



# IT Automation

## Collect/Compare function [Practice]

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

Exastro IT Automation Version 1.9  
Exastro developer

# 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.

The screenshot shows the Exastro dashboard interface. On the left is a sidebar with a 'Main menu' containing 'Device list', 'Operation list', 'Movement list', and 'ER Diagram'. The main area is titled 'DASHBOARD' and contains a 'Menu group' with several icons:

- Management Con... (grey)
- Basic Console (grey)
- Export/Import (blue)
- SOD (blue)
- Conductor (blue)
- Create Menu (blue)
- Compare (red box)
- HostGroup manag... (blue)
- Ansible Common (red box)
- Ansible-Legacy (blue)
- Ansible-Pioneer (blue)
- Ansible-LegacyRole (blue)
- Cobbler (blue)
- Terraform (blue)
- CI/CD for IaC (blue)

A red callout box points from the bottom right towards the 'Compare' and 'Ansible Common' icons. Inside the callout box, the text reads:

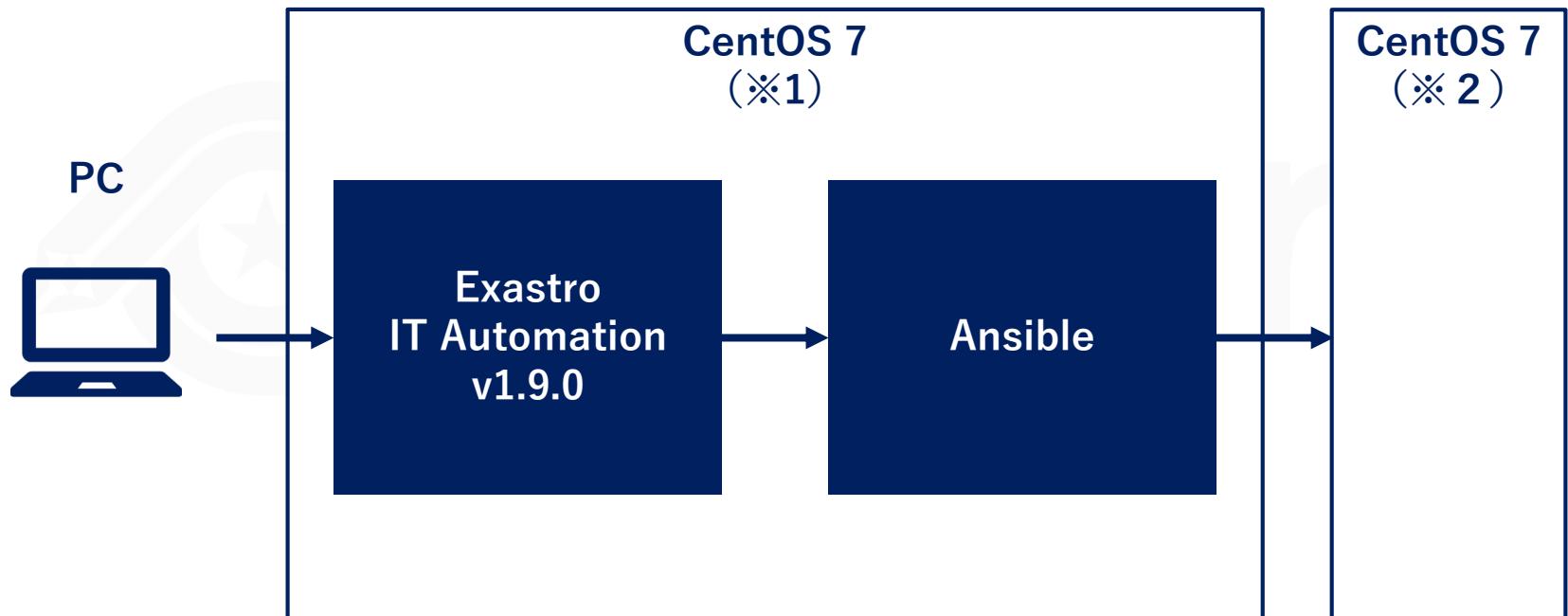
Menus related to the Collect function

- Collect interface information
- Collected item value list

# (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	<b>Scenario 1</b> Collect the target host OS information	<b>Scenario 2</b> Compare the values and the expected values of the one collected in Scenario 1.
Collecting/ Comparing files	<b>Scenario 3</b> Collect the target host's SSL certificate file	<b>Scenario 4</b> 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

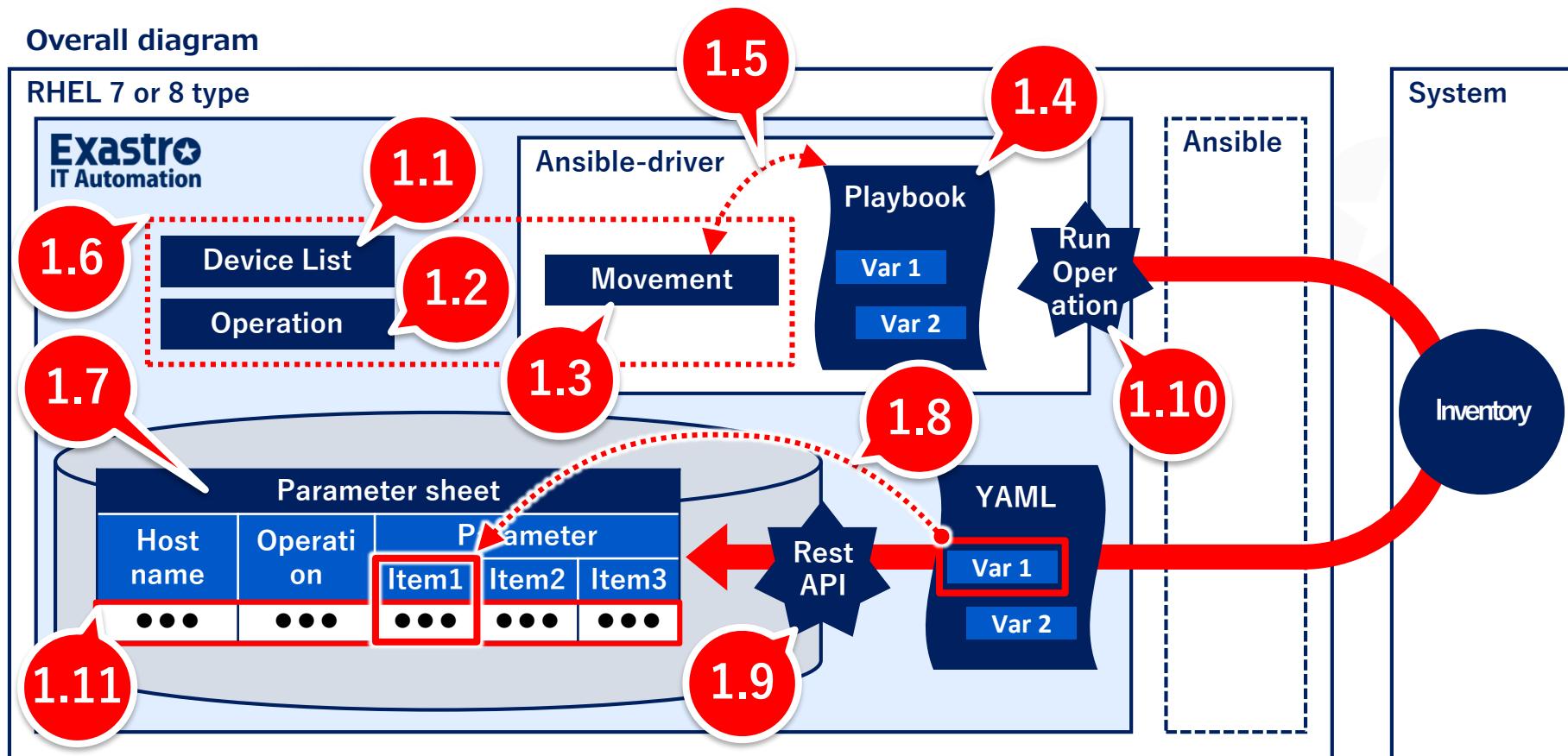


# Scenario 1 Overall diagram

## 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 for a managed system. The fields highlighted by red boxes are:

- HW device type: SV
- Host name: targethost
- IP address: 192.0.2.1
- Login user ID: root
- Management: ●
- Login password: \*\*\*\*\*
- Authentication method: Password authentication

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

# 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	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*	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 (1/2)

## 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"](#)

```
- 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
```

**File name : GatherFacts.yml**

# 1.4 Register Playbook (2/2)

## 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.

The screenshot shows the 'Playbook file collection' screen in Ansible-Legacy. A red box highlights the 'Playbook name\*' field containing 'GatherFacts'. Another red box highlights the 'Playbook files\*' section where a file named 'GatherFacts.yml' has been uploaded. A callout bubble points to the 'Register' button at the top right of the upload area, with the text: '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.

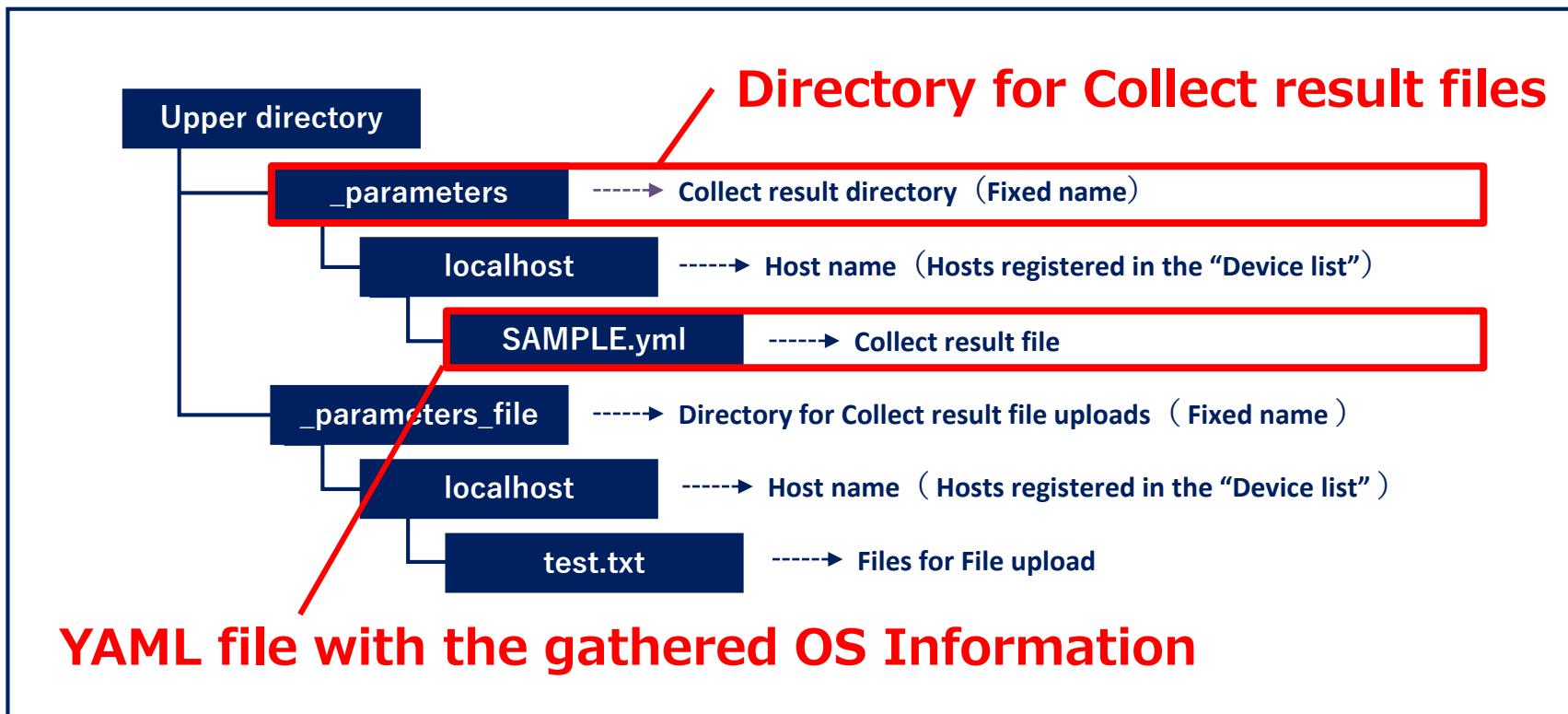
- Since Collect result files in ITA is specified in a YAML file format, we need to create a YAML file.
- The generated YAML file will be stored in the Collection directory specified by the ITA reserved variables.

GatherFacts.yml Line 7		
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



# 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*	Ad Setting
Auto-input	1:Gatherfacts	GatherFacts	1	Setting

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.

The screenshot shows a registration form for a target host. At the top, there are three dropdown menus labeled "Operation\*", "Movement\*", and "Host\*". Each menu has a value selected: "1:GatherFacts1" for Operation, "1:Gatherfacts" for Movement, and "1:targethost" for Host. These three fields are highlighted with a large red rectangular box. Below this, there is a summary table with three columns: "Operation", "Movement", and "Host". The values from the dropdown menus are listed here: GatherFacts1, GatherFacts, and targethost respectively. To the right of the table, there is an orange "Setting" button.

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.

**1.Basic info**

Fill out the following for each item

Item name  
Input method  
Maximum number of bytes

**3.Items**

**2.Target Menu group**

The screenshot shows the 'Menu creation information' dialog open. In the 'Basic information' tab, the 'Menu name\*' field is set to 'Gathered Facts'. The 'Creation target' dropdown is set to 'Parameter Sheet/Host/Ops'. The 'Display order\*' field is set to 1. Under 'Target menu group', 'Input\*' is selected. The 'Unique constraint(Multiple items)' section is collapsed. At the bottom left of the main window, there is a red box around 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	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]

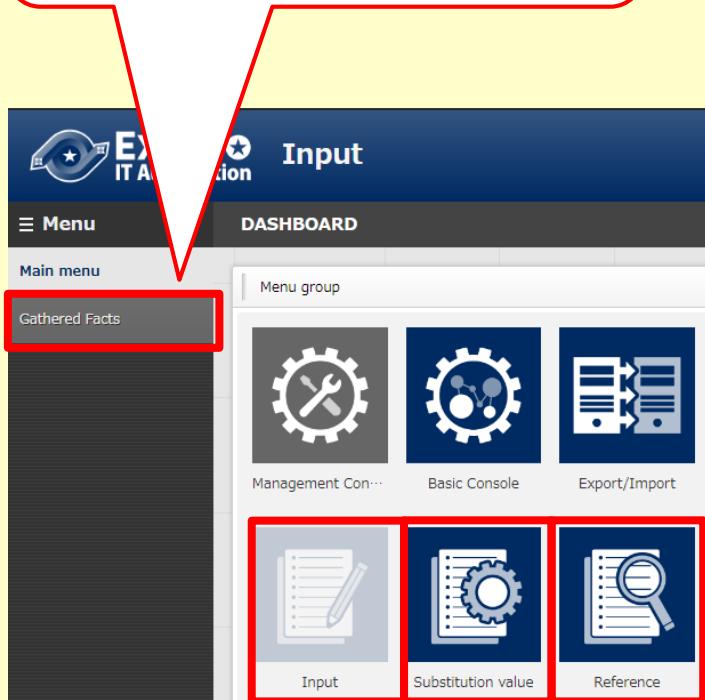
The screenshot shows a software interface for creating a parameter sheet. It features a grid of columns grouped into three main categories: 'address', 'interface', and 'network'. Each category has a dropdown menu set to 'String'. A red box highlights the header of the first column group, 'ansible\_default\_ipv4'. Below each group, there are sections for 'Maximum number of bytes\*', 'Regular expression', and checkboxes for 'Required' and 'Unique constraint'. Each group also contains 'Explanation' and 'Remark' fields.

# 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.

Register

No	Parameter		
	ansible_architecture	ansible_bios_version	ansible_default_ipv4
Auto-input			
4			

\*\*\* is a required item.

Parameter
ansible_distribution
ansible_distribution_file_path
ansible_distribution_file_variety
ansible_distribution_major_version

Parameter
ansible_machine
ansible_memtotal_mb
ansible_nodename
ansible_os_family
ansible_pkg_mgr
ansible_processor_cores

# 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**

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

Collected item (FROM)			Parameter sheet (TO)	
ID	Perth format*	PREFIX (file name)*	Variable name*	Member variables
1	YAML	gatherfacts	ansible_architectu	

The screenshot shows a registration interface for "Collected item (FROM)" and "Parameter sheet (TO)". The "Collected item (FROM)" section includes fields for ID (1), Perth format (YAML), PREFIX (file name) (gatherfacts), Variable name (ansible\_architectu), and Member variables. The "Parameter sheet (TO)" section includes fields for Menu group:Menu\* (2100011611:Substitution value:2:Gathered Facts) and Item\* (Parameter/ansible\_architecture). Red boxes highlight the "Perth format", "PREFIX (file name)", "Variable name", "Menu group:Menu\*", and "Item" fields.

# 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

# 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.

The screenshot shows the 'Collection interface information' configuration screen. At the top, there is a table with columns: History, Update, ID, hostname, IP, REST user, REST password, REST method, protocol, port, and Access permission. The 'Update' button is highlighted with a red box. Below this, a large blue arrow points down to a detailed view of the first row in the list. In the detailed view, the 'REST user' and 'REST password' fields are also highlighted with red boxes. At the bottom, there is a legend with two boxes: 'REST user' (User with execute permission) and 'REST password' (The password of the user).

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

<b>REST user</b>	<b>REST password</b>
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 shows the Ansible-Legacy interface with two main sections: Movement [List] and Operation [List].

**Movement [List]**

Select	Movement ID	Movement Name	Orchestrator	Delay timer	Dedicated information for ansible				Access permission	Remarks	Last update date/time	Last updated by
					Host specific format	WinRM connection	Header section	Optional parameter	Role to allow access			
<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	Remarks	Last update date/time	Last updated by
						Role to allow access			
<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**

**Operation ID: 1**  
**Operation Name: GatherFacts1**

**Movement [list]**      **Operation [list]**

GatherFacts	GatherFacts1
-------------	--------------

**Dry run**      **Execute**

# 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 Ansible-Legacy interface with the 'Check operation status' menu selected. The main window displays a table of operation details. The 'Status' row is highlighted with a red box, indicating it has been completed.

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

# 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

The diagram illustrates the relationship between the main execution list and a detailed view of a specific collection status. A dashed arrow points from the 'status' column in the main table to the 'status' column in the detailed view table, indicating that the status information shown in the main table corresponds to the detailed status shown here.

History	Execution No. $\downarrow$	Check execution status	Execution type $\downarrow$	Status $\downarrow$	execution engine $\downarrow$	Collection status	Collection log
History	2	Check execution status	Normal	Completed	Ansible Engine	Collected	<a href="#">CollectData_0000000002.log</a>

# 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	
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_default_ipv4 > address	ansible_default_ipv4 > interface	ansible_default_ipv4 > network	ansible_distribution	ansible_distribution_file_path
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]

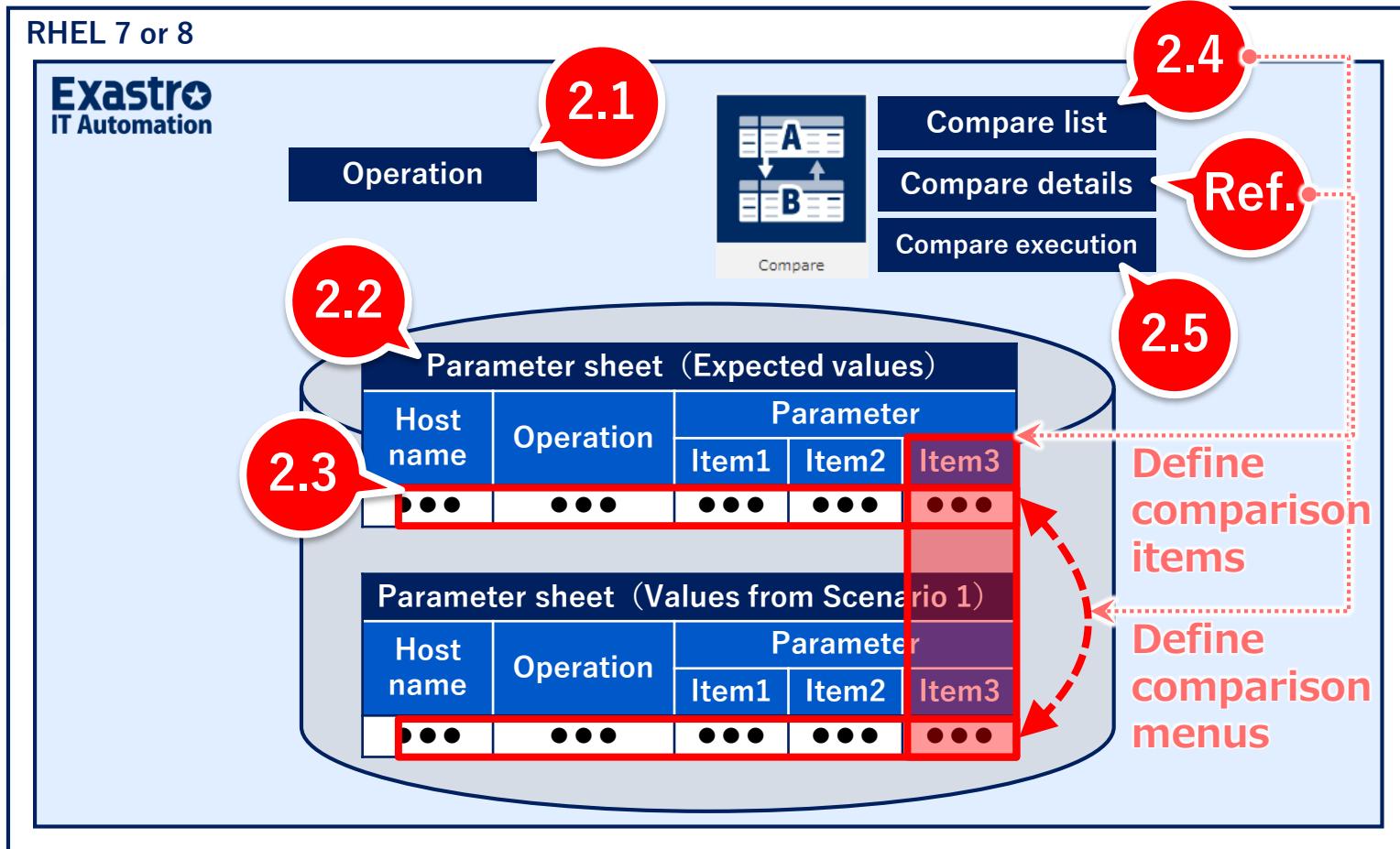
Compare the values and the expected values of the one collected in Scenario 1.

# 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
				<b>Setting</b> Role to allow access
Auto-input	Auto-input	GatherFacts2	2021/10/01 09:25	<b>Setting</b>

<b>Operation name (Free space)</b>	<b>Scheduled date for execution (Free space)</b>
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)

The screenshot shows the Exastro interface for creating a parameter sheet. On the left, there is a list of existing parameter sheets: 'ansible\_machine', 'ansible\_memtotal\_mb', and 'ansible\_nodename'. A new sheet is being created, with its details visible in the center and right panels.

**Basic Information:**

- Id:** Auto-input
- Menu name\***: OS information (highlighted with a red box and circled with a red '3')
- Creation target:** Parameter Sheet(Host/Operat (highlighted with a red box)
- Display order\***: 3 (highlighted with a red box and circled with a red '3')
- Create as hostgroup menu:** No
- Create as vertical menu**: No
- Last modified:** Auto-input
- Last updated by:** Auto-input

**Target menu group:**

- Input\***: Input
- Substitution value\***: Substitution value
- Reference\***: Reference

**Preview:** Shows a list view with columns: No, Host name, Operation, Operation name, Reference date and time, Scheduled date, Last run date. A red box highlights the 'Create' button at the bottom left of the preview area, and a red circle with the number '4' highlights the 'Host name' column header.

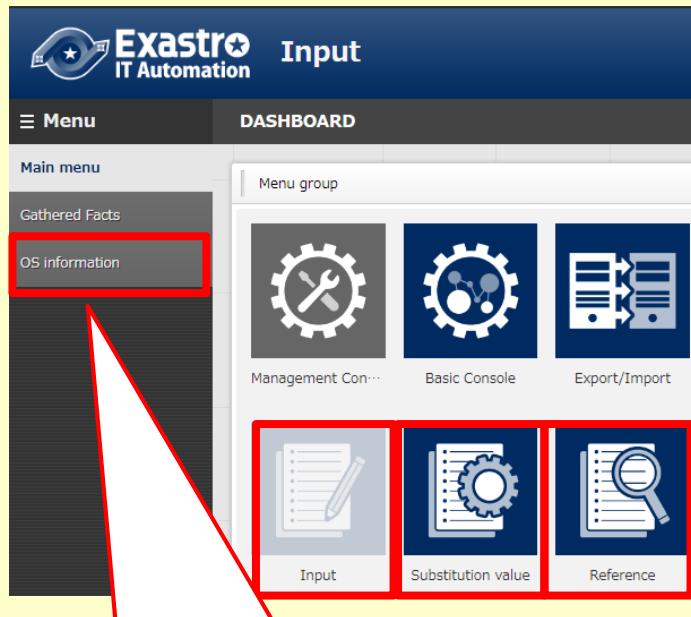
Menu name (Free space)	Display order
OS information	3

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.

The screenshot shows the 'Register' interface with three tabs:

- OS information:** Parameters: ansible\_architecture, ansible\_bios\_version, ansible\_default\_ipv4, address, interface, network. The first row (ansible\_architecture) is highlighted with a red box.
- Ansible distribution:** Parameters: ansible\_distribution, ansible\_distribution\_file\_path, ansible\_distribution\_file\_variety, ansible\_distribution\_major\_version. The first row (ansible\_distribution) is highlighted with a red box.
- Ansible machine:** Parameters: ansible\_machine, ansible\_memtotal\_mb, ansible\_nodename, ansible\_os\_family, ansible\_pkg\_mgr, ansible\_processor\_cores. The first row (ansible\_machine) is highlighted with a red box.

A red callout box points to the first row of the first tab with the text: "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.



The other items are shortened

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

Host name	Operation	Parameter/Item name		Other items
		ansible_default_ipv4_address		
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

The screenshot shows the 'Compare execution' interface. At the top, there is a 'Compare list' dropdown containing '1:OS info [ 5:OS information - 2:Gathered... ]'. Below it are two radio buttons for 'Output': 'ALL' (selected) and 'Difference Only'. A large orange 'Compare' button is at the bottom. The 'Output' section is highlighted with a red box.

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	192.0.2.2
2	Difference	targethost	Gathered Facts	1	GatherFacts1	2021/09/01 13:28	x86_64	1.11.0-2.el7	192.168.141.12

[Excel output](#)  
[CSV output](#)

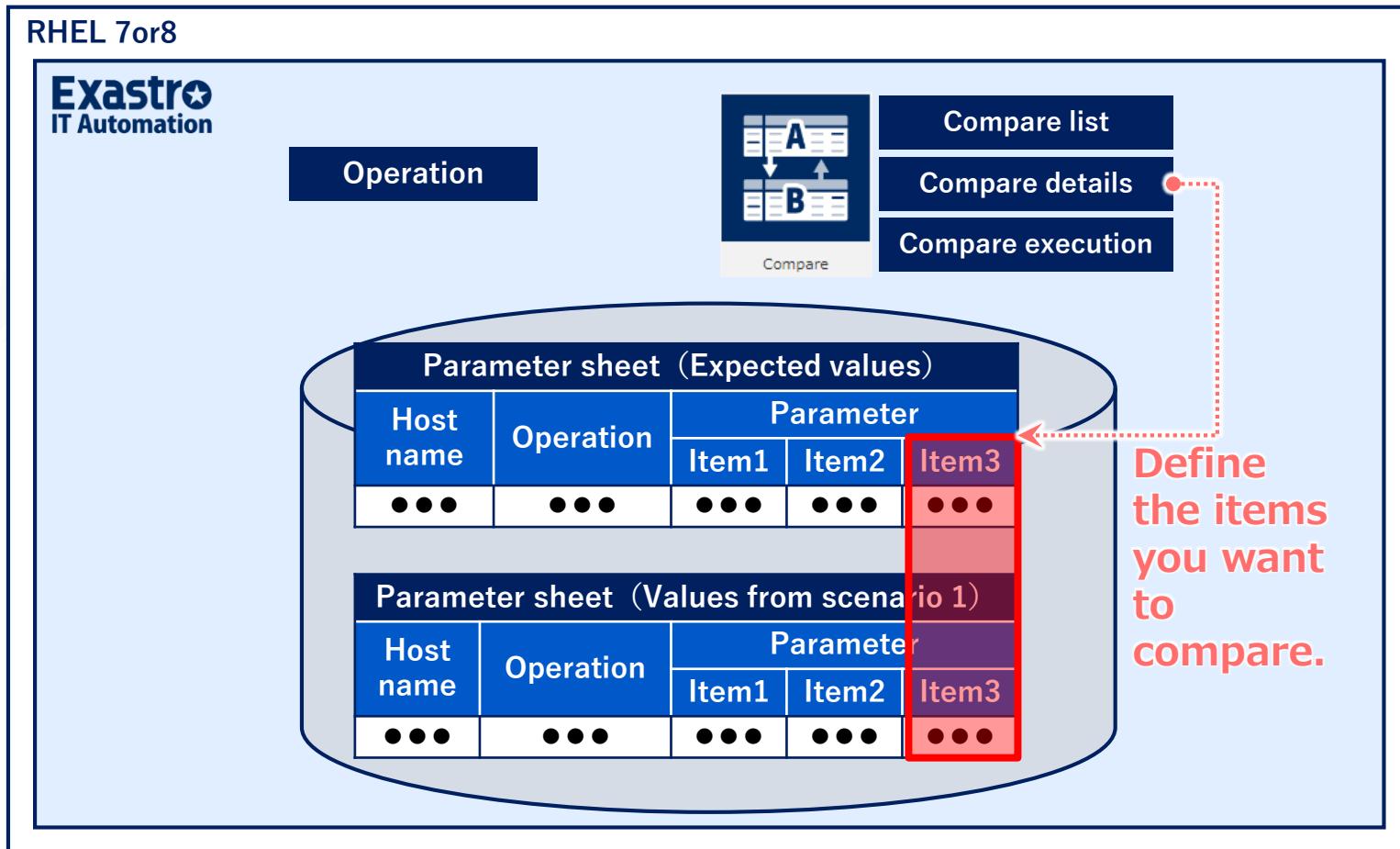
The address, which has different values, will be displayed in red.

# [Reference] Compare details

## 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	<input type="button" value="▼"/>

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 registration interface for 'Compare details'. It consists of two main sections: 'Target column 1' and 'Target column 2'. Each section has a 'Display item name\*' field where the user can type or select a name. In 'Target column 1', the display name 'IP address' is selected. In 'Target column 2', the display name 'Substitution value:2:Gathered Facts:3:Parameter/ansible\_default\_ipv4/address' is selected. Both sections also have dropdown menus for 'Compare name\*' and 'Target column'.

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

# [Reference] (3) Run comparison (1/2)

## 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]

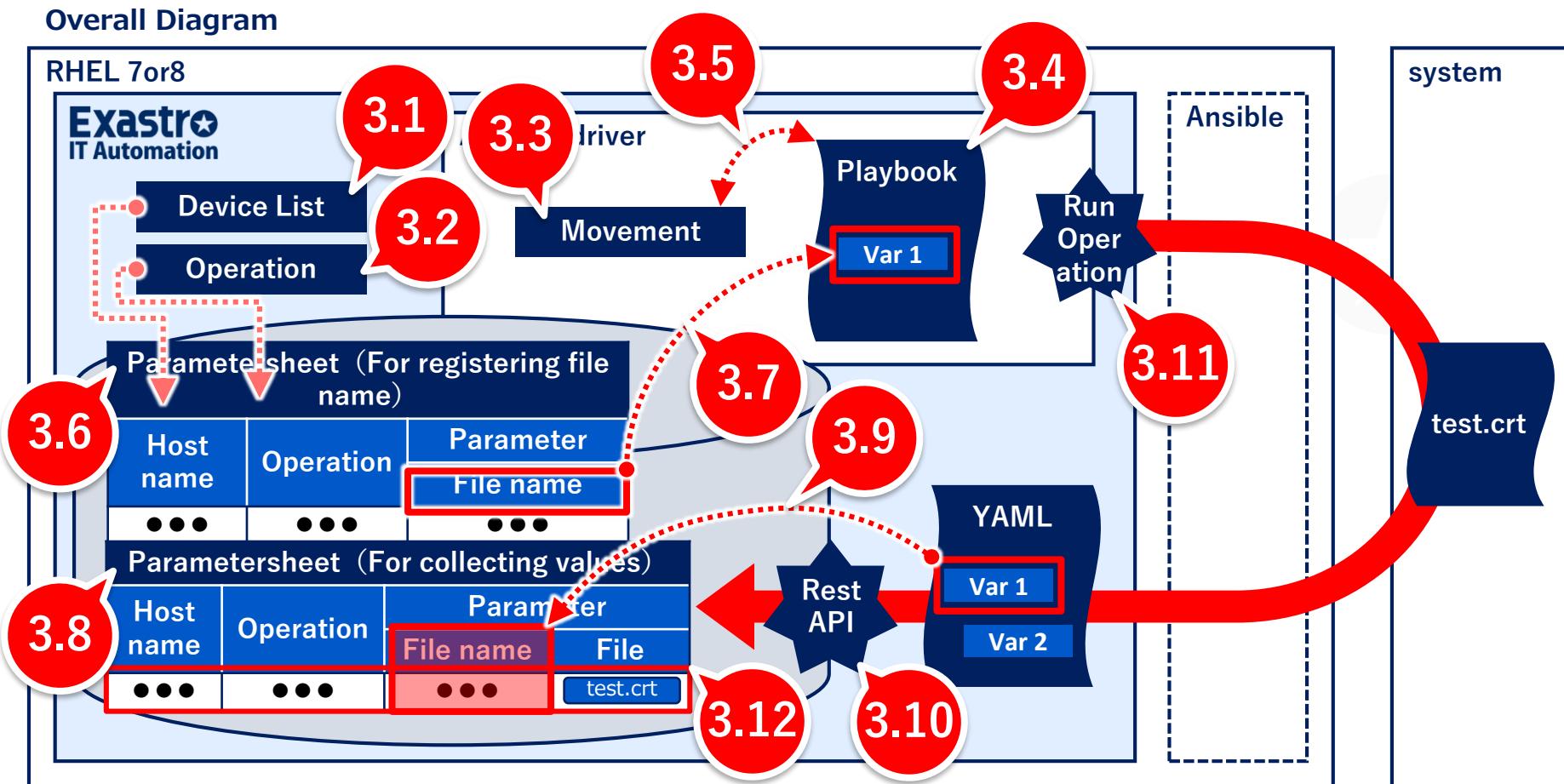
Collect the target host's SSL certificate file

# 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.

Managed system item number	HW device type	Host name	IP address	EtherWakeOnLan	MAC address	Network device name	Login user ID	Management	Login password	Ansible Dedicated information	Dedicated information for Legacy/Role	WinRM	Port no
Auto input	SV	targethost	192.0.2.1				root		*****				

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*	Action
3	3	getSSL1	2021/04/23 17:10	<b>Setting</b>

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

# 3.4 Register Playbook

## 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”](#)

```
- 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
```

**File name : getSSL.yml**

## 3.4 Register Playbook

### 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
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"/> Role to allow access

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

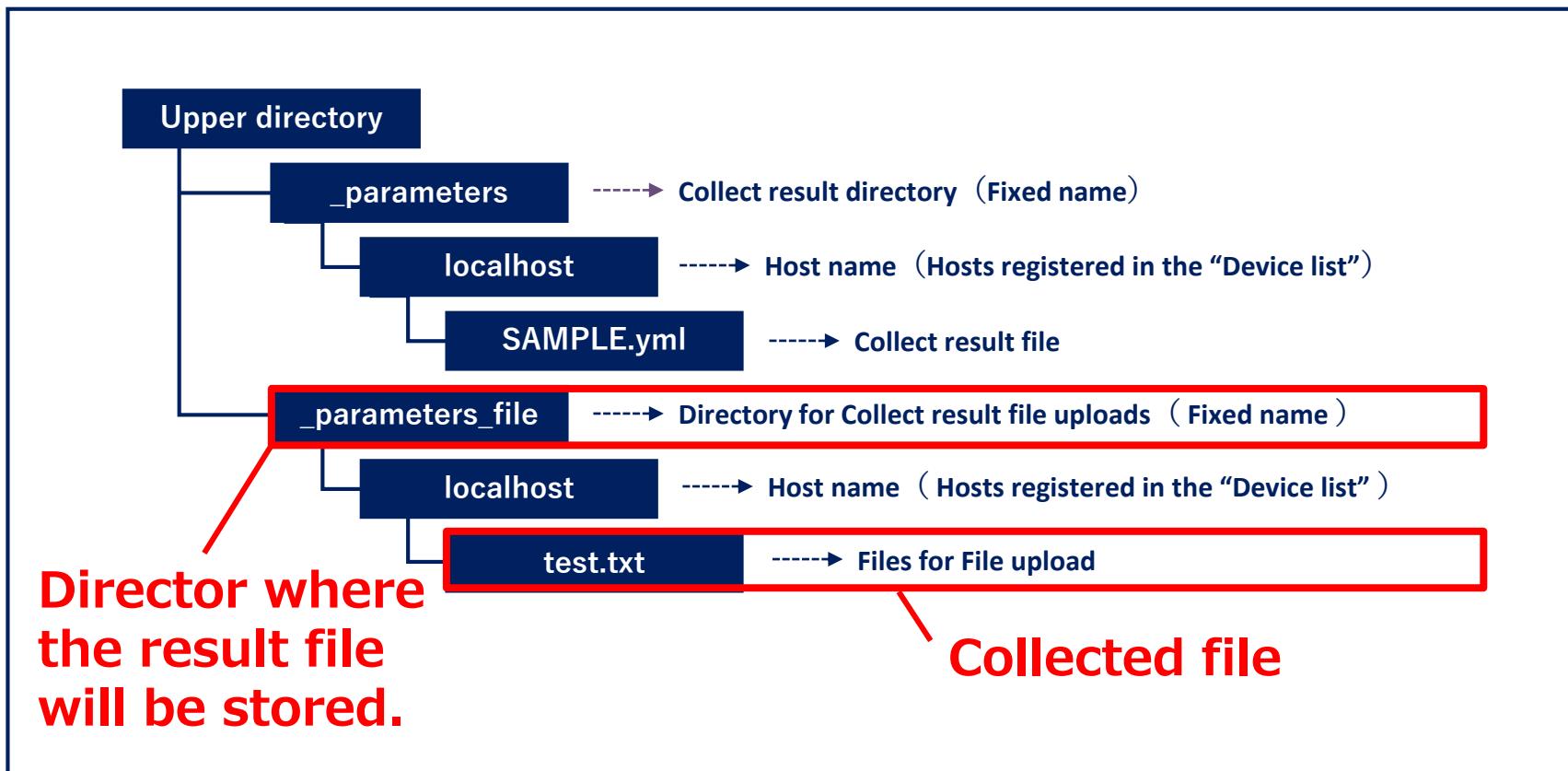
##### Path variables

	ITA reserved variable	Variable specified contents
Source file storage location	<code>__parameter_dir__</code>	“ <code>_parameters</code> ” path under the operation result directory
Collected file storage location	<code>__parameters_file_dir__</code>	“ <code>_parameters_file</code> ” 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*	Action Setting
Auto-input	2:getSSL	getSSL	1	Setting

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

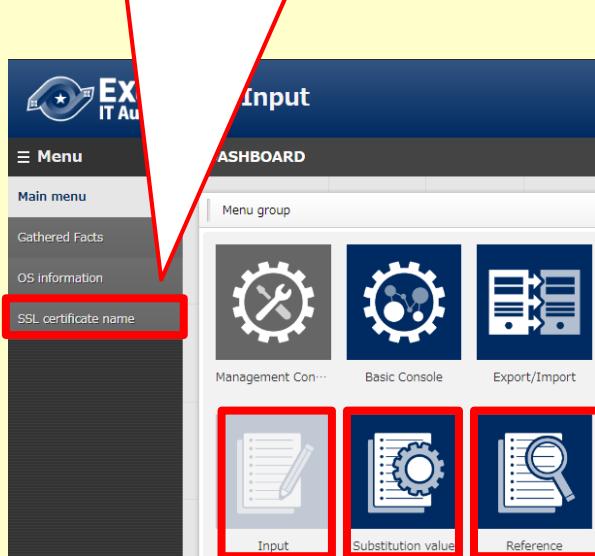
Item name (Free space)	Input method	Maximum number of bytes (Free value)
File name	String	128

### 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.

The screenshot shows a 'Register' dialog box. It has columns for 'No', 'Host name\*', 'Operation', and 'Parameter'. Under 'Parameter', there is a row for 'File name'. This 'File name' row is highlighted with a red box. The 'File name' input field contains the value 'Auto-input'.

No	Host name*	Operation	Parameter
		Operation*	File name
Auto-input			

### 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**

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

Parameter sheet(From)				IaC variable (To)	
Item No.	Menu group:Menu*	Item*	Registration method*	Movement*	Value variable
Auto-input	2100011611:Substitution value:8:SSL certificate name	Parameter/File name	Value type	2:getSSL	Variable name
				1:VAR_ssl_name	

**Parameter sheet (From)**

Menu group : Menu	Item	Registration method	Movement	Value variable
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

The screenshot shows a software interface for defining parameters. It displays two items side-by-side:

- File name:** Set to "String" with a maximum of 128 bytes.
- File:** Set to "File upload" with a maximum of 1,000,000 bytes.

Below each item are three red callout boxes with labels:

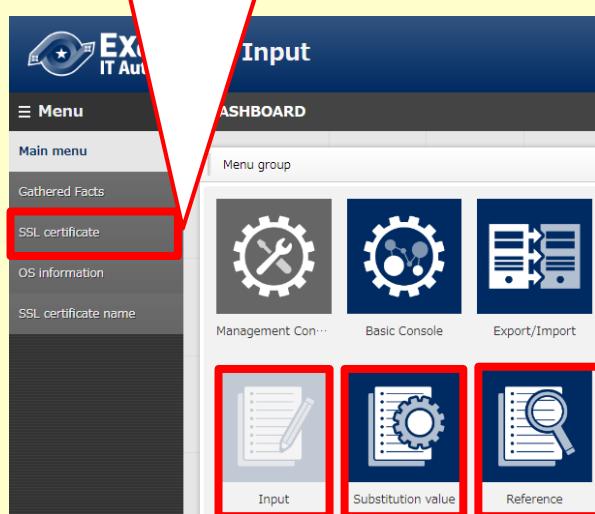
- Item name:** Points to the "File name" and "File" labels at the top of the interface.
- Input method:** Points to the dropdown menus for selecting "String" and "File upload".
- Maximum number of bytes:** Points to the respective byte counts (128 and 1,000,000) in the "Maximum number of bytes" fields.

### 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.

The screenshot shows a 'Register' dialog box. It has two main sections: 'Host name\*' and 'Operation\*'. In the 'Host name\*' section, 'Auto-input' is selected. In the 'Operation\*' section, there are dropdown menus for 'Host name\*' and 'Operation\*'. To the right, there's a 'Parameter' section with a 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:

A note at the bottom says: \* 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	Parse format*	PREFIX (file name)*	Variable name*	Member variables	Menu group:Menu*	Item*
Auto-input	Perth format*	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**

- Press the “Filter” button
- 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

A large blue arrow points from the 'Update' button in the list to the 'Edit' dialog for the first row.

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

**REST user**  
User with execute permission

**REST password**  
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 interface with two main windows open:

**Movement [List]**

Select	Movement ID	Movement Name	Orchestrator	Delay timer	Dedicated information for ansible			Access permis	Last update date/time	Last updated by
					Host specific format	WinRM connection	Header section	Optional parameter	Role to allow ac	
<input type="radio"/>	1	GatherFacts	Ansible Legacy		IP		- hosts: all remote_user: "{{ __loginuser__ }}" gather_facts: yes		2021/08/31 18:19:41	System Administrator
<input checked="" type="radio"/>	2	getSSL	Ansible Legacy		IP				2021/09/01 15:46:15	System Administrator

Filter result count: 2

**Operation [Filter]**

**Operation [List]**

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Access permission	Remarks	Last update date/time	Last updated by
						Role to allow access			
<input type="radio"/>	1	1	GatherFacts1	2021/04/22 17:09	2021/09/01 13:28			2021/09/01 13:28:17	Legacy execution procedure
<input type="radio"/>	2	2	GatherFacts2	2021/10/01 09:25				2021/09/01 13:50:51	System Administrator
<input checked="" type="radio"/>	3	3	getSSL1	2021/04/23 17:10				2021/09/01 15:40:58	System Administrator

Filter result count: 3

**Movement [List]      Operation [List]**

Movement ID: 2 Movement Name: getSSL	getSSL
Dry run	Execute

## 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**

Item	Value						
Execution No.	6						
Execution type	Normal						
Status	Completed						
execution engine	Ansible Engine						
Caller symphony							
Caller conductor							
Execution user	System Administrator						
Movement	<table border="1"><tr><td>ID</td><td>2</td></tr><tr><td>Name</td><td>getSSL</td></tr><tr><td>Delay timer (minutes)</td><td></td></tr></table>	ID	2	Name	getSSL	Delay timer (minutes)	
ID	2						
Name	getSSL						
Delay timer (minutes)							
Dedicated information for ansible	<table border="1"><tr><td>Host specific format</td><td>IP</td></tr><tr><td>WinRM connection</td><td></td></tr></table>	Host specific format	IP	WinRM connection			
Host specific format	IP						
WinRM connection							
Operation	<table border="1"><tr><td>No.</td><td>3</td></tr><tr><td>Name</td><td>getSSL1</td></tr><tr><td>ID</td><td>3</td></tr></table>	No.	3	Name	getSSL1	ID	3
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

The screenshot shows the Ansible-Legacy interface with the 'Execution list' page open. A 'Collection status' dialog box is displayed, connected by a dashed arrow from the 'Status' dropdown in the main table. The dialog box has two tabs: 'status' and 'Collection log'. The 'status' tab is selected, showing the value 'Collected'. The 'Collection log' tab is also visible. In the main table, the 'History' tab is selected, and the 'Status' column shows 'Completed' for the row with execution number 6. The 'Check execution status' button is highlighted in orange.

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

Collection status

status $\diamond$	Collection log
Collected	<a href="#">collectData_0000000006.log</a>

## 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	<a href="#">test.crt</a>

#### 4. Scenario 4 【Compare function】

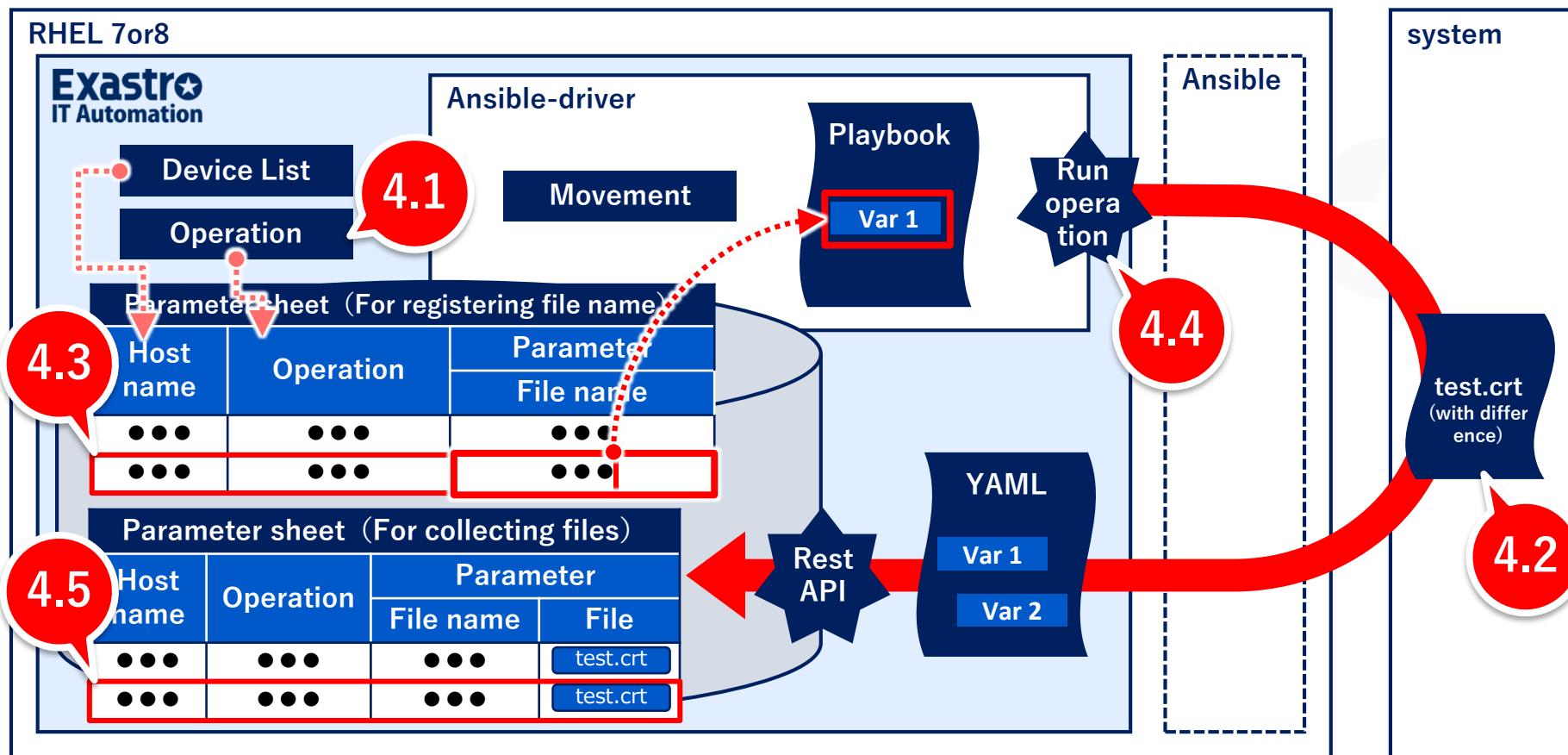
Compare the file downloaded in scenario 3 with the same file from a different date.

# 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.

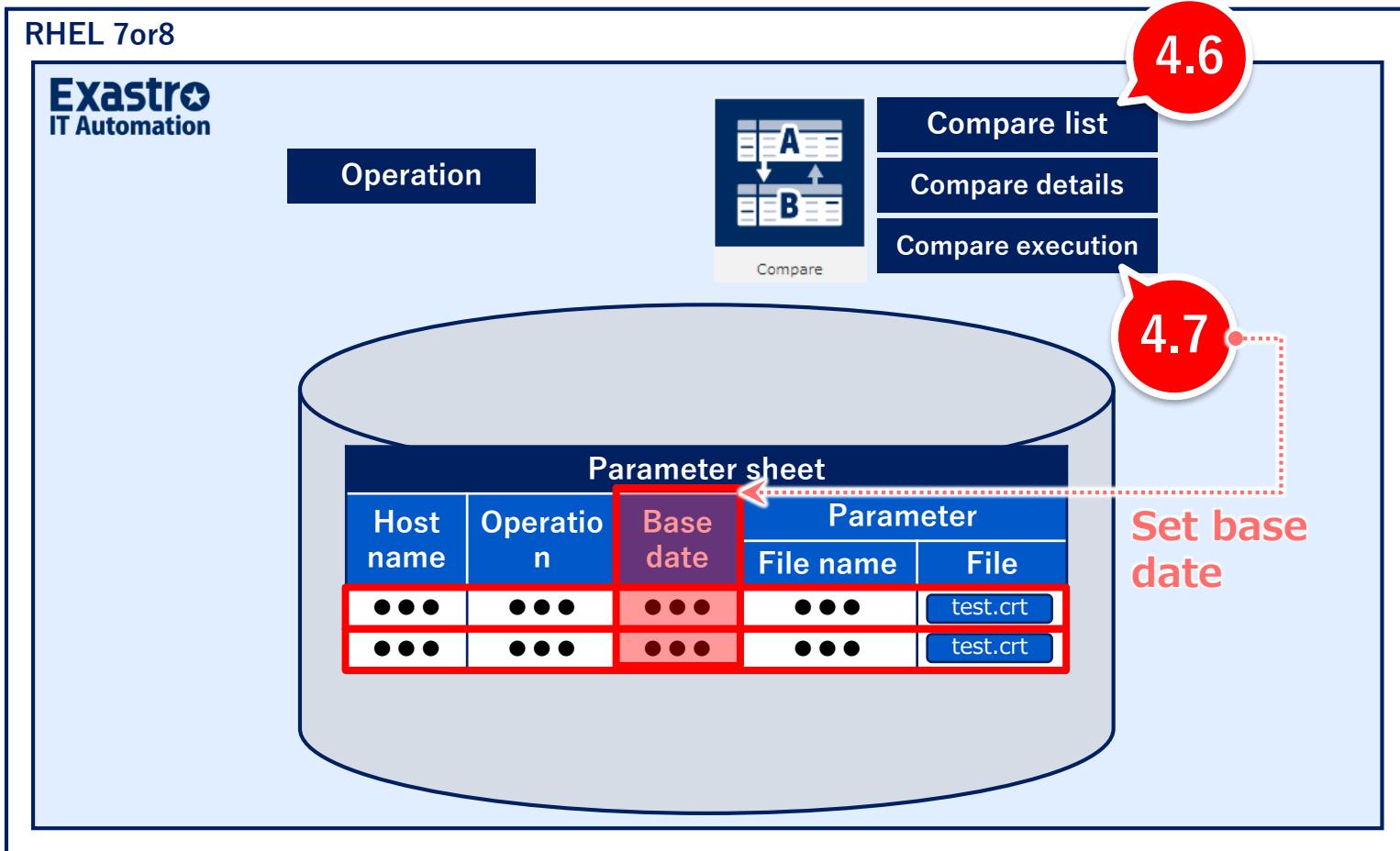
Overall diagram



# 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**

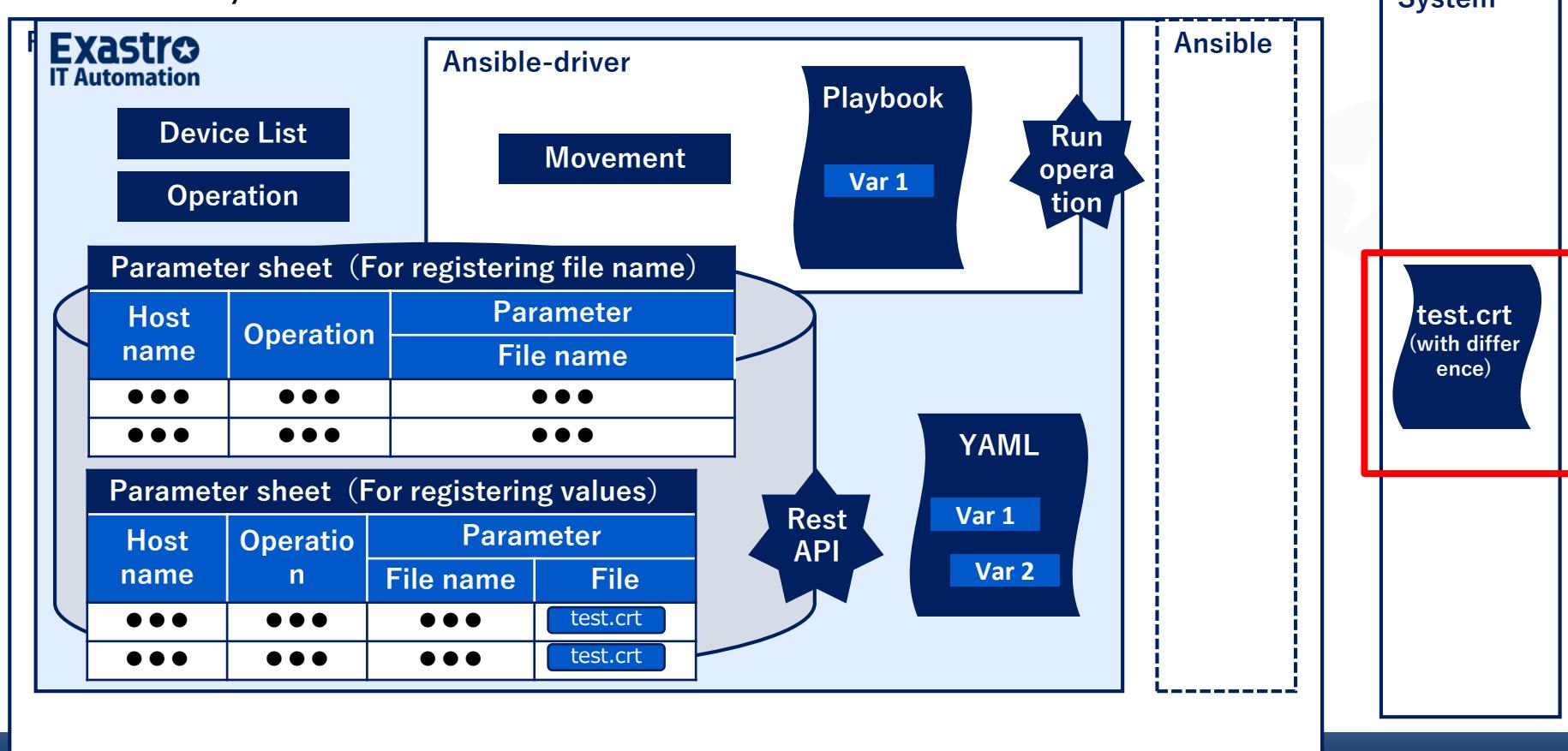
- ① Press “Register” -> “Start registration.”
- ② Input the following information and press the “Register” button.

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	<button>Setting</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)



## 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
	Host name*	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 interface with two main windows open:

- Movement [List]**: A table listing movements. The second row, "getSSL", is highlighted with a red border. The table columns include: Select, Movement ID, Movement Name, Orchestrator, Delay timer, Host specific format, WinRM connection, Header section, Optional parameter, Role to allow access, Remarks, Last update date/time, and Last updated by. The "getSSL" entry has the following details:
  - Host specific format: IP
  - WinRM connection: IP
  - Header section: - Hosts: all  
remote\_user: "{{ \_\_loginuser\_\_ }}"  
gather\_facts: yes
  - Role to allow access: System Administrator
  - Last update date/time: 2021/08/31 18:19:41
  - Last updated by: System Administrator
- Operation [Filter]**: A table listing operations. The fourth row, "getSSL2", is highlighted with a red border. The table columns include: Select, No., Operation ID, Operation name, Scheduled date for execution, Last execution date, Access permission, Remarks, Last update date/time, and Last updated by. The "getSSL2" entry has the following details:
  - No.: 4
  - Operation ID: 4
  - Operation name: getSSL2
  - Scheduled date for execution: 2021/04/28 12:19
  - Last execution date: 2021/09/02 09:38:47
  - Access permission: Legacy execution procedure
  - Remarks: System Administrator
  - Last update date/time: 2021/09/02 12:59:38
  - Last updated by: System Administrator

At the bottom of the interface, there is a navigation bar with two buttons:

- Movement [List]**: Contains the text "Movement ID: 2" and "Movement Name: getSSL".
- Operation [List]**: Contains the text "getSSL" and "getSSL2".

Below the Movement [List] button, there are two orange buttons: "Dry run" and "Execute". The "Execute" button is highlighted with a red border.

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

## 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="radio"/>

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="radio"/>

# 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

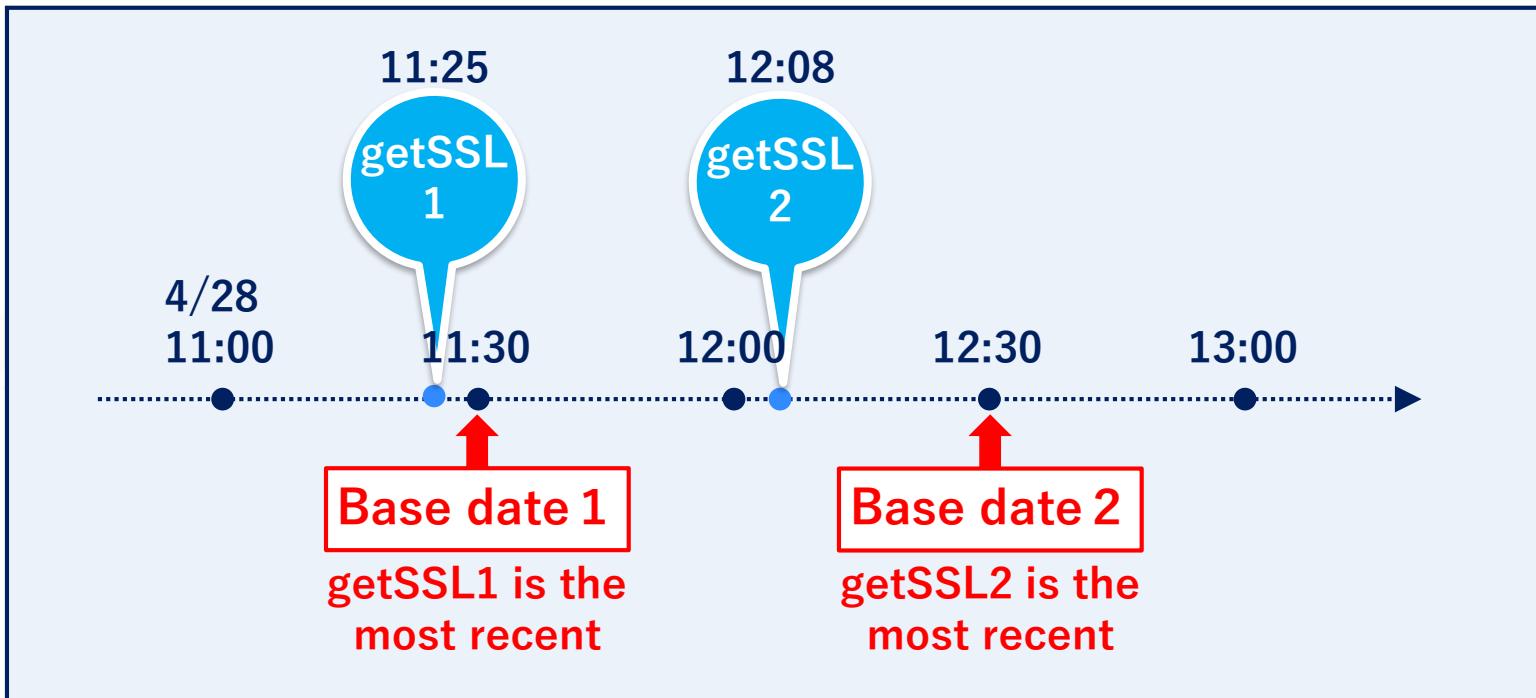
Output:  ALL  Difference Only

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