



IT Automation

Ansible Driver 【Tutorial】

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

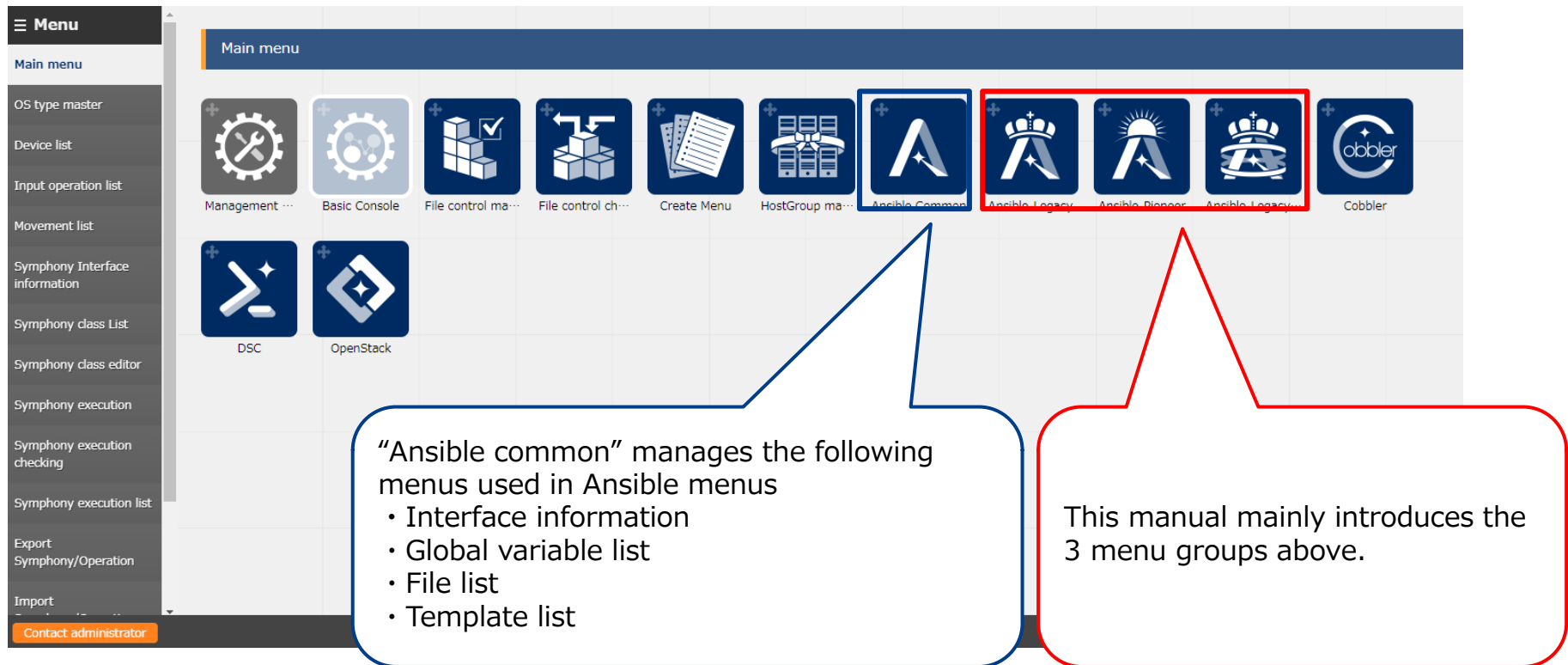
Table of contents

1. Introduction
2. About Ansible Driver
3. Associate with Ansible Tower
4. Description of the 3 modes
5. Features of each mode
 1. Legacy mode
 2. LegacyRole mode
 3. Pioneer mode

1. Introduction

Main menu

- This manual introduces the concept and function of **“Ansible-Legacy”**, **“Ansible-LegacyRole”**, **“Ansible-Pioneer”** in the main menu.
- Please refer to Practice manual (under construction) for the description of ITA screens.



The screenshot displays the ITA Main menu interface. On the left is a vertical sidebar menu with options: Menu, Main menu, OS type master, Device list, Input operation list, Movement list, Symphony Interface information, Symphony class List, Symphony class editor, Symphony execution, Symphony execution checking, Symphony execution list, Export Symphony/Operation, and Import. The main area is titled 'Main menu' and contains a grid of icons. A blue callout box points to the 'Ansible Common' icon, listing its managed menus: Interface information, Global variable list, File list, and Template list. A red callout box points to three icons: 'Ansible-Legacy', 'Ansible-Pioneer', and 'Ansible-Legacy', stating that the manual introduces these three menu groups. Other icons include Management, Basic Console, File control ma..., File control ch..., Create Menu, HostGroup ma..., DSC, OpenStack, and Cobbler.

“Ansible common” manages the following menus used in Ansible menus

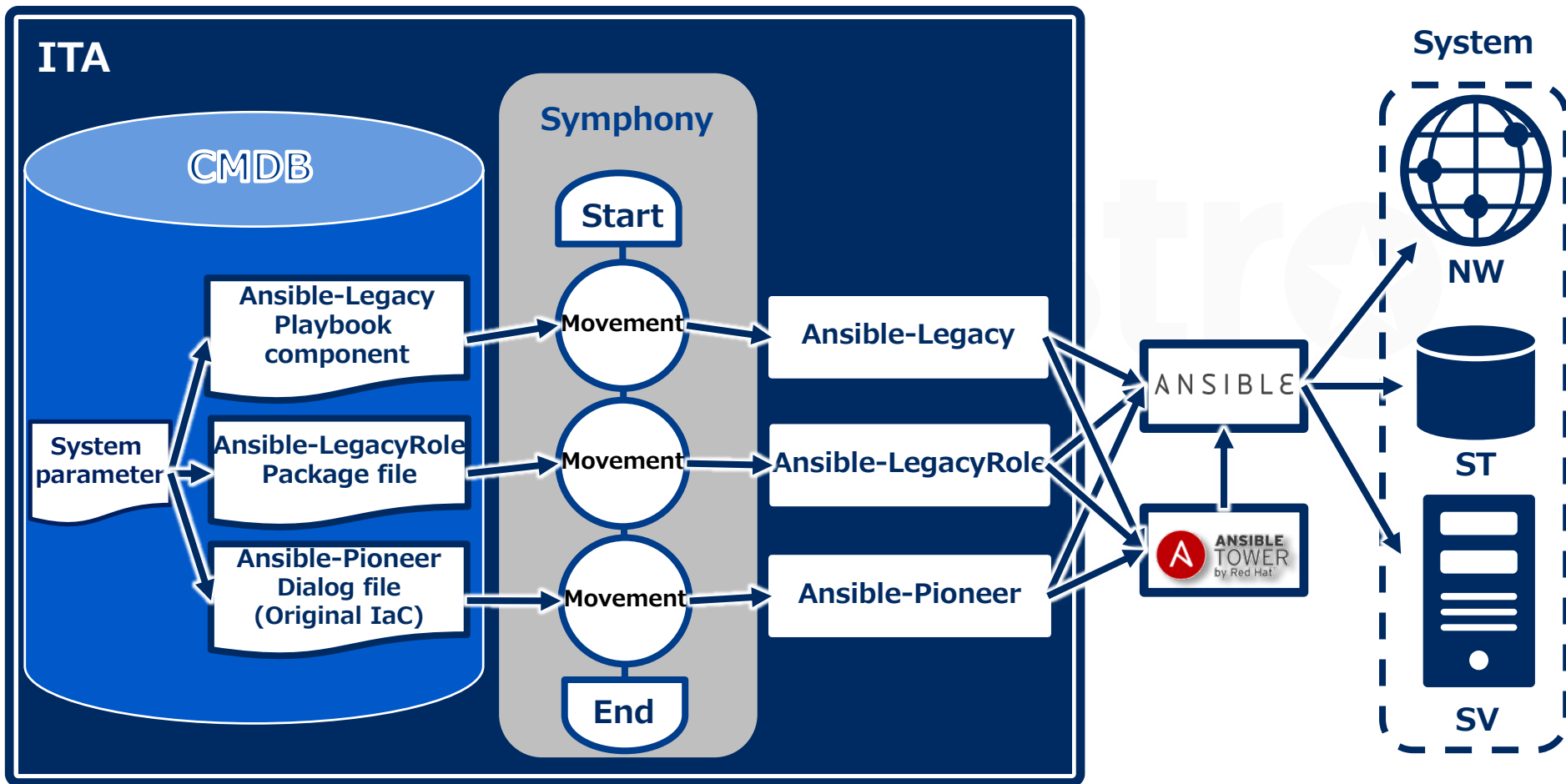
- Interface information
- Global variable list
- File list
- Template list

This manual mainly introduces the 3 menu groups above.

2. About Ansible Driver

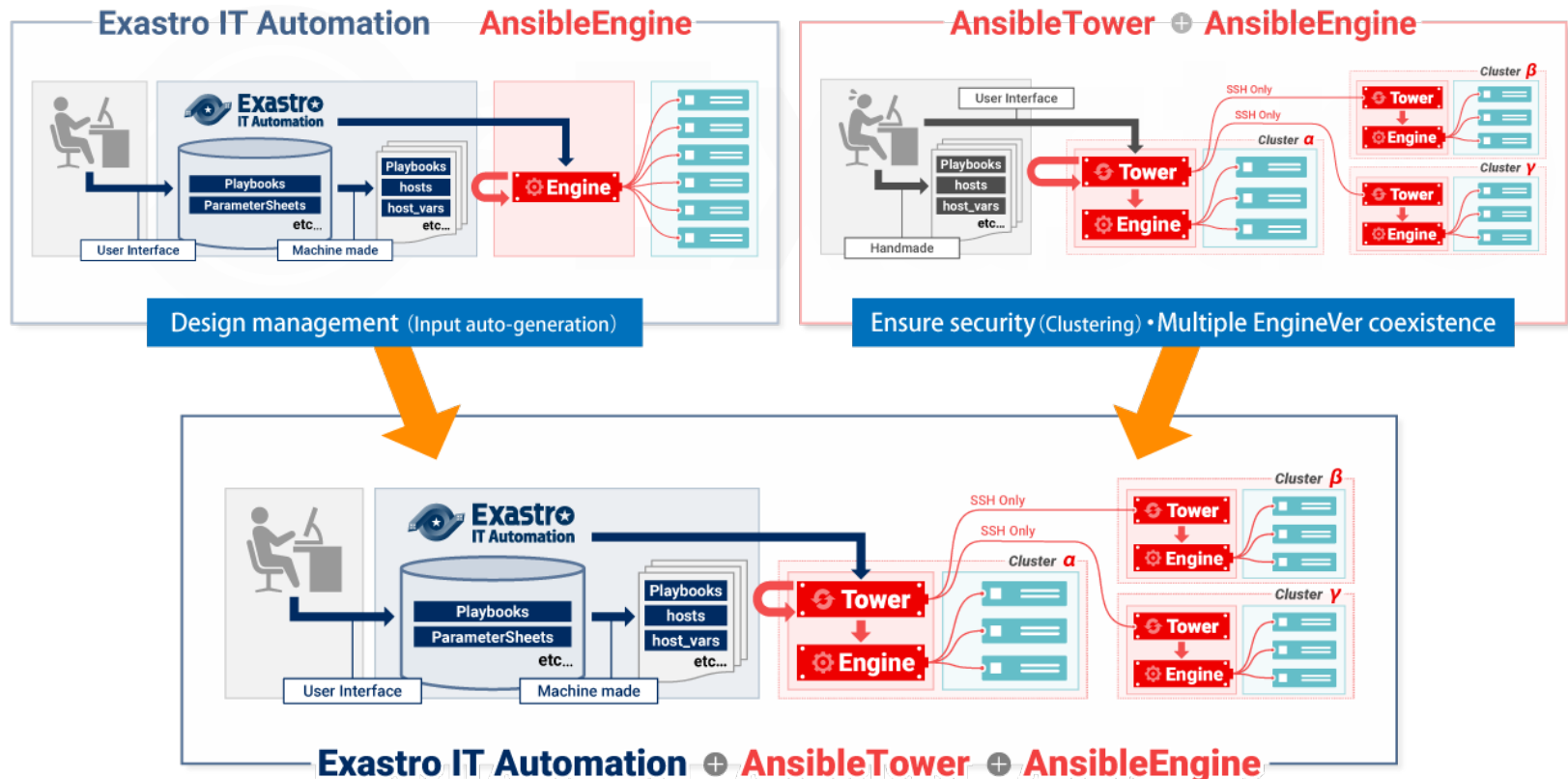
Ansible Driver links the parameters that are centrally managed by ITA with the variables of IaC(Playbook, etc.), which makes association with Ansible possible.

※For the advantages of using Ansible Tower, please refer to “3. Associate with Ansible tower”.



3. Associate with Ansible Tower

- IT Automation store/manages setting data and generates directories, configuration files that are required for Ansible to execute.
- AnsibleTower secures communication between clusters and controls different versions of AnsibleEngine
- The automated construction system consisting of IT Automation + AnsibleTower + AnsibleEngine, which combines each of their features, realizes efficiency and labor saving of work.



4. Description of the 3 modes

There are 3 modes with specific features of Ansible driver, depending on the usage.

- The following is comparison of the features of Ansible-Legacy, Ansible-LegacyRole, and Ansible-Pioneer

◆ IaC reuse

1. **Ansible-Legacy**
2. Ansible-Pioneer
3. Ansible-LegacyRole

◆ Know-how utilization

1. **Ansible-LegacyRole**
2. Ansible-Legacy
3. Ansible-Pioneer

◆ Application range

1. **Ansible-Pioneer**
2. Ansible-LegacyRole
3. Ansible-Legacy

※Legend

- ◎ : Has strength in such function
- : Can be used for such function
- × : Difficult to apply to such function/Recommend to use other mode

	Description	Ansible-Legacy	Ansible-LegacyRole	Ansible-Pioneer
IaC reuse	Modulize created Playbooks and reuse them in Exastro.	◎	×	○
Know-how utilization	Able to utilize variable functions provided by Ansible, and able to utilize published PlaybookRoles such as Ansible-galaxy.	○	◎	×
Application Range	The variation of the operation procedure that can be automated.	○	○	◎

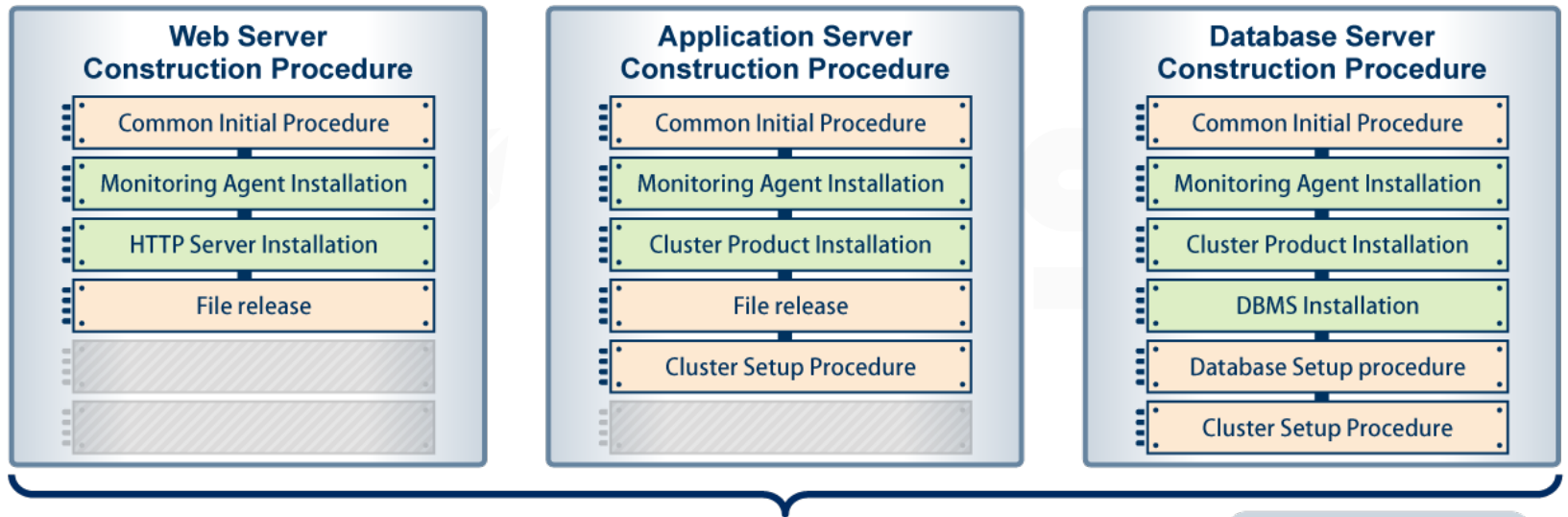
5. Features of each mode

5.1 Ansible-Legacy mode

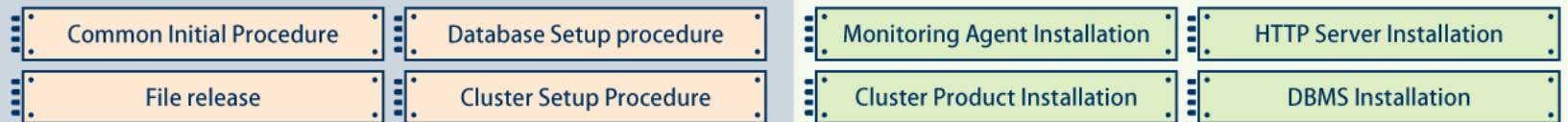
5.1 Ansible-Legacy mode

The spice that based on ITA - Ansible-Legacy mode

- The main feature of this mode is reusing IaC by modularizing them.
- By reusing the registered IaC, it is possible to construct system efficiently.

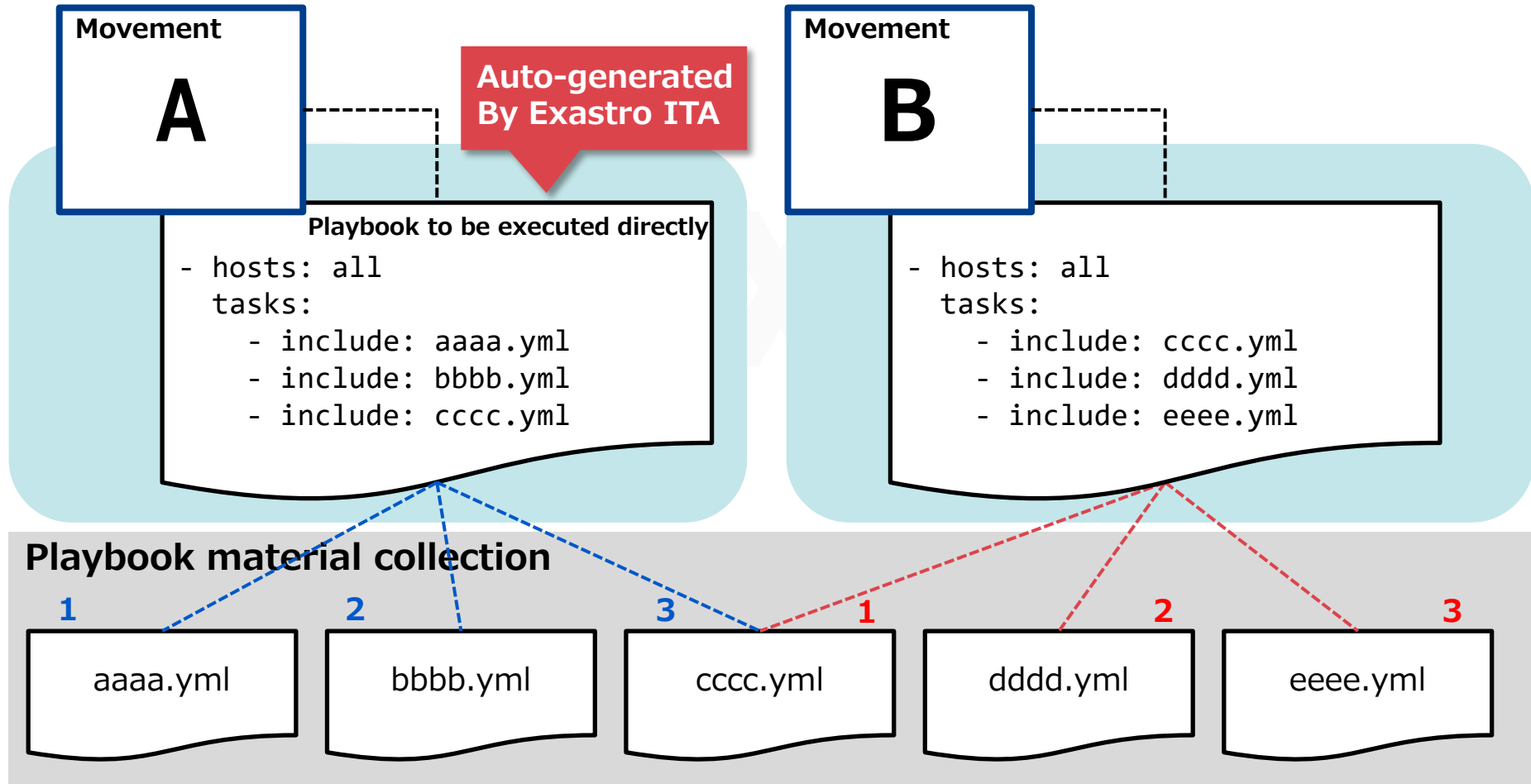


Manage common procedures so they can be modularized and reused



5.1 Ansible-Legacy mode

- The relationship between "Movement", a work execution unit in Exastro ITA, and playbook is specified in two layers.



5.1 Ansible-Legacy mode

Parameters that are given to variables during operation execution can be managed in the parameter sheets of ITA.

Parameter sheet

hosts	OPERATO N_ID	para1	para2	para3	para4	...
host_a	888	AAAA	1	100	1	
host_b	111	BBBB	2	78	0	
host_c	888	CCCC	3	93	0	
⋮						

Substitution value list

Movement	Parameter	Variable
A	para1	VAR_1
	para2	VAR_2
⋮

Parameter sheet and variables are linked and set in playbook

Movement

A

Playbook to be executed directly

- hosts: all
- tasks:
 - include: aaaa.yml
 - include: bbbb.yml
 - include: cccc.yml

1

aaaa.yml

- name: "set aaaa"
- debug: msg= "{{ VAR_4 }}"
- command: sleep {{ VAR_2 }}

2

bbbb.yml

- name: "set bbbb"
- debug: msg= "{{ VAR_5 }}"
- command: sleep {{ VAR_3 }}

3

cccc.yml

- name: "set cccc"
- debug: msg= "{{ VAR_1 }}"
- command: sleep {{ VAR_2 }}

5.1 Ansible-Legacy mode

- Users **don't need to be aware of** the behavior of ITA during operation, the following is only for reference of background behavior.

Parameter sheet

hosts	OPERATO N_ID	para1	para2	para3	para4	...
host_a	888	AAAA	1	100	1	
host_b	111	BBBB	2	78	0	
host_c	888	CCCC	3	93	0	
host_d	222	DDDD	10	78	0	
host_e	333	EEEE	5	84	1	
host_f	888	FFFF	4	80	0	
host_g	111	GGGG	3	90	1	
:						

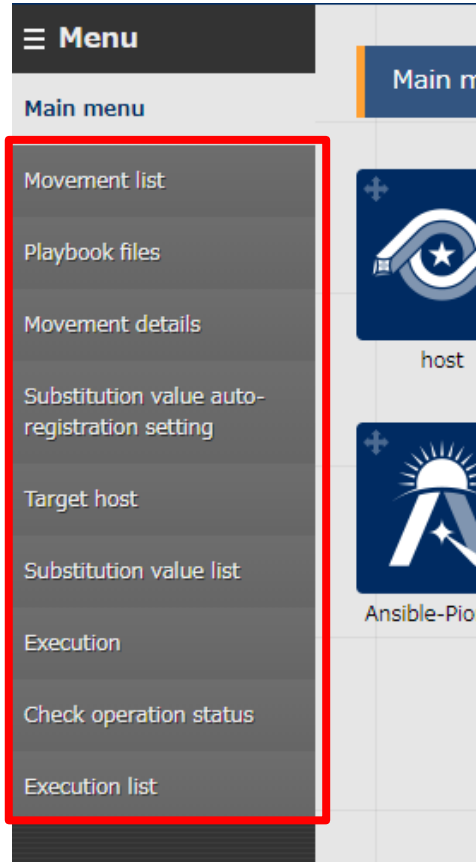
Operation execution directory

```
in
├── hosts
├── playbook.yml
├── child_playbooks
│   ├── aaaa.yml
│   ├── bbbb.yml
│   └── cccc.yml
├── host_vars
│   ├── host_a
│   │   └── main.yml
│   ├── host_c
│   │   └── main.yml
│   └── host_f
│       └── main.yml
└── out
```

hosts : The operation target hosts (OPERATION_ID=888)
playbook.yml : The Playbook to be executed directly.
child_playbooks : Stores the Playbook materials to be used for this operation.
host_vars : Stores the Playbooks with different variable definition for each host.
※Please refer to Learn: BASE for details of OPERATION_ID.

5.1 Ansible-Legacy mode

Menu function description



- **Movement list**
Create or check the list of Movement
- **Playbook files**
Create or check the list of IaC
- **Movement details**
Manage the playbooks to be included in Movement
- **Substitution value list**
Link registered Operation with the setting value of every item for each host.
Manage Movement and variable.
- **Target host**
Manage the Movement and host to be linked with Operation
- **Substitution value list**
Manage playbook and the substitution value of the "VAR_" variable used in Movements
- **Execution**
Execute single created Movement
- **Check operation status**
Check the details of the executed Movement.
- **Execution list**
Check the execution detail and history of created and executed Movements.

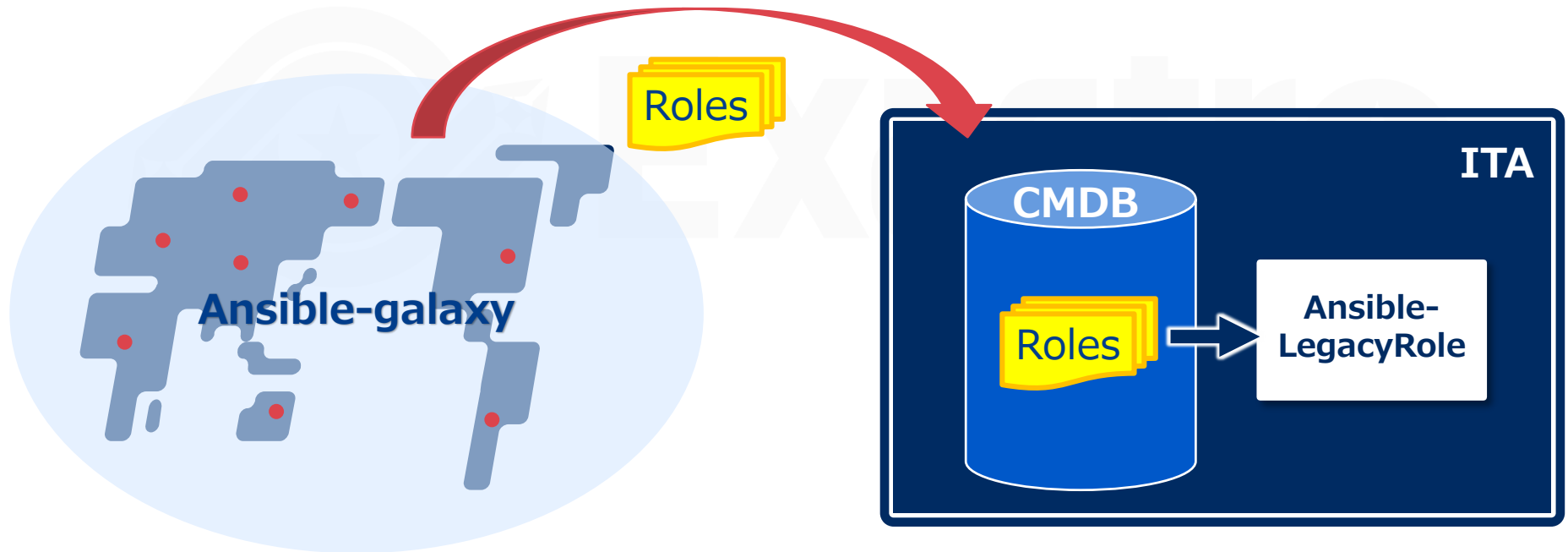
5. Features of each mode

5.2 Ansible-LegacyRole mode

5.2 Ansible-LegacyRole mode

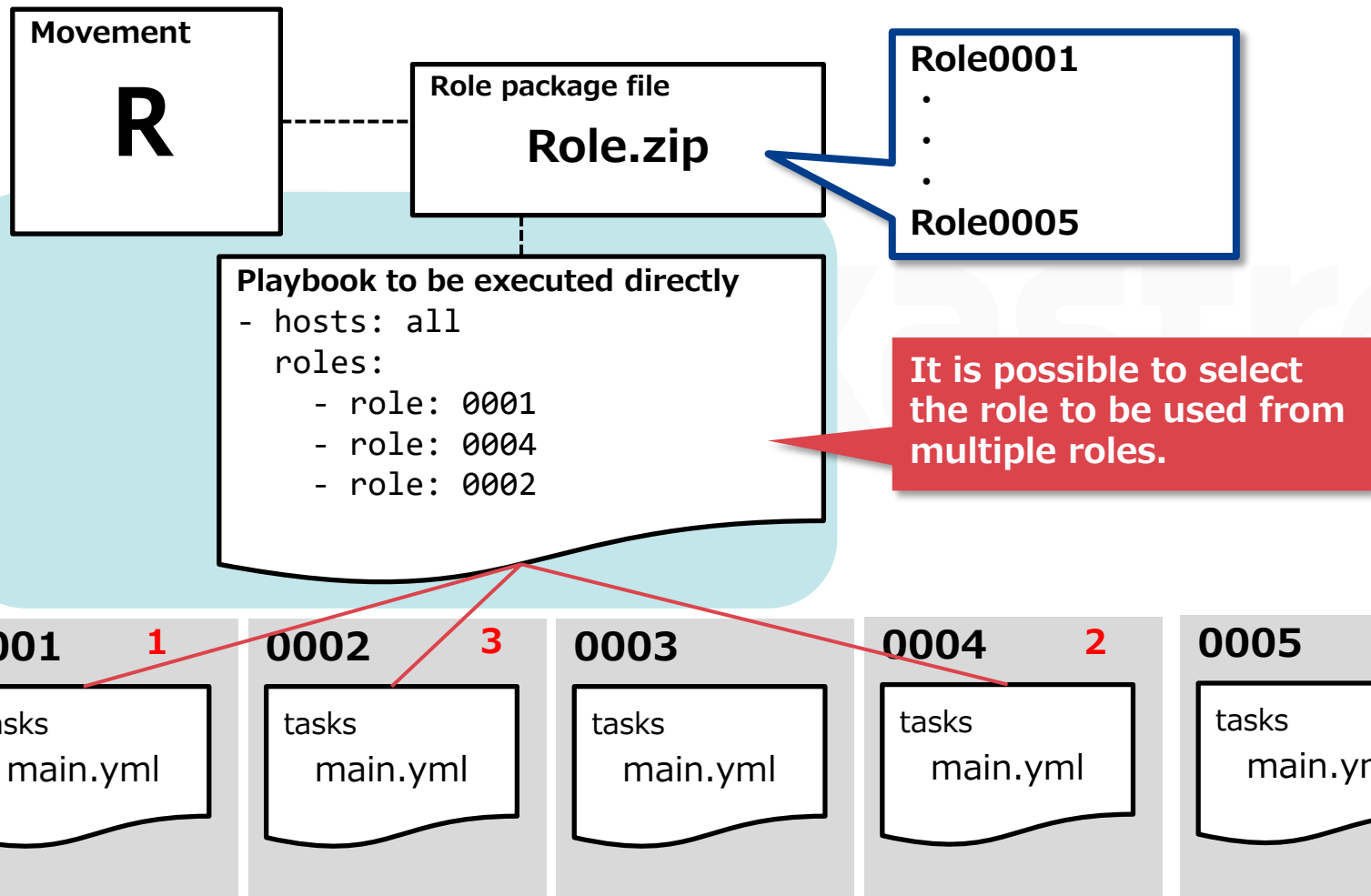
Gather the world's wisdom in your hand - Ansible-LegacyRole mode

- The feature of Ansible-Legacy mode is to register and use role package.
- Users can use self-created Role or obtain and use Roles from Ansible-galaxy



5.2 Ansible-LegacyRole mode

- Link “Movement”, the operation execution unit in Exastro ITA, with roles in role package.



5.2 Ansible-LegacyRole mode

- Users **don't need to be aware of** the behavior of ITA during operation, the following is only for reference of background behavior.

Operation execution directory

```
in
├── hosts
├── site.yml
├── host_vars
│   ├── host_a
│   └── host_b
├── roles
│   ├── role①
│   │   ├── defaults
│   │   │   └── main.yml
│   │   ├── tasks
│   │   │   └── main.yml
│   │   └── template
│   │       └── xxxx.yml
│   ├── role②
│   │   .
│   │   .
│   │   .
│   └── .
└── out
```

Role package file is a zip file of a directory that contains roles folder

Execution role directory is described in site.yml (playbook to be executed)

hosts	: Operation target hosts (Created by ITA)
site.yml	: Playbook to be executed directly(Created by ITA)
host_vars	: Stores Playbooks that defines variable for each host (Created by ITA)
roles	: Directory that stores files for playbook execution of each role

⇒Files under role directory

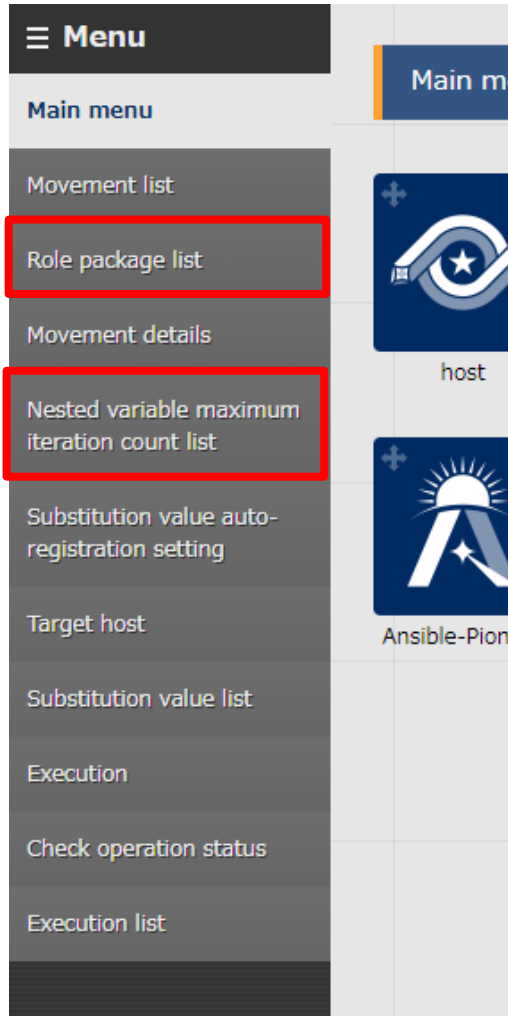
- defaults : Describes the parameter for the part to be substituted in playbook
- tasks : Playbook to be executed
- template : Text file used in the playbook to be executed

※The figure of directory on the left is only an example.

5.2 Ansible-LegacyRole mode

Menu function description

(Difference between Ansible-Legacy mode)



- **Role package list**

Manage created role package file.

- **Nested variable maximum iteration count list**

Manage the maximum iteration count of the nested variable array if there is nested array in role package file.

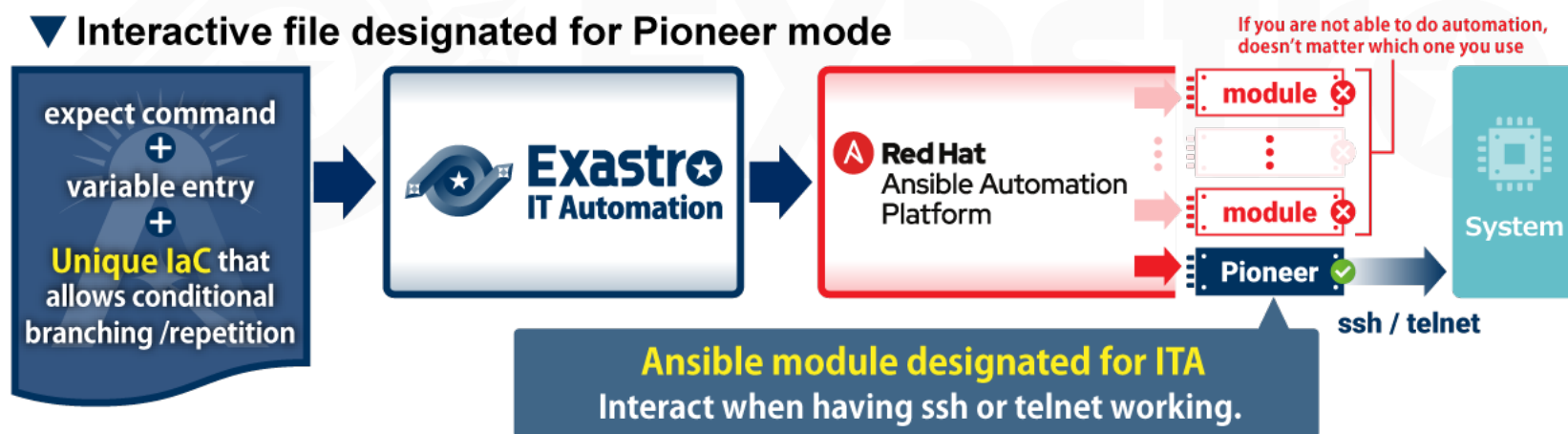
5. Features of each mode

5.3 Ansible-Pioneer Mode

5.3 Ansible-Pioneer Mode

The last way to continue automation - Ansible-Pioneer mode

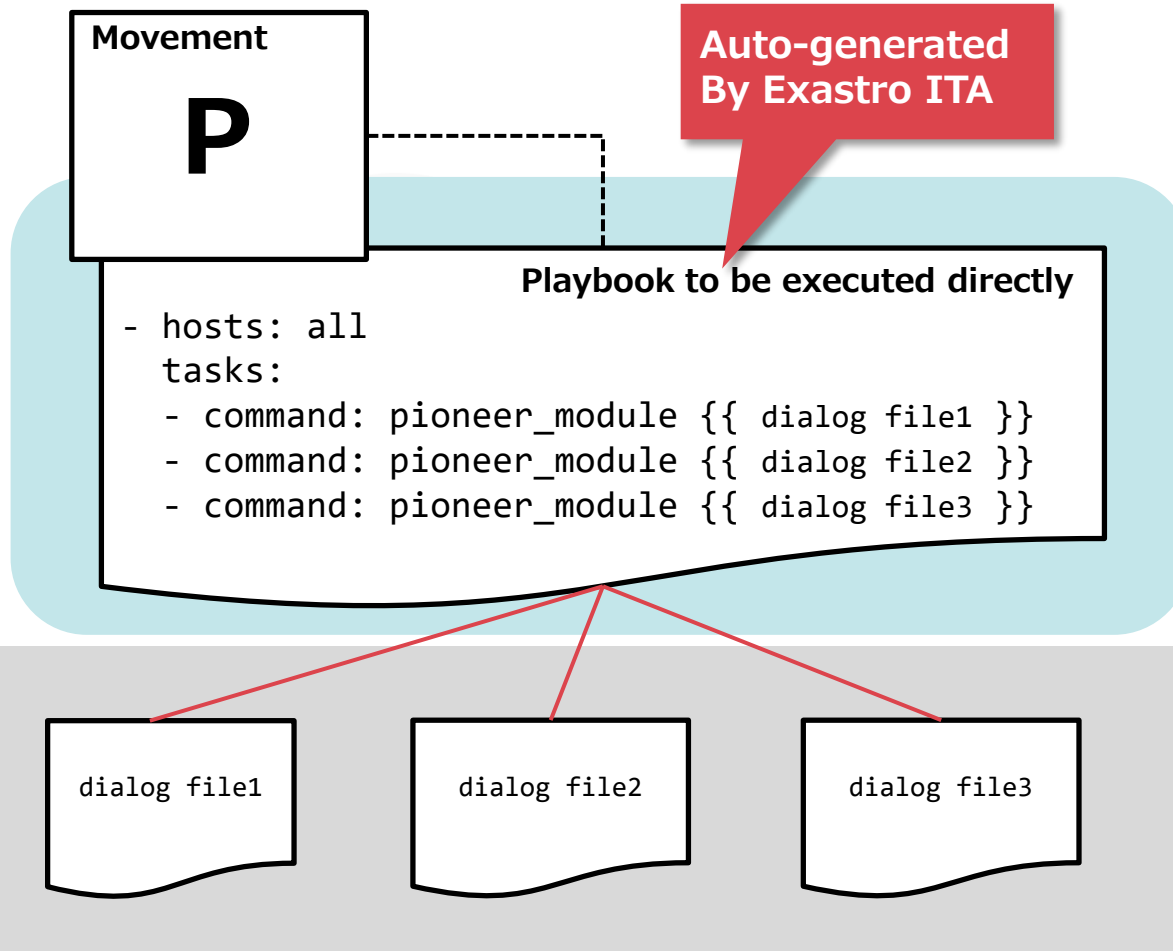
- If users can't automate operation by using any of the Ansible drivers, then the benefits of automation will be reduced. Therefore ITA offers Pioneer mode which is the last way to prevent automation from being stopped.



5.3 Ansible-Pioneer mode

Ansible-Pioneer executes dialog files(※) in order by using Pioneer module (ITA original module) from the playbook to be executed directly.

※The details of dialog file is described in the next slide.



5.3 Ansible-Pioneer mode

- In Ansible-Pioneer, it is possible to describe settings for target hosts in interactive(dialog) style. Furthermore, it is also possible to express advanced dialog such as using iterative and conditional branching compared to a simple expect command.

※Please refer to the [manual](#) for the details of dialog file.

Example of Dialog file

1. Login target system and check the status of the services assigned by variable.
2. If the status is "disable", the process will end as error. If the status is other than "disable" , then "complete!" will be displayed in prompt.

※The words in **red highlight** in the dialog file are displayed using the variable in parameter sheet.

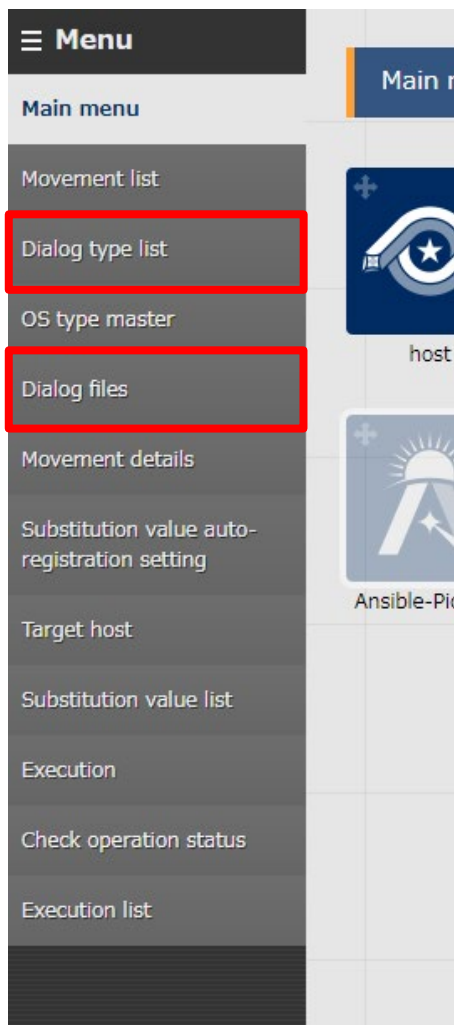
```
01 - expect: 'password:'
02   exec: "{{ login password }}"
03 - command: 'systemctl status {{ item.0 }}'
04 ① prompt: '{{ login user }}@'
05   with_items:
06     - '{{ service name 1 }}'
07     - '{{ service name 2 }}'
08   failed_when:
09 ② - stdout match(disable)
10   - command: 'echo complete!'
```

• Repeat process by using "with_items"

• Conditional branching by using "failed_when"

5.3 Ansible-Pioneer mode

Menu function description (Difference between Ansible-Legacy mode)



- **Dialog type list**

Manage dialog type.

(In Ansible-Pioneer, define different dialog file for each OS, so the difference for each OS will be automatically absorbed.)

- **Dialog files**

Manage dialog files for each os type.



Exastro