Exastro

ITA_User_Manual

Collect function

―Ver 1.10―

Disclaimer

Trademark

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

Table of contents

Introduction

This document explains the ITA Collect function and how to use it.

1 Collect function overview

This section explains the collect function.

1.1 About the collect function

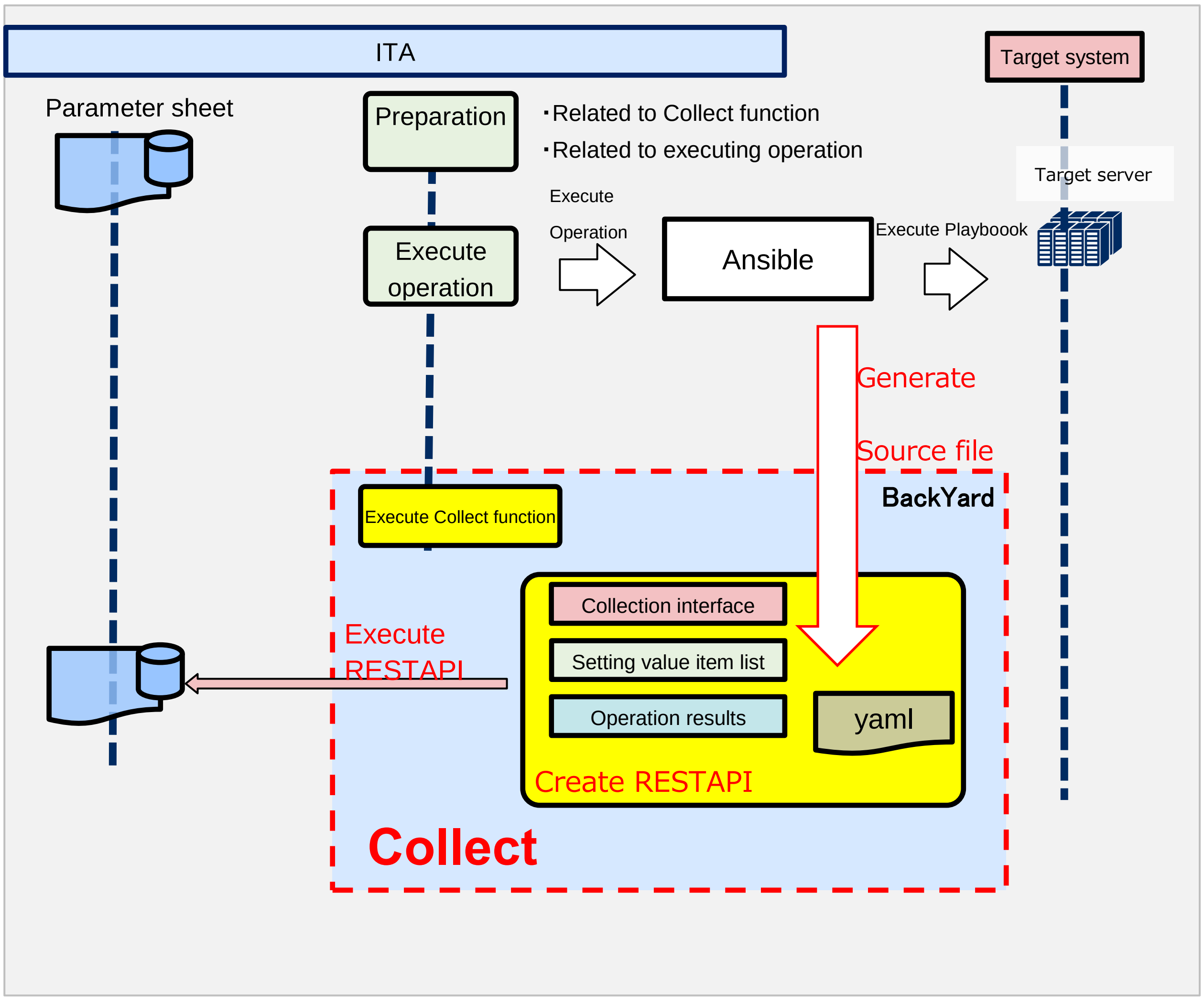
The collect function automatically registers values to parameter sheets. The values are based on the results of executed operations (source files output in a specified format) in ITA.

This function uses Ansible-Driver as target.

For more information about Ansible, please refer to the Ansible product manual
For more information about Ansible-Driver, please refer to “Exastro-ITA_User_Instruction_Manual_Ansible-Driver”
For more information about Parameter sheets, please refer to “Exastro-ITA_User_Instruction_Manual_Menu_creation_function”.

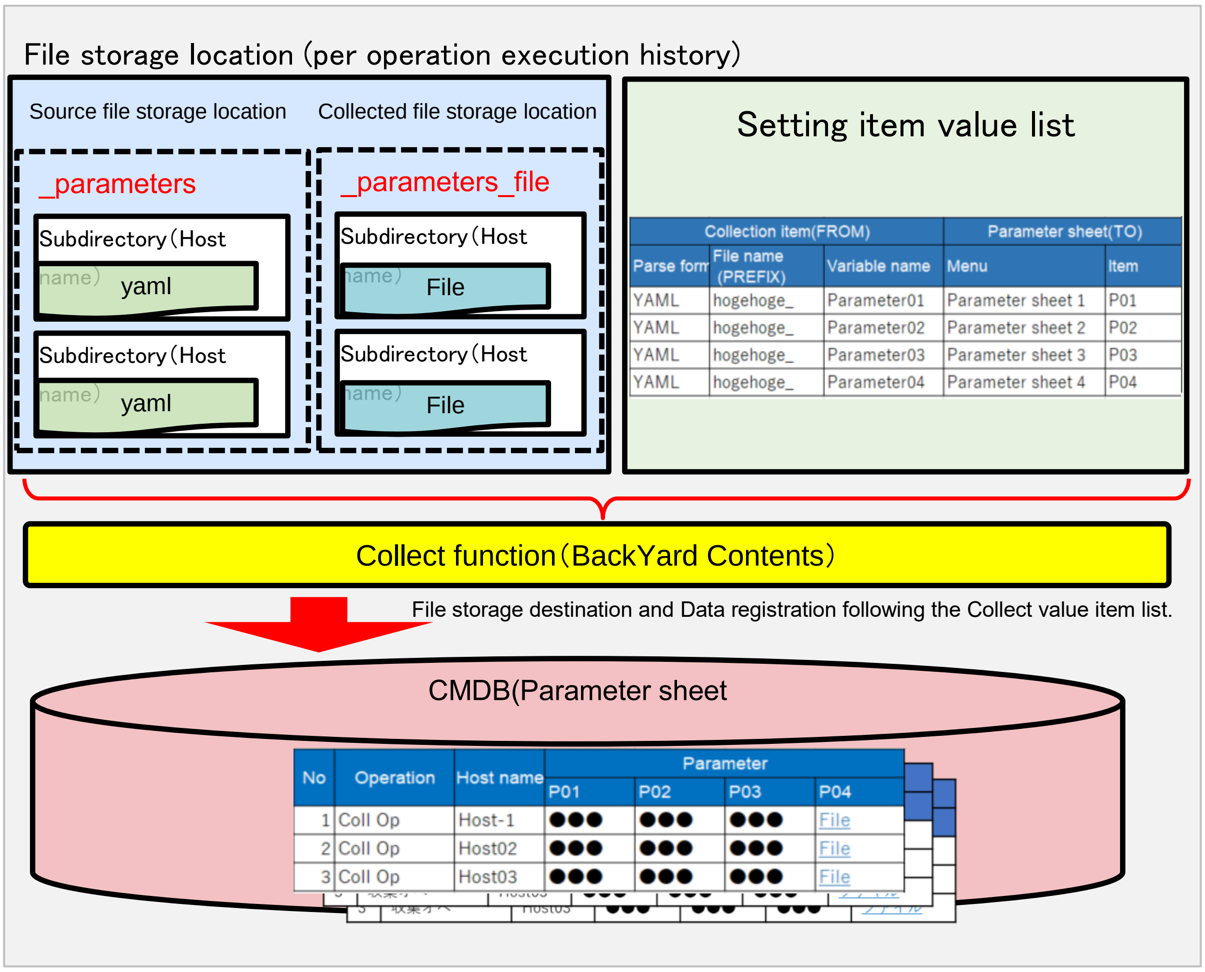
1.1.1 Collect function overview diagram

The following is a diagram that displays the entire process of using the Collect function.



1.1.2 **Collect function Data registration process overview diagram**

The following is a diagram of the Collect function Data registration process.


File storage location (per operation execution history)
Source file storage location
Collected file storage location
_parameters
Subdirectory(Host name)
yaml
_parameters_file
File
Setting item value list
Collect function(BackYard Contents)
File storage destination and Data registration following the Collect value item list.
CMDB(Parameter sheet

Collection item(FROM)		Parameter sheet(TO)		
Parse form	File name (PREFIX)	Variable name	Menu	Item
YAML	hogehoge_	Parameter01	Parameter sheet 1	P01
YAML	hogehoge_	Parameter02	Parameter sheet 2	P02
YAML	hogehoge_	Parameter03	Parameter sheet 3	P03
YAML	hogehoge_	Parameter04	Parameter sheet 4	P04

No	Operation	Host name	Parameter			
			P01	P02	P03	P04
1	Coll Op	Host-1	●●●	●●●	●●●	File
2	Coll Op	Host02	●●●	●●●	●●●	File
3	Coll Op	Host03	●●●	●●●	●●●	File

For more information regarding data types for the collect function, please see “7.2.2 Collection file values”.

1.2 Parametersheets registration

The collect function is an option of ITA and uses ITA's standard REST API function for the Parameter sheet registration process

For more information about the REST API Function, please refer to "Exastro-ITA_User_Instruction_Manual_RESTAPI"

1.2.3 **Collect function requirements**

Make sure that the requirements below are met.

- ITA is installed with "Createparam" and "Ansible_driver" selected. (done in the installer)
- A parameter sheet (with Host/Operation) is created in the Menu definition/creation screen.
- The registration information (source file) is linked to the items in the Parameter sheet in the "Setting value item list"
- The Collection interfance information's REST access information is updated.
- The Collection target device (Host name) is already registered in the device list.

If the executed operations outputs any of the statuses below, it will be registered to the parameter sheet.

- Operation execution result, the operation has successfully ended.
- Directories and files are arranged in a specific structure as a result of the output of the operation execution.

  ※Each user must prepare the IaC(Plabook, Role) that generates source files going to be registered to the parameter sheets.

  Reference：Ansible Playbook Collection（OS Setting collection）
  https://github.com/exastro-suite/playbook-collection-docs/blob/master/README.ja.md

2 Handling Directories, File structures and varibles in the Collect function.

2.1 Collectable Directories and File structures.

2.1.1 Collectable File formats

(1) Files output in YAML format.

```
e.g.)
■File name：RH_snmp.yml
■File contents：
VAR_RH_sshd_config:
  - key: PermitRootLogin
    value: yes
  - key: PasswordAuthentication
    value: no
```

2.1.2 Collectable Directory configuration

The collectable directory path (output destination for the source file) can be handled as the following variable in IaC (Playbook, Role).

Table 2-1 Collectable directory ITA Original variables

ITA original variable	Variable specified contents	Remarks
__parameter_dir__	「_parameters」 Operation result directory path	
__parameters_file_dir__	「_parameters_file」Operation result directory path	
__parameters_dir_for_epc__	「_parameters」Operation result directory path	
__parameters_file_dir_for_epc__	「_parameters_file」Operation result directory path	

The upper directory of the collectable directories (parameters) depends on the "Data relay storage path (Ansible", Ansible driver execution mdoe and the No. of the operation.
(The "Data relay storage path (Ansible) can be found in Ansible Common -> Interface information in ITA.)

Table 2-2 Collect function target Directory and file hierarchy

Hierarchy structure	Remarks

【Upper directory】
    |- _parameters ※1
    | |- localhost ※2
    | |- SAMPLE.yml ※3
    |- _parameters_file ※4
    | |- localhost ※2
    |- test.txt ※5
	※1 Collectable directory（Fixed name）
	※2Host name
	（Items registered in the device list are collectable）
	※3 Collectable file
	※4Collectable directory for file uploads（Fixed name）
	※5 Uploadable file

※Hierarchical structure after data relay storage path (Ansible)

When creating a playbook that generate source files, not using the “Table 2-1 Collectable directory ITA Original variables”” for the output destination will require the user to write the Playbook with the following structure in mind.

Table 2-3 Upper directory paths for the different Ansible-Driver modes

Mode	Mode identifier	Hierarchy structure	Remarks
Ansible-Legacy	legacy/ns/	/DataRelayStoragePath(Ansible)/legacy/ns/	
Ansible-Pioneer	pioneer/ns/	/ DataRelayStoragePath(Ansible)/pioneer /ns/	
Ansible-LegacyRole	legacy/rl/	/ DataRelayStoragePath(Ansible)/legacy/rl/	

e.g.) Collectable file paths and directory structures
Execution mode: Ansible-Legacy
Operation No : 1
Target host: localhost
Operation execution directory; /DataRelayStoragePath (Ansible)/legacy/ns/0000000001/in/
Operation results directory; /DataRelayStoragePath (Ansible)/legacy/ns/0000000001/out/

Collectable file path and directory structures:
/ DataRelayStoragePath (Ansible)/legacy/ns/0000000001/in/_parameters/localhost/SAMPLE.yml
/ DataRelayStoragePath (Ansible)/legacy/ns/0000000001/in/_parameters/localhost/OS/RH_snmpd.yml
/ DataRelayStoragePath (Ansible)/legacy/ns/0000000001/in/_parameters_file/localhost/TEST.txt

Or,

/ DataRelayStoragePath (Ansible)/legacy/ns/0000000001/out/_parameters/localhost/SAMPLE.yml
/ DataRelayStoragePath (Ansible)/legacy/ns/0000000001/out/_parameters/localhost/OS/RH_snmpd.yml
/ DataRelayStoragePath (Ansible)/legacy/ns/0000000001/out/_parameters_file/localhost/TEST.txt

If the user wants the file upload menu to be collectable, a file with the same name as the value of the source file variable (file name) must be placed under _parameters_.
For more information about Collection item value list settings, please refer to “5.1.2 Collection item value list”
As the maximum file size for uploads depends on the server specifications, please refer to “Exastro-ITA_User_Instruction_Manual_RESTAPI” for more details.

Way of selecting uploadable file under _parameters_file are as following.

Table 2-4 Specifying Upload target files

Format	YAML file description format	Remarks
File name specification	VAR_FILE_NAME : <‘ file name>’	
File path specification (Perfect match)	VAR_FILE_NAME : ‘/<upper directory>/_parameters_file/ localhost/<directoryX>/<file name>’	
File path specification (Ending match)	VAR_FILE_NAME : ‘/<directoryX>/<file name>’	

※Target file will be randomly selected when multiple files are applicable with file name.
Applicable file are selected for file path (perfect match)
Target file will be randomly selected when multiple files are applicable for file path (end match)

e.g.) Directory structure and source file contents when using variables of Normal variable structure.
■Structure
【Upper directory】

```
        |-  _parameters
        |    |-  localhost
        |         |-  SAMPLE.yml       ※Source file
        |-  _parameters_file
        |    |-  localhost
                 |-  test.txt              ※Uploadable file
```

■Collectable file name：SAMPLE.yml
■File contents
VAR_upload_file: test.txt

2.2 Variable and variable types

The following 3 types of variables can be handled in the Collect function source file.

Table 2.1 Variables and types

Type	Contents	Remarks
Normal variable	Can have one specific value defined per each variable name. e.g.) VAR_users: root	
Multiple specific value variable	Can have multiple specific values defined per each variable name e.g.) VAR_users: - root - mysql	
Multistage variable	Hierarchical variable. e.g.) VAR_users: - user-name: alice authorized: password } Member variable Member variable names can contain any ascii character excluding the seven characters below. (‘0x20~0x7e can be used) " . [ ] ' ¥ : Keep in mind that there are a few characters that can’t be used at the beginning of a variable name unless they are enclosed in quotation marks.For more information, please refer to the Yansible “Yaml syntax” document.	

3 Collect function console menu

This section explains the ITA Console menu structure

3.1 Menu/Screen list

① Ansible common console menu

The Ansible common console menu list is as following.

Table 3-1 Common console Menu/screen list

No	Menu group	Menu/Screen	Description
1	Ansible common console	Collection interface information	Manage the connection interface information to the server that accesses the ITA standard REST Function. The REST function is used when registering data to parameter sheets.
2		Collected item value list	Set up the connection between the executed operation output results (Source file) and the parameter sheet items and manages the Collection function parameter sheets.

② Ansible console menu

The list of menus corresponding to the Ansible consoles are as written below.

Table 3-2 Ansible driver console Menu/Screen list

No	Menu group			Menu/Screen	Description
	Ansible Console				
	Legacy	Legacy Role	Pioneer		
14	○	○	○	Execution list	Manages operation execution history. Refers to the registration status of the parameter sheet and execution log by the Collect function.

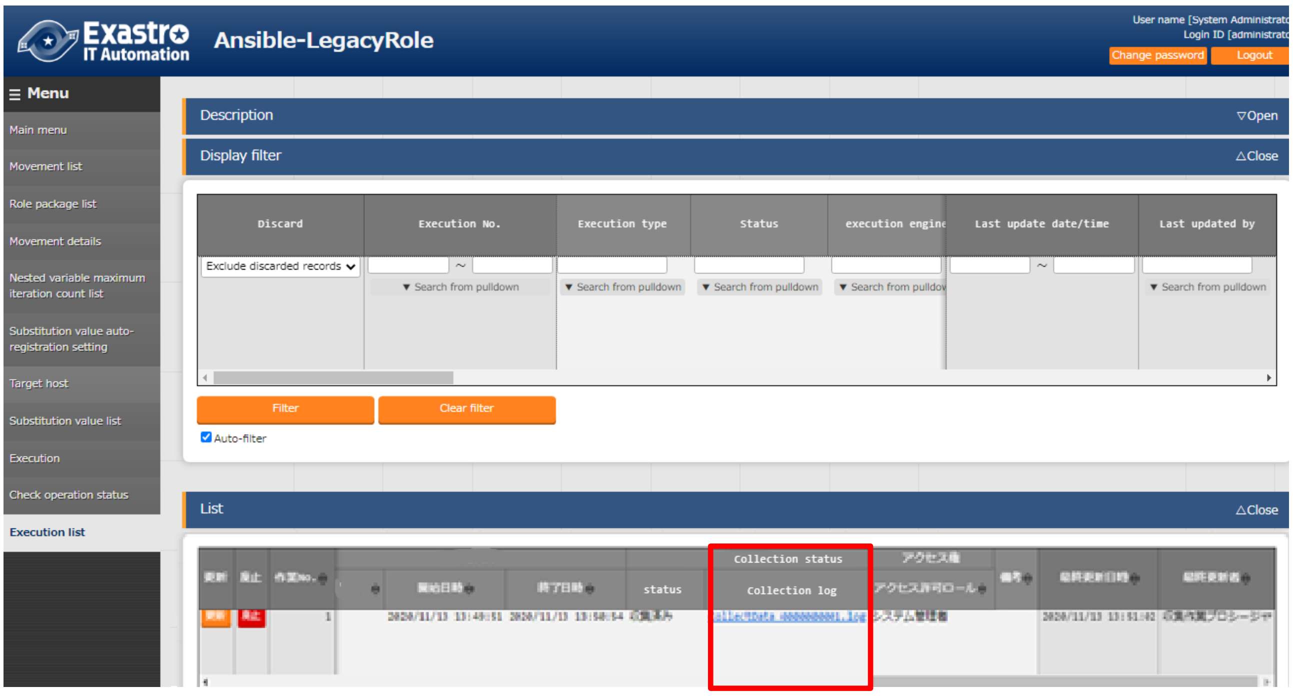


Figure 3.1-1 Execution list screen

4 Collect function user manual

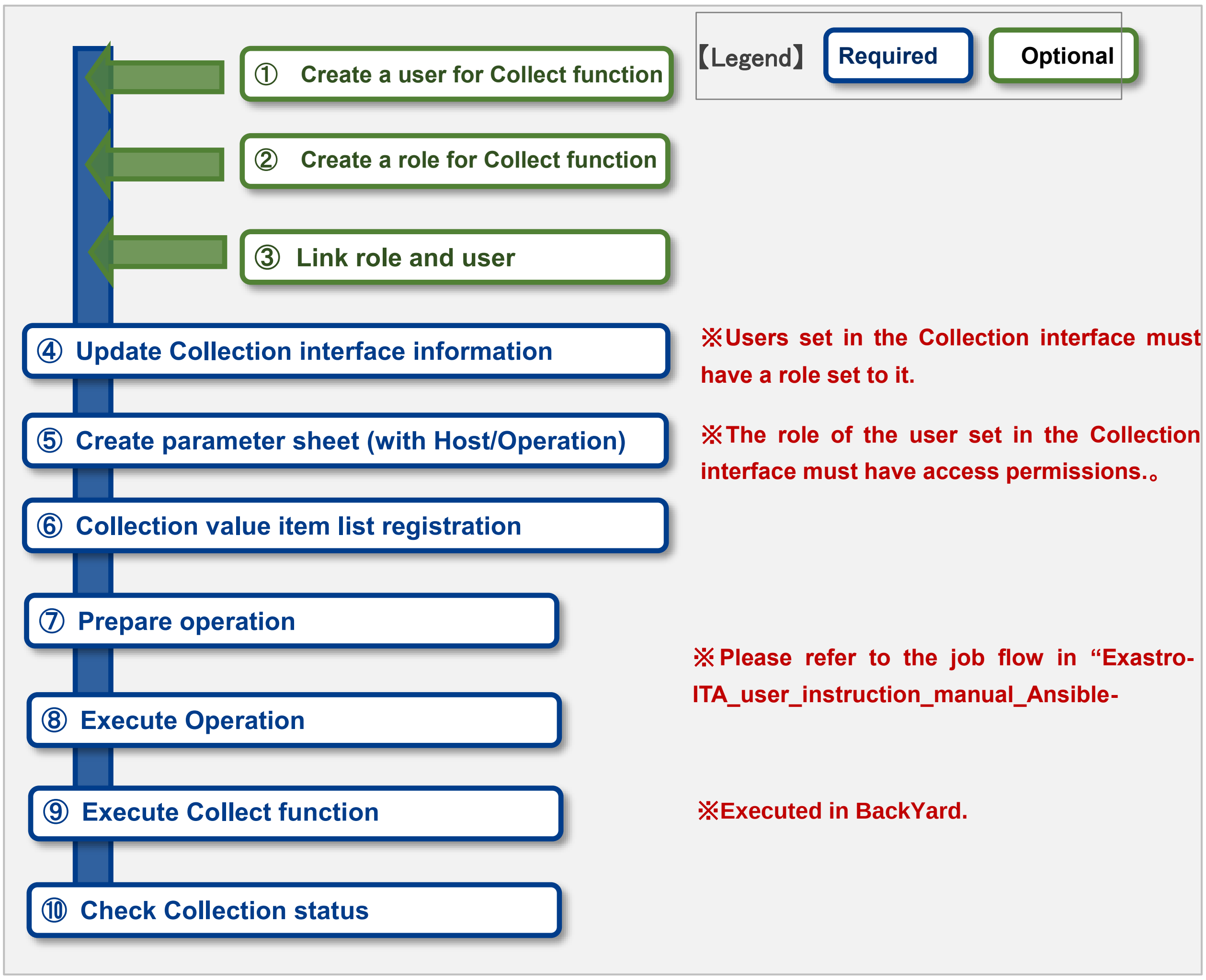
This section describes the how to use the Collect function.

4.1 Work flow.

The standard workflow for implementing the Colelct function is as following
For details on how to use ITA Ansible-Driver, please refer to Exastro-ITA_User_Instruction_Manual_Ansible-driver"
For details on how to use ITA Basic console, please refer to Exastro-ITA_User_Instruction_Manual_Basic_console"

4.1.1 **Collect function work flow.**

The following is the process before using Ansible-Legacy


【Legend】
Required
Optional
① Create a user for Collect function
② Create a role for Collect function
③ Link role and user
④ Update Collection interface information
⑤ Create parameter sheet (with Host/Operation)
⑥ Collection value item list registration
⑦ Prepare operation
⑧ Execute Operation
⑨ Execute Collect function
⑩ Check Collection status
※Users set in the Collection interface must have a role set to it.
※The role of the user set in the Collection interface must have access permissions.。
※Please refer to the job flow in "Exastro-ITA_user_instruction_manual_Ansible-
※Executed in BackYard.

- **Workflow and references.**

① **Create a user for the Collect function.**
Register a user for the Collect function in the ITA Management Console - Device list screen.
For details on how to register, please refer to "Exastro-ITA_User_Instruction_Manual_Management_console."

② **Create a role for the Collect function**
Register a role for the Collect function in the ITA Management Console – Role list screen
For details on how to register, please refer to "Exastro-ITA_User_Instruction_Manual_Management_console."

③ **Link role and user**
Link the role and user in the ITA Management console – Role/User link screen
For details, please refer to "Exastro-ITA_User_Instruction_Manual_Management_console."

④ **Register Collection interface information**
Register the connection information in the Ansible Common console – Collection interface information screen
For details, please refer to "5.1.1 Collection interface information"

⑤ **Create Parameter sheet (with host/operation)**
Create a parameter sheet in the Menu creation console – Menu definition/creation screen
For details, pelase refer to "Exastro-ITA_User_Instruction_Manual_Menu_creation_function"

⑥ **Register Collection item value list.**
Register the information that links the source files to the items in the parameter sheet.
(Ansible common console – Collection item value list screen)
Fore details, please refer to "5.1.2 Collection item value list".

⑦ **Prepare Operation**
Prepare the Operation to be executed.
For details, please refer to "Exastro-ITA_User_Instruction_Manual_Ansible-Driver", "Exastro-ITA_User_Instruction_Manual_Symphony" and "Exastro-ITA_User_Instruction_Manual_Conductor".

⑧ **Execute Operation**
Select the execution date/time, input operation, movement and workflow, and start the execution process.
For details regarding execution, please refer to "Exastro-ITA_User_Instruction_Manual_Ansible-Driver", "Exastro-ITA_User_Instruction_Manual_Symphony" and "Exastro-ITA_User_Instruction_Manual_Conductor".

⑨ **Execute Collect function**
Initiate the Parameter sheet registration process with the executed operation's operation No. as target for the Collect function.
For details, please refer to "5.3 BackYard contents".

⑩ **Check Collection status**

Ain the operation list screen, (Ansible-Legacy/ Ansible-Pioneer/Ansible-LegacyRole), users can check the Collection status of completed operations and download the log file(s).
For details, please refer to “5.2.1 Check Collection status”

5 Collect function operation explanation

This section explains how to operate the Collect function.

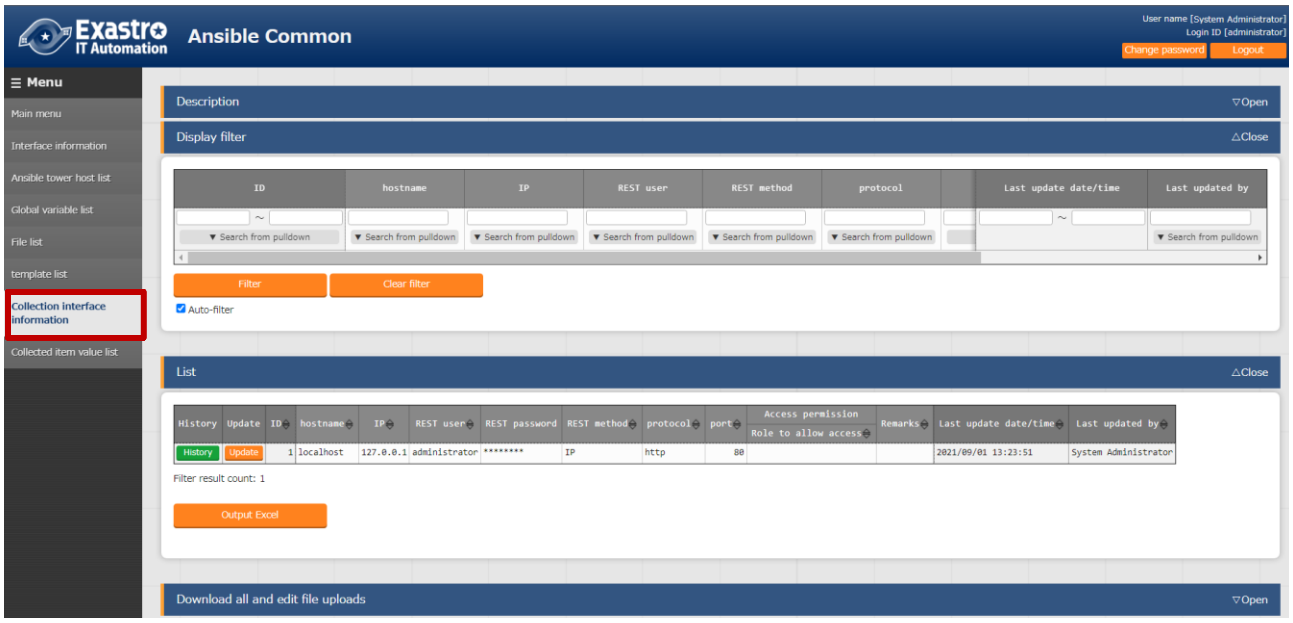
For details on how to register, please refer to “Exastro-ITA_User_Instruction_Manual_Basic_console”

5.1 Ansible Common console

This section explains how to operate the Ansible Common console.

5.1.1 Collection interface information

(1) Since the ITA’s standard REST API is used in this menu, it is required to update the Connection interface information for RESTAPI.

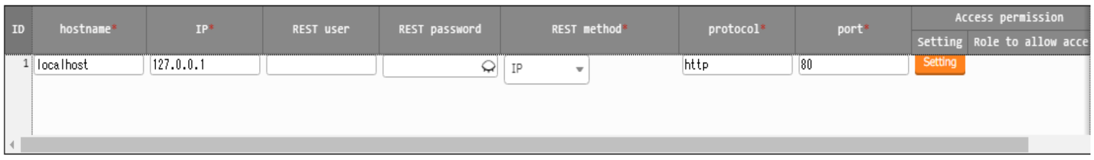


History	Update	ID	hostname	IP	REST user	REST password	REST method	protocol	port	Access permission Role to allow access	Remarks	Last update date/time	Last updated by
History	Update	1	localhost	127.0.0.1	administrator	********	IP	http	80			2021/09/01 13:23:51	System Administrator

Filter result count: 1

Figure 5.1-1 Submenu screen（Collection interface information）

(2) Register Collection interface information with the “List”-“Update” button.



ID	hostname*	IP*	REST user	REST password	REST method*	protocol*	port*	Access permission Setting	Role to allow acce
1	localhost	127.0.0.1			IP	http	80	Setting	

Figure 5.1-2 Update screen（Collection interface information）

(3) The item list for the Collection interface information is shown below.
If the operation was executed with no Collection interface information registered or with multiple records registered, the Collect function will not register any information to the

Parameter sheet.

Table 5.1-1 Registration screen, Item list（Interface information）

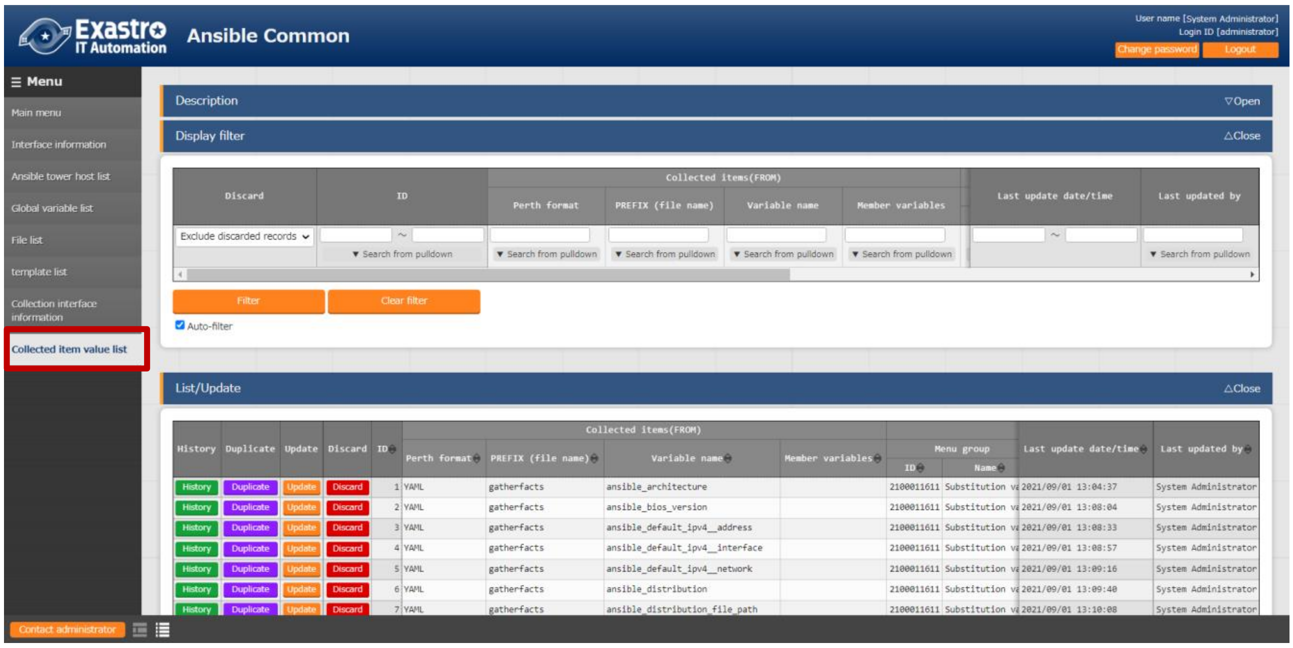
Item	Description	Input required	Input method	Constraints
Host name	Input host name Initial value：localhost	○	Manual input	
IP	Input IP Address Initial value：127.0.0.1	○	Manual input	
REST user	Input ITA user login ID		Manual input	※1
REST password	Input ITA user login password		Manual input	
RESTmethod	Choose IP or Host name ● IP ● Host name	○	Choose from list	
Protocol	Input protocol Initial value：http	○	Manual input	
Port	Input port Initial value：80	○	Manual input	
Remarks	Free description field	-	Manual input	

※1 Users entered in the “REST user” field will have the following required.
- The role that the user belongs to has to have permission to access the menu items in the created parameter sheet.
- The role linked to the user (in the Menu’s role information) has to be “Can Maintain” set to it.

For more information regarding Users, creating Roles and linking them, please refer to “Exastro-ITA_User_Instruction_Manual_Management_console.”

5.1.2 **Collection item value list**

(1) In the “Collection item value list”, set the link between the Colelction items and the items in the parameter sheet.

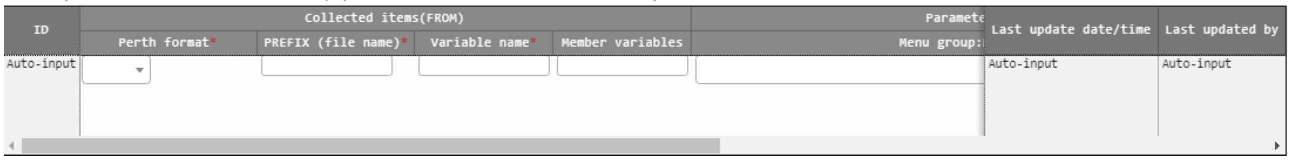


Discard	ID	Collected items(FROM)				Last update date/time	Last updated by
		Perth format	PREFIX (file name)	Variable name	Member variables		
Exclude discarded records	~					~	
	▼ Search from pulldown	▼ Search from pulldown	▼ Search from pulldown	▼ Search from pulldown	▼ Search from pulldown		▼ Search from pulldown

History	Duplicate	Update	Discard	ID	Collected items(FROM)				Menu group		Last update date/time	Last updated by
					Perth format	PREFIX (file name)	Variable name	Member variables	ID	Name		
History	Duplicate	Update	Discard	1	YAML	gatherfacts	ansible_architecture		2100011611	Substitution va	2021/09/01 13:04:37	System Administrator
History	Duplicate	Update	Discard	2	YAML	gatherfacts	ansible_bios_version		2100011611	Substitution va	2021/09/01 13:08:04	System Administrator
History	Duplicate	Update	Discard	3	YAML	gatherfacts	ansible_default_ipv4__address		2100011611	Substitution va	2021/09/01 13:08:33	System Administrator
History	Duplicate	Update	Discard	4	YAML	gatherfacts	ansible_default_ipv4__interface		2100011611	Substitution va	2021/09/01 13:08:57	System Administrator
History	Duplicate	Update	Discard	5	YAML	gatherfacts	ansible_default_ipv4__network		2100011611	Substitution va	2021/09/01 13:09:16	System Administrator
History	Duplicate	Update	Discard	6	YAML	gatherfacts	ansible_distribution		2100011611	Substitution va	2021/09/01 13:09:40	System Administrator
History	Duplicate	Update	Discard	7	YAML	gatherfacts	ansible_distribution_file_path		2100011611	Substitution va	2021/09/01 13:10:08	System Administrator

Figure 5.1-3 Submenu screen（Collection item value list）

(2) Register Collection item(s) with the “List“-“Start Registration” button.



ID	Collected items(FROM)				Paramete	Last update date/time	Last updated by
	Perth format*	PREFIX (file name)*	Variable name*	Member variables	Menu group:		
Auto-input						Auto-input	Auto-input

Figure 5.1-4 Registration screen（Collection item value list）

(3) The Collection item value list screen’s item list is as follows.

Table 5.1-1 Registration screen Item list（Collection item value list）

Item		Description	Input required	Input method	Constraints
Collected items (FROM)	Parse format	Select source file format.	○	Select from list	
	PREFIX(File name)	Enter the file name of the source file (Exclude the file extension).	○	Manual input	※1
	Variable name	Input variable name	○	Manual input	※1
	Member variables	Input if the variable is a multilevel variable or if it has multiple concrete		Manual input	※1

Item		Description	Input required	Input method	Constraints
		valies.			
Parameter sheet(TO)	Menu group	Select from a list of menus created by the Menu creation function Group name：Menu name	○	Select from list	
	Menu				
	Item	Select item.	○	Select from list	

※1 Example of file name, variable and member value input value

```
e.g.) If the variable has a normal variable structure.
■File name：SAMPLE.yml
■File contents

VAR_sample_config_1: yes
VAR_sample_config_2: test_parameter


■Values that can be input in the Collected item (from) in the Collected value item list.
PREFIX(File name)：        SAMPLE
Variable name：            VAR_sample_config_1
                           VAR_sample_config_2
```

```
e.g.) If the variable has a multiple variable structure.
■File name：SAMPLE_2.yml
■File contents

VAR_sample2_conf:
  SAMPLE1
  SAMPLE2
  SAMPLE3
■Values that can be input in the Collected item(from) in the Collected value item list.
PREFIX(File name)：        SAMPLE_2
Variable name：            VAR_sample2_conf
Member variables：        [0]
                           [1]
                           [2]
```

e.g.) If the variables has Multiple specific value structure.
■File name：RH_sshd.yml
■File contents

```
VAR_RH_sshd_config:
  - key: PermitRootLogin
    value: yes
  - key: PasswordAuthentication
    value: no
```

■Values that can be input in the Collected item(from) in the Collected value item list.

```
PREFIX(File name)：        RH_sshd
Variable name：                      VAR_RH_sshd_config:
Member variables：                   [0].key
                           [0].value
                           [1].key
                           [1].value
```

e.g.) If the variable has Multiple specific value structure 2
■File name：RH_snmp.yml
■File contents

```
VAR_RH_snmpd_info:
  com2sec:
    - sec_name: "testsec"
      source: "192.168.1.0/24"
      community: "public"
    - sec_name: "local"
      source: "localhost"
      community: "private"
```

■Values that can be input in the Collected item(from) in the Collected value item list.

```
PREFIX(File name)：        RH_snmp
Variable name：            VAR_RH_snmp_config:
Member variables：         com2sec[0].sec_name
                           com2sec[0].source
                           com2sec[0].community
                           com2sec[1].sec_name]
                           com2sec[1].source
                           com2sec[1].community
```

5.2 Ansible-Legacy、Ansible-Pioneer、Ansible-LegacyRole Console

5.2.1 Check Collection status

It is possible to check the status of completed operations and download the log files in each console's (Ansible-Legacy/ Ansible-Pioneer/Ansible-Legacy role) Execution list screen.

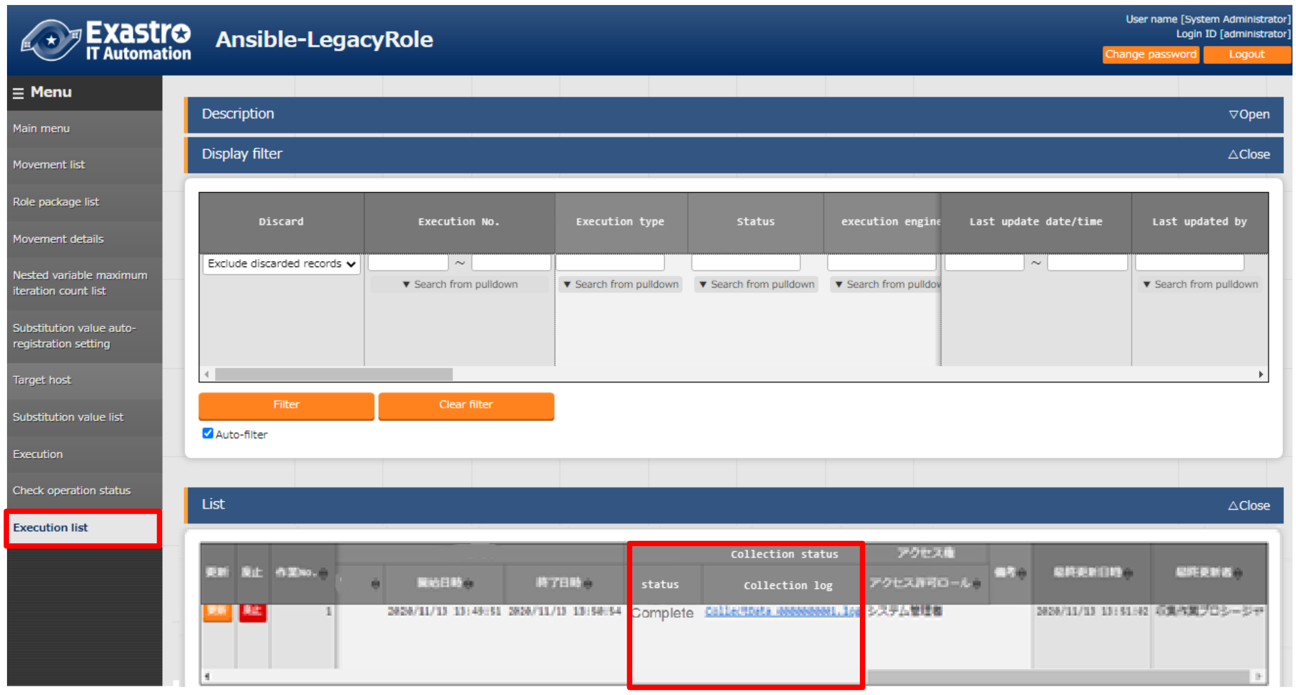


Figure 5.2-1 Execution list screen

Table 5.2-1 Execution list Collection status details

Item	Description	Remarks
Status	Collection function execution status Not target; Not a Collect function target (No target file) Collected: Collect function executed. Collected (with notification): If any errors occurred during registration/update. Collection error：Flaw is found on Movement operation or host	※
Collection log	Download the collect function execution log.	

Table 5.2-2 Collection status details

Operation status		Collect function target	Collection status		Remarks
Status	Target file		Status	Collection log	
Other than Complete	No	Not target	Blank	Blank	

Operation status		Collect function target	Collection status		Remarks
Status	Target file		Status	Collection log	
Other than Complete	Yes	Not target	Blank	Blank	
Complete	No	Target	Not target	Blank	
Complete	Yes	Target	Collected	With log file	
Complete	No	Target	Collected (with notification)	With log file	

※Regarding Status notations

- If the Operation status shows “Not complete”, the collection status will not be updated because it is not subject to the Collect function. Therefore, it will remain as “Blank”
- If the operation status shows as “Complete” and there are no files to collect, the status will show as “Collected” and the collection log will be blank.
- Even if the RESTAPI registration process fails during collecting the “ Collection interface information”, Settings item value list” or “ Menu access rights/permission roles”, the collection will show as “ Complete (with notification)”.

Example of Log file output contents.

```
e.g.)Example of Log file output contents (Registration process succeeded)
2020-11-13 13:51:02 Collect START ( Host name:ita-sample File name:RH_snmpd )
2020-11-13 13:51:02 REST DATA ( Host name: ita-sample Menu ID: 0000000004 Operation NO: 1 )
Array
(
    [0] => http://127.0.0.1:80/default/menu/07_rest_api_ver1.php?no=0000000004
    [1] => [["update","","3","ita-sample","","","","","","2023¥/10¥/26 16:35_1:OP001","Root
<root@localhost> (configure ¥/etc¥/snmp¥/snmp.local.conf)","Unknown (edit
¥/etc¥/snmp¥/snmpd.conf)","public","notConfigUser","","","T_2020111115557819037",""]]
    [2] =>
{"status":"SUCCEED","resultdata":{"LIST":{"NORMAL":{"register":{"name":"¥u767b¥u9332","ct":0},"update
":{"name":"¥u66f4¥u65b0","ct":1},"delete":{"name":"¥u5ec3¥u6b62","ct":0},"revive":{"name":"¥u5fa9¥u6d3
b","ct":0},"error":{"name":"¥u30a8¥u30e9¥u30fc","ct":0}},"RAW":[["000","200",""]]}}}
)
2020-11-13 13:51:02 Collect END ( Host name:ita-sample File name:RH_snmpd )
```

```
e.g.) Example of Log file output contents (Registration process failed)
2020-11-06 13:32:52 Collect START ( Host name:ita-sample File name:RH_snmpd )
2020-11-06 13:32:52 [process]REST Access failed
Array
(
    [0] => http://127.0.0.1:80/default/menu/07_rest_api_ver1.php?no=0000000005
    [1] => [["Register","","","ita-sample","","","","","","2023¥/10¥/26 16:35_1:OP001","Root
<root@localhost>(configure ¥/etc¥/snmp¥/snmp.local.conf)","Unknown (edit
¥/etc¥/snmp¥/snmpd.conf)","public","notConfigUser","","","",""]]
    [2] =>
{"Error":"¥u30e1¥u30f3¥u30c6¥u30ca¥u30f3¥u30b9¥u6a29¥u9650¥u304c¥u3042¥u308a¥u307e¥u305b
¥u3093¥u3002","Exception":"Generic error","StackTrace":"none"}
)
2020-11-06 13:32:52 Collect END ( Host name:ita-sample File name:RH_snmpd )
```

```
e.g.) Example of Log file output contents (Not target)
2020-11-05 16:55:31 [Process]The target device is not registered or is obsolete, so skip the registration
and update process(Host name:ita-test)
```

5.3 BackYard contents

5.3.1 Overview of the Parameter sheet registration process.

(1) Acquire Collection interface information

(2) Acquire list over completed operations (with Normal end)

Collection target status :	Complete

(3) Acquire the following information from the collectable operation no.

- Operation information
- Target host
- Target source file

(4) Inquire wether the target host is registered in the Device list or not

Registered :	Collectable
Not registered :	Not collectable

(5) Acquire the Menu ID of the target Parameter sheet from the source file and the Collection item value list.

(6) Create RESTAPI Parameter with the information gathered in Step 1-4.

Query the Menu ID for data and determine the RESTAPI Execution type.

Register :	Unique operation and Host combination data is not registered.
Update :	Unique operation and Host combination data is registered

(7) Register/Update the data using ITA Standard RESTAPI functions

(8) Update the status of the Collection status to the Operation No.

Keep in mind that the timing of the data registration to the Parameter sheet depends on the startup cycle of the Automatic process.

For more information regarding changing the startup cycle, please refer to “6.2 Maintenance”.

The access permission roles of the Registered/Updated records will inherit the access permission roles of the Collectable operation results.。

For more information regarding Target operation results, please refer to “Exastro-ITA_User_Instruction_Manual_Ansible-driver”.

6 Operation

Operation that uses this function includes: Inputs from users using browsers from client PCs and Operations done directly from the system operation/maintenance.

6.1 Maintenance

The following files are required to Start/Stop/Restart the Collect function process.

Description	File name
Automatic Parameter registration The operation is executed and will be registered to the parameter sheet based on the information registered in the setting item value list from the Operation results.	ky_std_synchronize-Collector.service

The files are stored in「/usr/lib/systemd/system」
The Start/Strop/Restart process methods are as following:
(Execute the commands with Root privileges)

① Start process

```
# systemctl start ky_std_synchronize-Collector.service
```

① Stop process

```
# systemctl stop ky_std_synchronize-Collector.service
```

② Restart process

```
# systemctl restart ky_std_synchronize-Collector.service
```

Replace each file name with the target file name and start/stop/restart.

6.2 Maintenance

① Change level to NORMAL
Rewrite the eigth row, “NORMAL, to “DEBUG”.
Log level settings file： <Install directory>/ita-root/confs/backyardconfs/ita_env

② Change level to DEBUG
Rewrite the eigth row, “DEBUG”, to “NORMAL”.
Log level settings file： <Install directory>/ita-root/confs/backyardconfs/ita_env

③ Change boot cycle.
Change the 5th parameter of ExecStart for each target file. (Unit: seconds)
Use the default value for boot cycles (except for exceptions).

```
ExecStart=/bin/sh ${ITA_ROOT_DIR}/backyards/common/ky_loopcall-php-procedure.sh
/bin/php /bin/php ${ITA_ROOT_DIR}/backyards/ansible_driver/ky_std_synchronize-
Collector.php ${ITA_ROOT_DIR}/logs/backyardlogs 10 ${ITA_LOG_LEVEL} > /dev/null
2>&1
```

Anything rewritten will take effect after the process is restarted.
Log file output destination： <Install directory>/ita-root/logs/backyardlogs

7 Appendix

7.1 References

Below are examples of IaCs (Playbook and Role)

1. Ansible Playbook Collection（Collect OS Settings）
   https://github.com/exastro-suite/playbook-collection-docs/blob/master/README.ja.md

2. Ansible config collecting and Parameter creating Playbook.

   makeYml_Ansible.yml

```
- name: make yaml file
  blockinfile:
    create: yes
    mode: 0644
    insertbefore: EOF
    marker: ""
    dest: "/tmp/Ansible_conf.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 }}
      ansible_processor_count: {{ ansible_processor_count }}
      ansible_processor_threads_per_core: {{ ansible_processor_threads_per_core }}
      ansible_processor_vcpus: {{ ansible_processor_vcpus }}
      ansible_product_name: {{ ansible_product_name }}
      ansible_product_serial: {{ ansible_product_serial }}
      ansible_product_uuid: {{ ansible_product_uuid }}
      ansible_product_version: {{ ansible_product_version }}
      ansible_python__executable: {{ ansible_python.executable }}
      ansible_python_version: {{ ansible_python_version }}
```

```
          ansible_service_mgr: {{ ansible_service_mgr }}
          ansible_php_config: php.ini

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

  - name: get php config
    fetch:
      src: /etc/php.ini
      dest:   "{{ __parameters_file_dir__ }}/{{ inventory_hostname }}/"
      flat: yes
```

※ When you run makeYML_Ansible.yml and generate the Collectable source file (yaml), you need to enable gather_facts.
When editing the Movement list in Ansible Legacy, enter the following in the header section.
For details regarding Changing settings, pleaser refer to
“Exastro-ITA_User_Instruction_Manual_Ansible-driver”.

e.g) gather_facts Valid setting example.

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

7.2 Examples from using the Collect function.

7.2.1 If you have multiple files with the same menu as target.

The following section shows an example of the directory and the collect process when you multiple "PREFIX (file name)-Variable names" set to a single "Menu-Item" and there are multiple corresponding source files in the target host's collect directory.

Target collection files

```
【Upper directory】
        |-  _parameters
             |-  ita-sample01
                  |-  SAMPLE_01.yml
                  |-  SAMPLE_02.yml
```

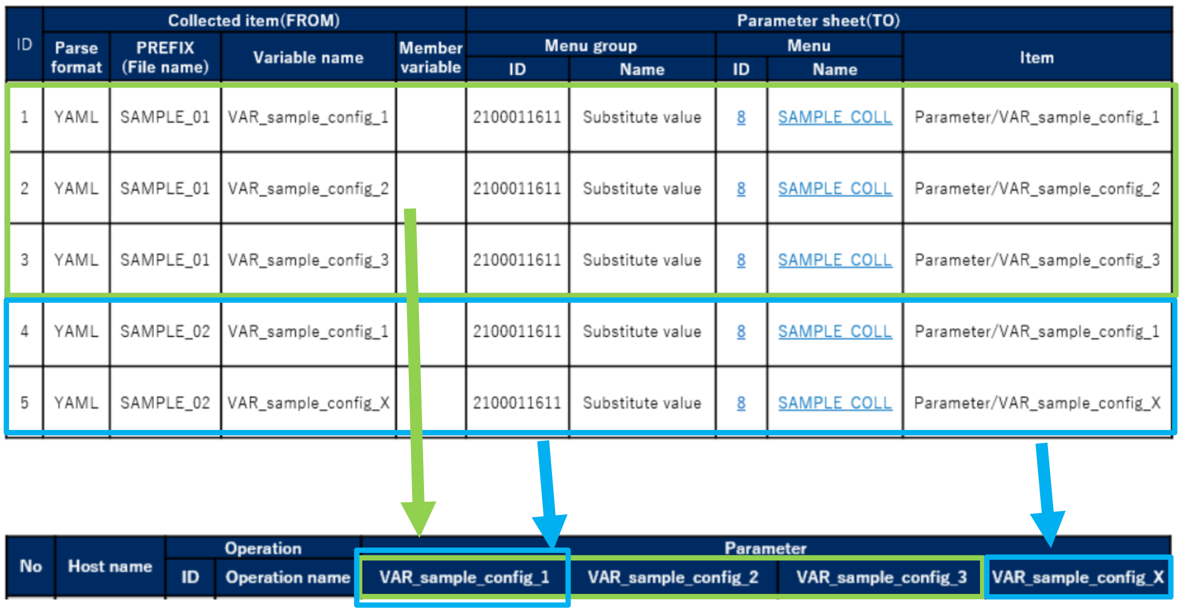
Collected item value management settings.

■File name：SAMPLE_01.yml
SAMPLE_02.yml

■File contents

SAMPLE_01.yml	SAMPLE_02.yml
VAR_sample_config_1: 1 VAR_sample_config_2: 2 VAR_sample_config_3: 3	VAR_sample_config_1: “A” VAR_sample_config_B: “B” VAR_sample_config_X: “X”

■Collecting target menu items and configuring collected value item lists.
e.g) Collected value item list settings and target menu items

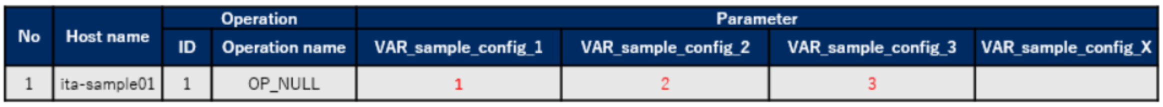


ID	Collected item(FROM)				Parameter sheet(TO)				
	Parse format	PREFIX (File name)	Variable name	Member variable	Menu group ID	Menu group Name	Menu ID	Menu Name	Item
1	YAML	SAMPLE_01	VAR_sample_config_1		2100011611	Substitute value	8	SAMPLE_COLL	Parameter/VAR_sample_config_1
2	YAML	SAMPLE_01	VAR_sample_config_2		2100011611	Substitute value	8	SAMPLE_COLL	Parameter/VAR_sample_config_2
3	YAML	SAMPLE_01	VAR_sample_config_3		2100011611	Substitute value	8	SAMPLE_COLL	Parameter/VAR_sample_config_3
4	YAML	SAMPLE_02	VAR_sample_config_1		2100011611	Substitute value	8	SAMPLE_COLL	Parameter/VAR_sample_config_1
5	YAML	SAMPLE_02	VAR_sample_config_X		2100011611	Substitute value	8	SAMPLE_COLL	Parameter/VAR_sample_config_X

No	Host name	Operation		Parameter			
		ID	Operation name	VAR_sample_config_1	VAR_sample_config_2	VAR_sample_config_3	VAR_sample_config_X

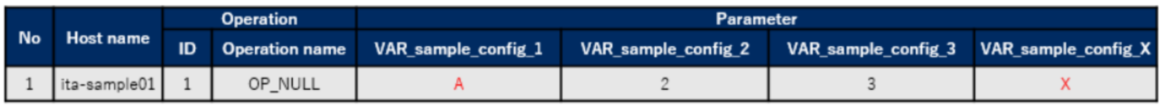
■Run the collect process for each file according to the target file settings and the collected value item management settings contents.

1. SAMPLE_01.yml registration process（Register）



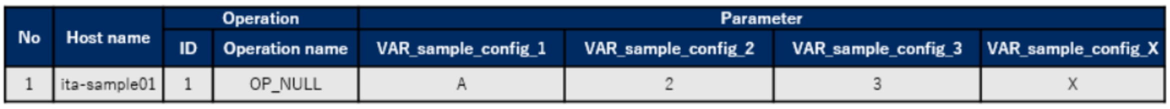
No	Host name	Operation		Parameter			
		ID	Operation name	VAR_sample_config_1	VAR_sample_config_2	VAR_sample_config_3	VAR_sample_config_X
1	ita-sample01	1	OP_NULL	1	2	3	

2.SAMPLE_02.yml registration process（Update）



No	Host name	Operation		Parameter			
		ID	Operation name	VAR_sample_config_1	VAR_sample_config_2	VAR_sample_config_3	VAR_sample_config_X
1	ita-sample01	1	OP_NULL	A	2	3	X

3. Record status after running the collect function.



No	Host name	Operation		Parameter			
		ID	Operation name	VAR_sample_config_1	VAR_sample_config_2	VAR_sample_config_3	VAR_sample_config_X
1	ita-sample01	1	OP_NULL	A	2	3	X

7.2.2 **Handling values of collectable file**

For collectable file output in Yaml format, handle parameter sheet registration values are handled as following.

Sample.yml

```
VAR_TEST: TEST
VAR_STR_TEST1: 'TEST1'
VAR_STR_TEST2: "TEST2"
VAR_null: null
VAR_NULL: NULL
VAR_STR_null: "null"
VAR_STR_NULL: "NULL"
VAR_true: true
VAR_false: false
VAR_STR_true: "true"
VAR_STR_false: "false"
VAR_YES: YES
VAR_NO: NO
VAR_STR_YES: "YES"
VAR_STR_NO: "NO"
VAR_NON:
VAR_Quotation: ''
VAR_WQuotation: ""
```

Table 7-1 Keys and values of Collectable YAML(sample.yml)

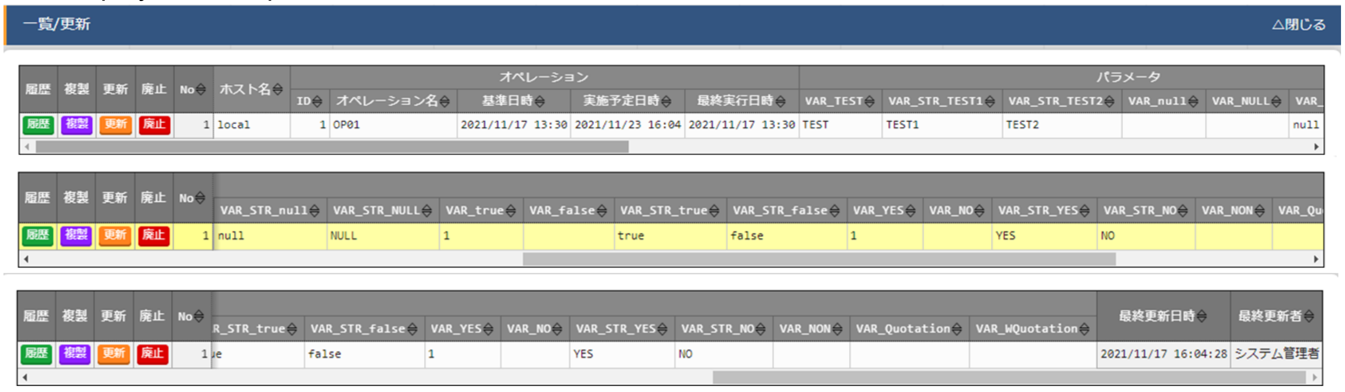
No	Keys	Value	Remarks
1.	VAR_TEST	TEST	
2.	VAR_STR_TEST1	'TEST1'	
3.	VAR_STR_TEST2	"TEST2"	
4.	VAR_null	null	
5.	VAR_NULL	NULL	
6.	VAR_STR_null	"null"	
7.	VAR_STR_NULL	"NULL"	
8.	VAR_true	true	
9.	VAR_false	false	
10.	VAR_STR_true	"true"	
11.	VAR_STR_false	"false"	
12.	VAR_YES	YES	
13.	VAR_NO	NO	
14.	VAR_STR_YES	"YES"	
15.	VAR_STR_NO	"NO"	
16.	VAR_NON		
17.	VAR_Quotation	''	
18.	VAR_WQuotation	""	

Table 7-2 Collecting collectable YAML(sample.yml)

No	Colletable (Key:value)	Parameter sheet	RESTAPI	WEB screen
1.	VAR_TEST: TEST	parameter/VAR_TEST	"TEST"	TEST
2.	VAR_STR_TEST1: 'TEST1'	parameter/VAR_STR_TEST1	"TEST1"	TEST1
3.	VAR_STR_TEST2: "TEST2"	parameter/VAR_STR_TEST2	"TEST2"	TEST2
4.	VAR_null: null	parameter/VAR_null	null	
5.	VAR_NULL: NULL	parameter/VAR_NULL	null	
6.	VAR_STR_null: "null"	parameter/VAR_STR_null	"null"	null
7.	VAR_STR_NULL: "NULL"	parameter/VAR_STR_NULL	"NULL"	NULL
8.	VAR_true: true	parameter/VAR_true	"1"	1
9.	VAR_false: false	parameter/VAR_false	""	
10.	VAR_STR_true: "true"	parameter/VAR_STR_true	"true"	true
11.	VAR_STR_false: "false"	parameter/VAR_STR_false	"false"	false
12.	VAR_YES: YES	parameter/VAR_YES	"1"	1
13.	VAR_NO: NO	parameter/VAR_NO	""	
14.	VAR_STR_YES: "YES"	parameter/VAR_STR_YES	"YES"	YES
15.	VAR_STR_NO: "NO"	parameter/VAR_STR_NO	"NO"	NO
16.	VAR_NON:	parameter/VAR_NON	null	
17.	VAR_Quotation: ''	parameter/VAR_Quotation	null	
18.	VAR_WQuotation: ""	parameter/VAR_WQuotation	null	

※Obtain result of RESTAPI FILTER value of parameter sheet listed are surrounded by “” parameter sheet.
※RESTAPI（FILTER）obtain results are listed in WEB screen display result.

- Display result of parameter sheet



一覧/更新 △閉じる

履歴	複製	更新	廃止	No	ホスト名	オペレーション					パラメータ					
						ID	オペレーション名	基準日時	実施予定日時	最終実行日時	VAR_TEST	VAR_STR_TEST1	VAR_STR_TEST2	VAR_null	VAR_NULL	VAR_
履歴	複製	更新	廃止	1	local	1	OP01	2021/11/17 13:30	2021/11/23 16:04	2021/11/17 13:30	TEST	TEST1	TEST2			null

履歴	複製	更新	廃止	No	VAR_STR_null	VAR_STR_NULL	VAR_true	VAR_false	VAR_STR_true	VAR_STR_false	VAR_YES	VAR_NO	VAR_STR_YES	VAR_STR_NO	VAR_NON	VAR_Qu
履歴	複製	更新	廃止	1	null	NULL	1		true	false	1		YES	NO		

履歴	複製	更新	廃止	No	R_STR_true	VAR_STR_false	VAR_YES	VAR_NO	VAR_STR_YES	VAR_STR_NO	VAR_NON	VAR_Quotation	VAR_WQuotation	最終更新日時	最終更新者
履歴	複製	更新	廃止	1	e	false	1		YES	NO				2021/11/17 16:04:28	システム管理者

- Obtain result of parameter sheet from RESTAPI(FILTER)

```
{
        "status": "SUCCEED",
        "resultdata": {
                "CONTENTS": {
                        "RECORD_LENGTH": 1,
                        "BODY": [
                                [
                                        "Exection process type",
                                        "Abollish",
                                        "No",
                                        "host name",
                                        "operation/ID",
                                        "operation/operation name",
                                        "operation/standard date",
                                        "operation/scheduled operation date",
                                        "operation/last operation date",
                                        "operation/operation",
                                        "parameter/VAR_TEST",
                                        "parameter/VAR_STR_TEST1",
                                        "parameter/VAR_STR_TEST2",
                                        "parameter/VAR_null",
                                        "parameter/VAR_NULL",
                                        "parameter/VAR_STR_null",
                                        "parameter/VAR_STR_NULL",
                                        "parameter/VAR_true",
                                        "parameter/VAR_false",
                                        "parameter/VAR_STR_true",
                                        "parameter/VAR_STR_false",
                                        "parameter/VAR_YES",
                                        "parameter/VAR_NO",
                                        "parameter/VAR_STR_YES",
                                        "parameter/VAR_STR_NO",
                                        "parameter/VAR_NON",
                                        "parameter/VAR_Quotation",
                                        "parameter/VAR_WQuotation",
                    "Access permission/Access permission role",
                    "Remarks",
                    "Last updated",
                    "Last updated date for update",
                    "Last updated user"
                ],
                [
                    null,
                    "",
                    "1",
                    "local",
                    "1",
                    "OP01",
                    "2021/11/17 13:30",
                    "2021/11/23 16:04",
                    "2021/11/17 13:30",
                    "2021/11/23 16:04_1:OP01",
                    "TEST",
                    "TEST1",
                    "TEST2",
                    null,
                    null,
                    "null",
                    "NULL",
                    "1",
                    "",
                    "true",
                    "false",
                    "1",
                    "",
                    "YES",
                    "NO",
                    null,
                    null,
                    null,
                    "",
                    null,
                    "2021/11/17 16:04:28",
                    "T_20211117160428242847",
                    "system administrator"
                ]
            ],
            "UPLOAD_FILE": []
        }
    }
}
```

7.2.3 **Selecting upload file from multiple file with same name**

Selecting upload file when collectable file output by Yaml format with same file name and same host but different hierarchy are as listed.

```
e.g.)  collectable file path and directory structure
【Upper directory】
        |-  _parameters
        |    |-  localhost
        |         |-  SAMPLE.yml
        |-  _parameters_file
        |    |-  localhost
                  |-  APP001
                       |-  config                 #①
                  |-  APP002
                       |-  config                 #②
                  |-  APP003
                       |-  config                 #③
                       |-  APP002
                           |-  config             #④

■collectable file name：SAMPLE.yml
■Content
VAR_upload_file_1: config
VAR_upload_file_2: '/<Upper directory>/_parameters_file /localhost /APP002/config'
VAR_upload_file_3: '/APP002/config'
VAR_upload_file_4: '/APP001/config'
VAR_upload_file_5: '/APP003/APP002/config'
```

※Refer to "エラー! 参照元が見つかりません。" for upper directory

Collectable file and target files are following

Table 7-3 collectable file and target files

Collectable item (FROM)/ variable name	Target file	Remarks
VAR_upload_file_1	Randomly from files ①、②、③、④	
VAR_upload_file_2	File ②	
VAR_upload_file_3	Randomly from files、②、④	
VAR_upload_file_4	File ①	
VAR_upload_file_5	File ④	

7.2.4 Collectable file contents when deleting files.

This section describes an example of specifying collection files when deleting files in a file upload item.
Users can delete files by changing the value of the target variable name to be blank.

```
e.g.)    Directory structure and file path for Collection files when deleting file upload items.
■Structure
【Upper directory】
        |-  _parameters
        |    |-  localhost
        |         |-  SAMPLE.yml                ※Source file
        |-  _parameters_file
        |    |-  localhost

■Collectable file name：SAMPLE.yml
■File contents
VAR_upload_file: ""                         ※Upload file
```