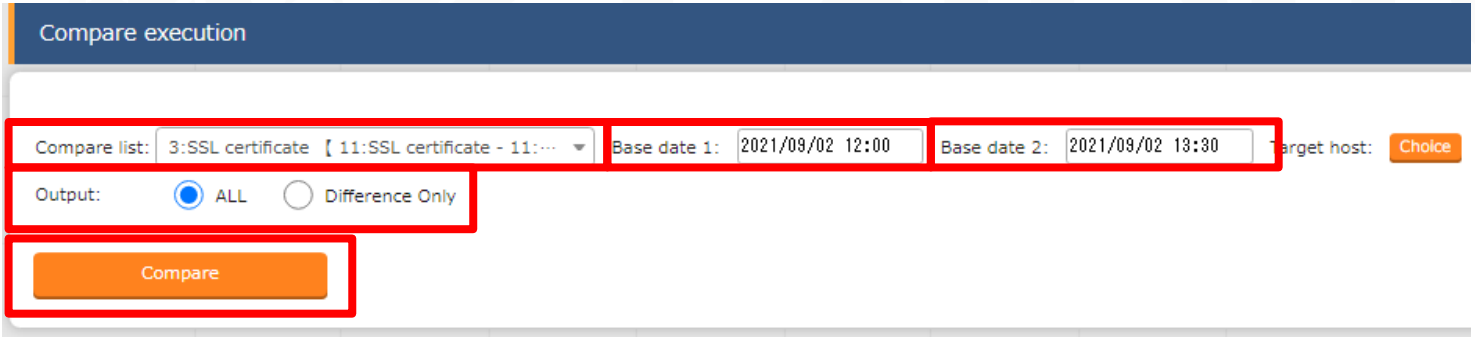
Run the comparison

- 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.

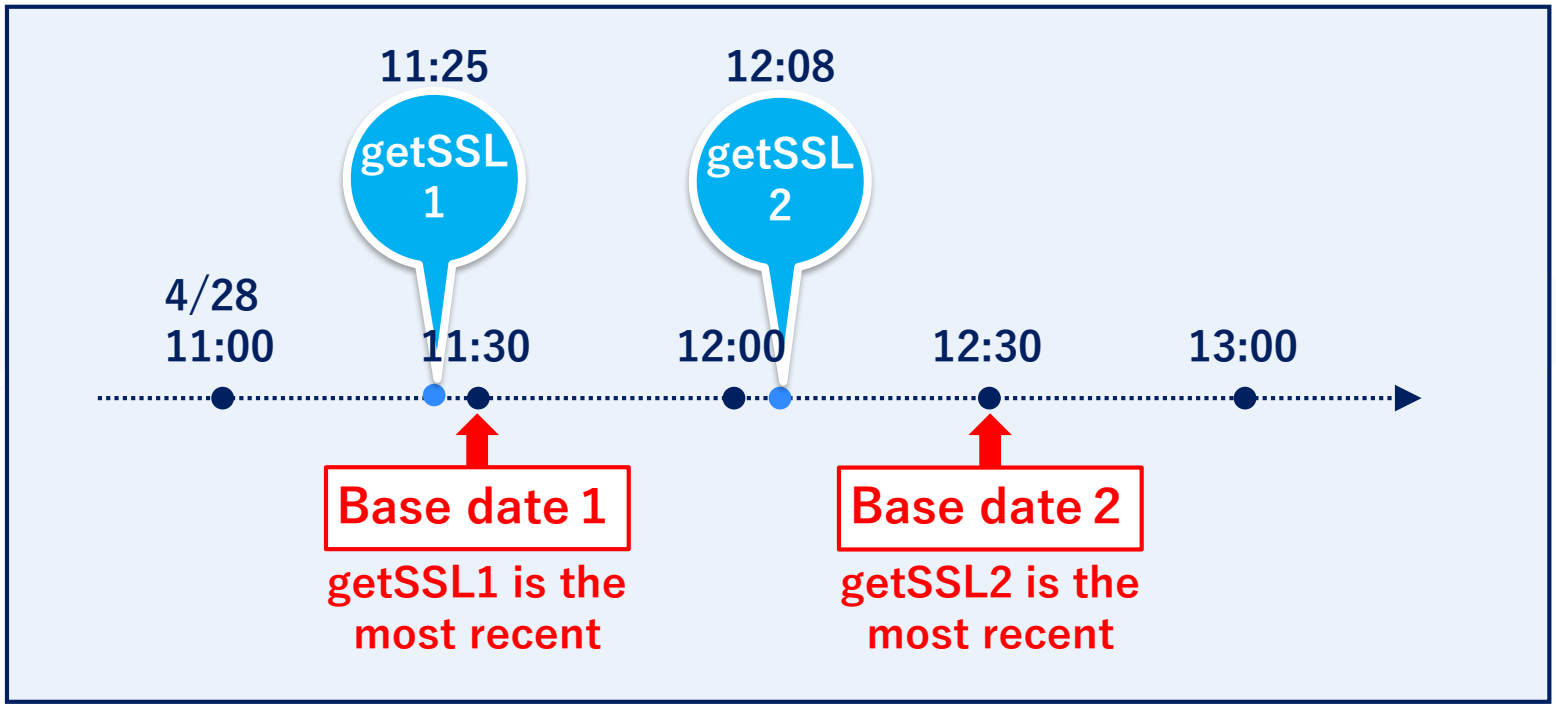
② The Comparison result should be displayed



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)

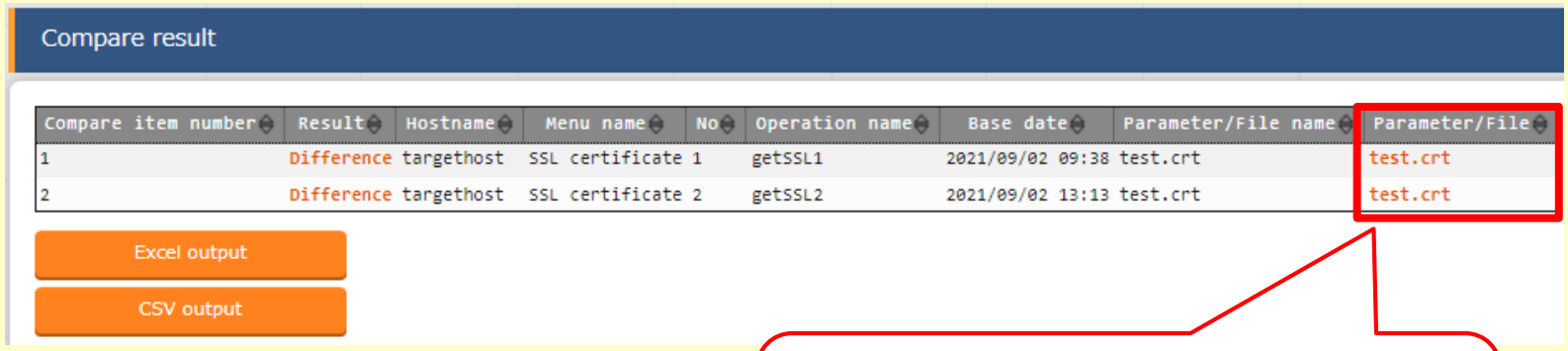
- The Standard dates are displayed below.



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

4.7 Run comparison (3/3)

Compare results



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


Exastro