



ITA_利用手順マニュアル

収集機能

—第1.6版—

免責事項

本書の内容はすべて日本電気株式会社が所有する著作権に保護されています。

本書の内容の一部または全部を無断で転載および複製することは禁止されています。

本書の内容は将来予告なしに変更することがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任を負いません。

日本電気株式会社は、本書の内容に関し、その正確性、有用性、確実性その他いかなる保証もいたしません。

商標

- ・ LinuxはLinus Torvalds氏の米国およびその他の国における登録商標または商標です。
- ・ Red Hatは、Red Hat, Inc.の米国およびその他の国における登録商標または商標です。
- ・ Apache、Apache Tomcat、Tomcatは、Apache Software Foundationの登録商標または商標です。
- ・ Ansibleは、Red Hat, Inc.の登録商標または商標です。
- ・ AnsibleTowerは、Red Hat, Inc.の登録商標または商標です。

その他、本書に記載のシステム名、会社名、製品名は、各社の登録商標もしくは商標です。

なお、® マーク、TM マークは本書に明記しておりません。

※本書では「Exastro IT Automation」を「ITA」として記載します。

目次

目次	2
はじめに	3
1 収集機能概要	4
1.1 収集機能について	4
1.1.1 収集機能概要図	4
1.1.2 収集機能データ登録処理概要図	5
1.2 パラメータシートへの登録方法について	6
1.2.3 収集機能の動作要件	6
2 収集機能でのディレクトリ、ファイル構造、変数取り扱い	7
2.1 収集対象ディレクトリ、ファイル構造	7
2.1.1 収集対象のファイルフォーマット	7
2.1.2 収集対象ディレクトリ構造	7
2.2 取り扱う変数と種類	10
3 収集機能 コンソールメニュー構成	11
3.1 メニュー/画面一覧	11
4 収集機能の利用手順	13
4.1 作業フロー	13
4.1.1 収集機能作業フロー	13
5 収集機能・操作方法説明	15
5.1 Ansible 共通コンソール	15
5.1.1 収集インターフェース情報	15
5.1.2 収集項目値管理	17
5.2 Ansible-Legacy、Ansible-Pioneer、Ansible-LegacyRole コンソール	20
5.2.1 収集状況の確認	20
5.3 BackYard コンテンツ	22
5.3.1 パラメータシートへの登録処理の概要	22
6 運用操作	23
6.1 メンテナンス	23
6.2 メンテナンス方法について	23
7 付録	25
7.1 参考 URL	25

はじめに

本書では、ITA の機能および操作方法について説明します。

1 収集機能概要

本章では収集機能について説明します。

1.1 収集機能について

収集機能とは、ITA で実施した、作業実行結果（規定のフォーマットで出力されたソースファイル）を元に、パラメータシートへ値を自動で登録する機能です。

本機能は、Ansible-Driver を対象としています。

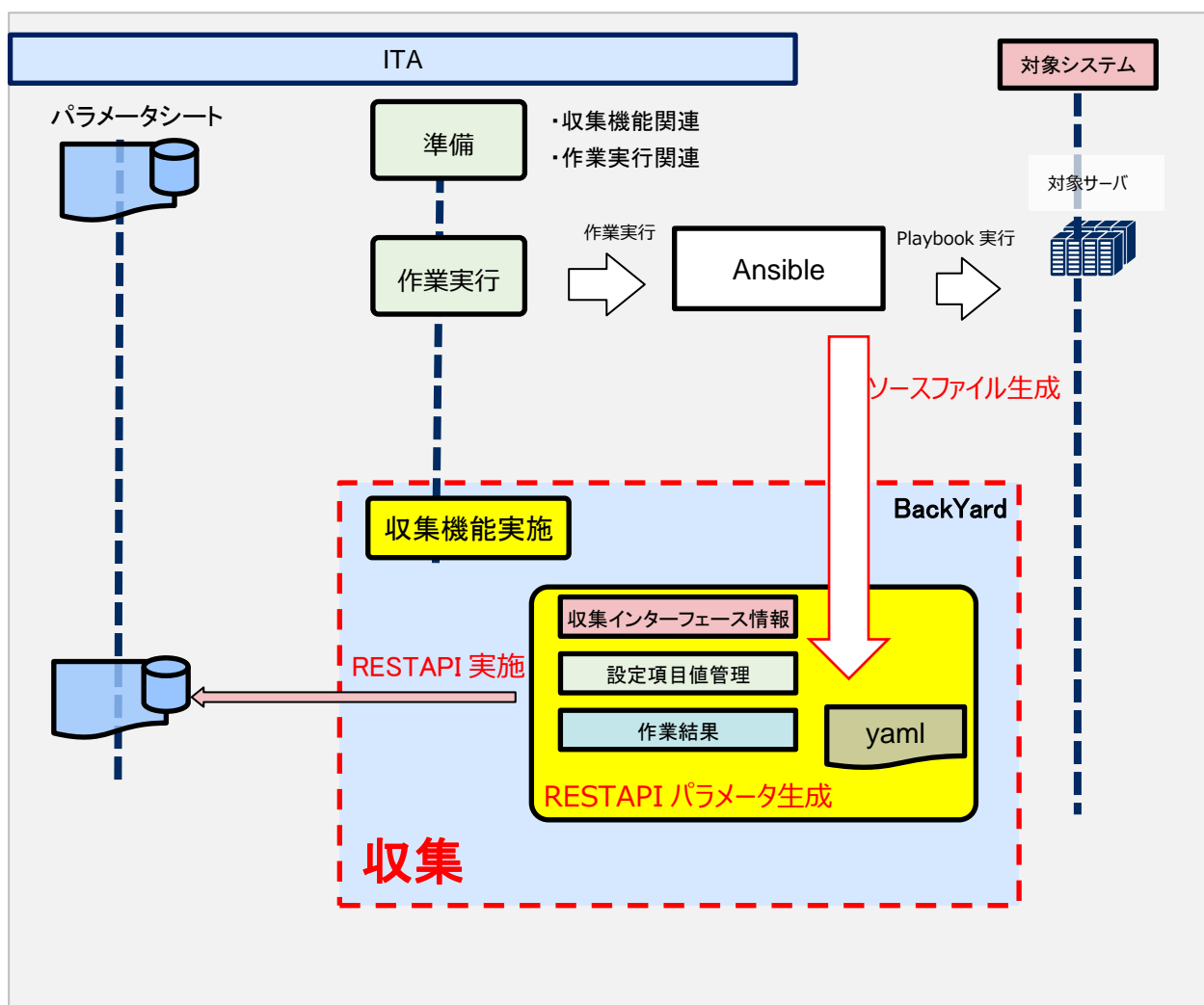
Ansible の詳細情報については、「Ansible の製品マニュアル」を参照してください。

Ansible-Driver の詳細情報については、「利用手順マニュアル Ansible-driver」を参照してください。

パラメータシートの詳細については、「利用手順マニュアル メニュー作成機能」を参照してください。

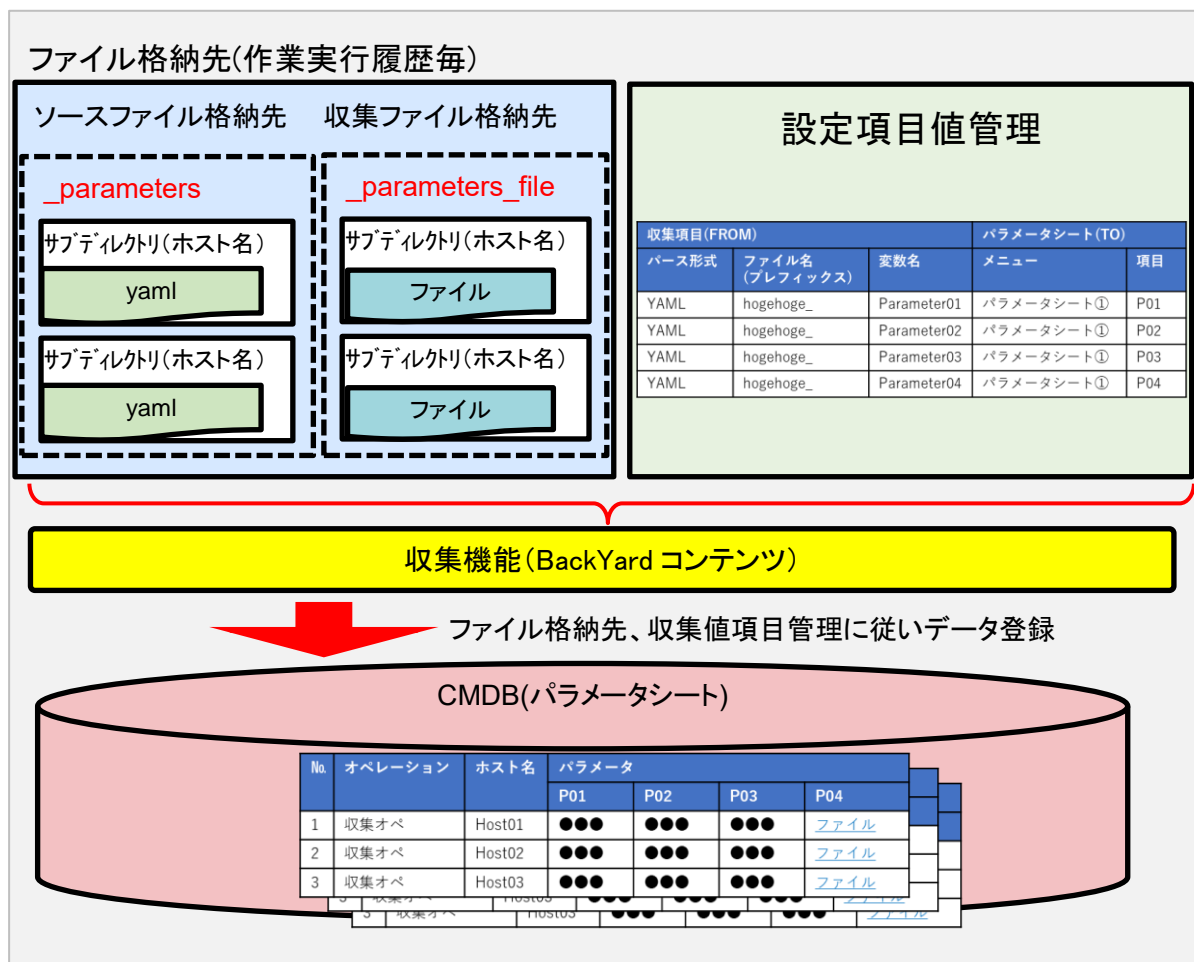
1.1.1 収集機能概要図

以下は、収集機能実行までの全体の概要図です。



1.1.2 収集機能データ登録処理概要図

以下は、収集機能のデータ登録処理の概要図です。



1.2 パラメータシートへの登録方法について

収集機能は、ITA のオプションとして機能し、ITA の標準 RESTAPI 機能を利用してパラメータシートへの登録処理を実施します。

RESTAPI 機能の詳細については、「利用手順マニュアル_RestAPI」を参照してください。

1.2.3 収集機能の動作要件

ITA で以下の設定がされていること。

- ・ ITA インストーラにて、「createparam」、「ansible_driver」が選択され、インストールされている
- ・ メニュー定義/作成にて、パラメータシート(ホスト/オペレーションあり)が作成されている
- ・ 設定値項目管理にて、登録情報(ソースファイル)とパラメータシートの項目と紐づけ設定がされている
- ・ 収集インターフェース情報の REST アクセス情報を更新済み
- ・ 収集対象機器(ホスト名)が、機器一覧に登録済み

作業実行後に、以下の状態である場合、パラメータシートへの登録を実施します。

- ・ 作業実行の結果、正常に完了している
- ・ 作業実行の出力結果として、規定の構造でディレクトリ、ファイルが配置されている

※パラメータシートへの登録元となるソースファイルを生成する IaC(Playbook、Role)については、各ユーザー様で準備する必要があります。

参考：Ansible Playbook Collection(OS 設定収集)

<https://github.com/exastro-suite/playbook-collection-docs/blob/master/README.ja.md>

2 収集機能でのディレクトリ、ファイル構造、変数取り扱い

2.1 収集対象ディレクトリ、ファイル構造

2.1.1 収集対象のファイルフォーマット

(1) Yaml 形式で出力されたファイル

```
e.g.)
■ファイル名: RH_snmp.yml
■ファイルの内容:
VAR_RH_sshd_config:
  - key: PermitRootLogin
    value: yes
  - key: PasswordAuthentication
    value: no
```

2.1.2 収集対象ディレクトリ構造

収集対象ディレクトリについて、収集対象ディレクトリパス（ソースファイルの出力先として）を IaC(Playbook,Role)内にて、で以下の変数として扱えます。

表 2-1 収集機能対象ディレクトリ ITA 独自変数

ITA 独自変数	変数指定内容	備考
__parameter_dir__	作業結果ディレクトリ配下の「_parameters」のパス	
__parameters_file_dir__	作業結果ディレクトリ配下の「_parameters_file」のパス	
__parameters_dir_for_epc__	作業ディレクトリ配下の「_parameters」のパス	
__parameters_file_dir_for_epc__	作業ディレクトリ配下の「_parameters_file」のパス	

収集対象のディレクトリ(_parameters)の上位ディレクトリのパスについて、「Ansible 共通」-「インターフェース情報」の「データリストレージパス(Ansible)」、Ansible-Driver の実行モード、実施した作業 No に依存します。

表 2-2 収集機能対象ディレクトリ、ファイル階層

階層構造	備考
【上位ディレクトリ】	※1 収集対象ディレクトリ(名称固定)
- _parameters	※2 ホスト名
- localhost	(機器一覧に登録されているものが収集対象)
- SAMPLE.yml	※3 収集対象ファイル
- _parameters_file	※4 収集対象ディレクトリファイルアップロード用(名称固定)
- localhost	※5 ファイルアップロード対象ファイル
- test.txt	

※データリストレージパス(Ansible)以降の階層構造

ソースファイルを生成する Playbook を作成する際、の出力先について、「表 2-1 収集機能対象ディレクトリ ITA 独自変数」を使用しない場合、以下の構造を認識して Playbook を記述する必要があります。

表 2-3 Ansible-Driver モード別上位ディレクトリパス

モード	モード別 識別子	階層構造	備考
Ansible-Legacy	legacy/ns/	/データリレイストレージパス(Ansible)/legacy/ns/	
Ansible-Pioneer	pioneer/ns/	/データリレイストレージパス(Ansible)/pioneer /ns/	
Ansible-LegacyRole	legacy/rl/	/データリレイストレージパス(Ansible)/legacy/rl/	

e.g.) 収集対象ファイルのファイルパス、ディレクトリ構造

実行モード: Ansible-Legacy

作業 No : 1

対象ホスト: localhost

作業実行ディレクトリ; /データリレイストレージパス(Ansible)/legacy/ns/0000000001/in/

作業結果ディレクトリ; /データリレイストレージパス(Ansible)/legacy/ns/0000000001/out/

収集対象ファイルパス、ディレクトリ構造:

/データリレイストレージパス(Ansible)/legacy/ns/0000000001/in/_parameters/localhost/SAMPLE.yml

/データリレイストレージパス(Ansible)/legacy/ns/0000000001/in/_parameters/localhost/OS/RH_snmpd.yml

/データリレイストレージパス(Ansible)/legacy/ns/0000000001/in/_parameters_file/localhost/TEST.txt

もしくは、

/データリレイストレージパス(Ansible)/legacy/ns/0000000001/out/_parameters/localhost/SAMPLE.yml

/データリレイストレージパス(Ansible)/legacy/ns/0000000001/out/_parameters/localhost/OS/RH_snmpd.yml

/データリレイストレージパス(Ansible)/legacy/ns/0000000001/out/_parameters_file/localhost/TEST.txt

ファイルアップロードのメニューを収集対象とする場合、ソースファイルの変数の値(ファイル名)と同名のファイルが_parameters_file 配下に配置されている必要があります。

収集項目値管理の設定は、「5.1.2 収集項目値管理」参照してください。

ファイルアップロードの最大ファイルサイズについて、サーバースペックに依存するため、詳細は、「利用手順マニュアル_RestAPI」を参照してください

e.g.) 通常変数の構造の変数の場合のディレクトリ構造とソースファイルの内容

■構造

【上位ディレクトリ】

```
|- _parameters
|   |- localhost
|       |- SAMPLE.yml    ※ソースファイル
|- _parameters_file
|   |- localhost
|       |- test.txt        ※アップロード対象ファイル
```

■収集対象ファイル名: SAMPLE.yml

■ファイルの内容

VAR_upload_file: test.txt

2.2 取り扱う変数と種類

収集機能で扱うソースファイル内で扱える変数は以下の 3 種類があります。

表 2.1 変数の種類

種類	内容	備考
通常変数	変数名に対して具体値を 1 つ定義できる変数です。 e.g.) VAR_users: root	
複数具体値変数	変数名に対して具体値を複数定義できる変数です。 e.g.) VAR_users: - root - mysql	
多段変数	階層化された変数です。 e.g.) VAR_users: - user-name: alice authorized: password } メンバー変数 メンバー変数名は、下記の 7 文字を除く ascii 文字 (0x20～0x7e)が使用出来ます。 " . [] ' ¥ : 尚、コーテーションで囲まないと変数名の先頭に使用出来ない文字がいくつかあります。詳しくは、Ansible ドキュメント Yaml syntax を参照下さい。	

3 収集機能 コンソールメニュー構成

本章では、ITA コンソールのメニュー構成について説明します
なお、Web コンソールへのログイン方法、およびメニュー画面の構成要素/基本的な操作については、「[ファーストステップガイド](#)」を参照してください。

3.1 メニュー/画面一覧

① Ansible 共通コンソールのメニュー

Ansible 共通コンソールのメニュー一覧を以下に記述します。

表 3-1 共通コンソール メニュー/画面一覧

No	メニューグループ	メニュー・画面	説明
1	Ansible 共通 コンソール	収集インターフェース情報	パラメータシートへのデータの登録時に使用する、ITA 標準 REST 機能にアクセスするサーバへの接続インターフェース情報を管理します。
2		設定項目値管理	作業実行の出力結果(ソースファイル)と、パラメータシートの項目の紐づけ設定を行い、収集機能で登録する対象パラメータを管理します。

② Ansible コンソールのメニュー

各 Ansible コンソールに対応するメニュー一覧を以下に記述します。

表 3-2 Ansible driver コンソール メニュー/画面一覧

No	メニューグループ			メニュー・画面	説明
	Ansible コンソール				
	Legacy	Legacy Role	Pioneer		
14	○	○	○	作業管理	作業実行履歴を管理します 収集機能によるパラメータシートの登録状況、実行ログを参照します。

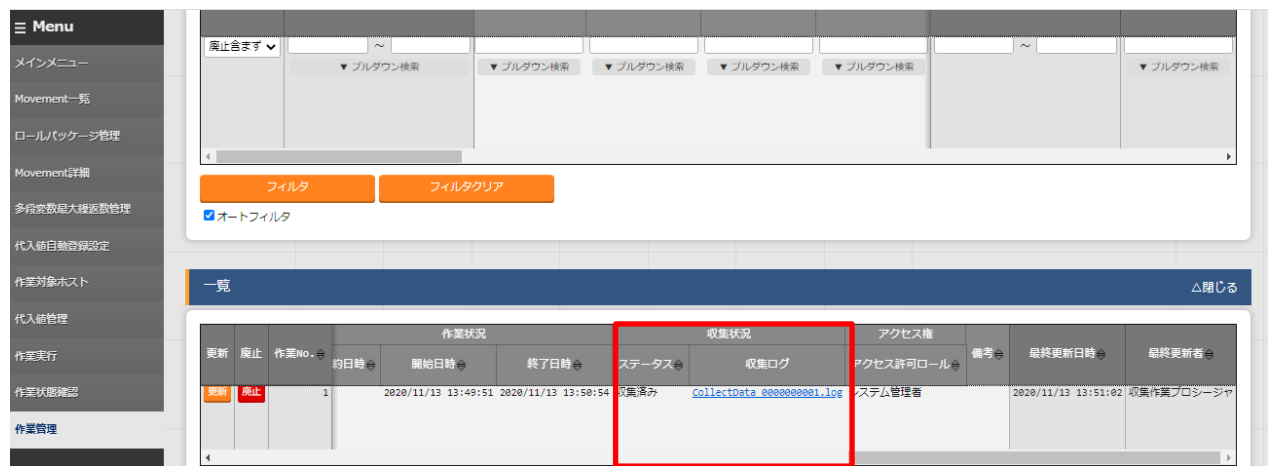


図 3.1-1 作業管理画面

4 収集機能の利用手順

収集機能の利用手順について説明します

4.1 作業フロー

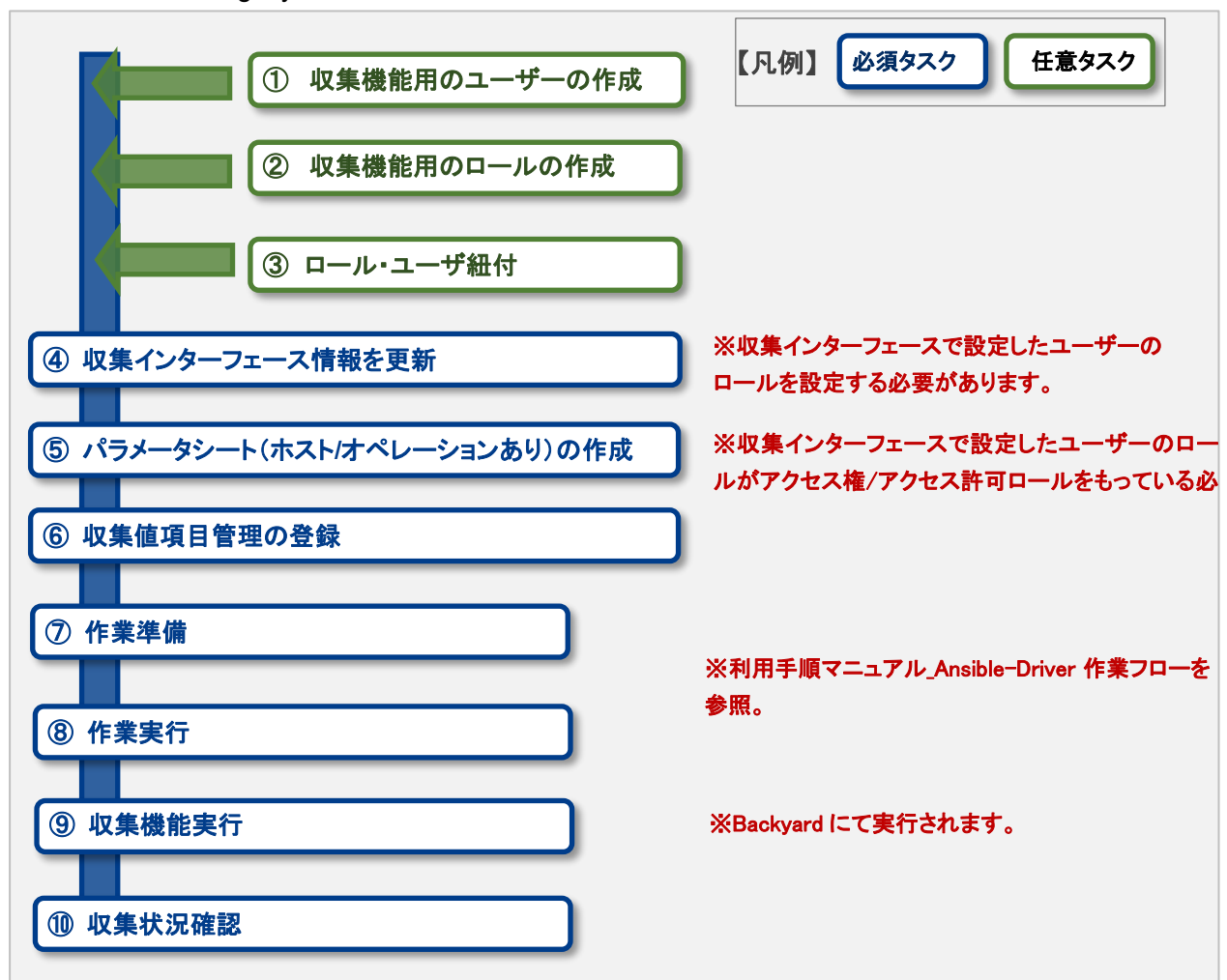
収集機能の実施における標準的なフローは以下のとおりです。

ITA Ansible-Driver の利用方法は、「利用手順マニュアル_Ansible-Driver」を参照してください。

ITA 基本コンソールの利用方法は、「利用手順マニュアル_基本コンソール」を参照してください。

4.1.1 収集機能作業フロー

以下は、Ansible-Legacy で作業を実行するまでの流れです。



● 作業フロー詳細と参照先

① 収集機能用のユーザーの作成

ITA 管理コンソールの機器一覧の画面から、収集機能用のユーザーを登録します。
登録方法は「利用手順マニュアル_管理コンソール」を参照してください。

② 収集機能用のロールの作成

ITA 管理コンソールのロール一覧の画面から、収集機能用のロールを登録します。
登録方法は「利用手順マニュアル_管理コンソール」を参照してください。

③ ロール・ユーザーの紐づけ

ITA 管理コンソールのロール・ユーザー紐付の画面から、ロールの紐づけを行います。
登録方法は「利用手順マニュアル_管理コンソール」を参照してください。

④ 収集インターフェース情報の登録

Ansible 共通コンソールの収集インターフェース情報の画面から、接続情報を登録します。
詳細は「5.1.1 収集インターフェース情報」を参照してください。

⑤ パラメータシート(ホスト/オペレーションあり)の作成

メニュー作成コンソールのメニュー定義/作成の画面からのパラメータシートを作成します。
詳細は「利用手順マニュアル_メニュー作成機能」を参照してください。

⑥ 収集項目値管理の登録

Ansible 共通コンソールの収集項目値管理の画面から、ソースファイルとパラメータシートの項目の紐付情報の登録をします。
詳細は「5.1.2 収集項目値管理」を参照してください。

⑦ 作業準備

作業実行のための準備を行います。。
詳細は、「利用手順マニュアル_Ansible-Driver」、「利用手順マニュアル_Symphony」、「利用手順マニュアル_Conductor」を参照してください。

⑧ 作業実行

実行日時、投入オペレーション、Movement、ワークフローを選択し処理の実行を指示します。
実行について「利用手順マニュアル_Ansible-Driver」、「利用手順マニュアル_Symphony」、「利用手順マニュアル_Conductor」を参照してください。

⑨ 収集機能実行

作業実行が完了した作業 No を収集機能の対象として、パラメータシートへの登録処理を実施します。
詳細は「5.3 BackYard コンテンツ」を参照してください。

⑩ 収集状況確認

Ansible-Legacy/Ansible-Pioneer/Ansible-LegacyRole コンソールの作業管理の画面では、完了した作業の収集状態の確認、ログファイルが DL 可能です。
詳細は「5.2.1 収集状況の確認」を参照してください。

5 収集機能・操作方法説明

本章では、収集機能で利用する各コンソールの機能について説明します。

登録方法の詳細は、関連マニュアルの「利用手順マニュアル_基本コンソール」をご参照下さい。

5.1 Ansible 共通コンソール

本節では、Ansible 共通コンソールでの操作について記載します。

5.1.1 収集インターフェース情報

- (1) [収集インターフェース情報]では、収集機能で利用する ITA の標準 RESTAPI を利用する為、RESTAPI アクセス用の接続インターフェース情報の更新を行います。

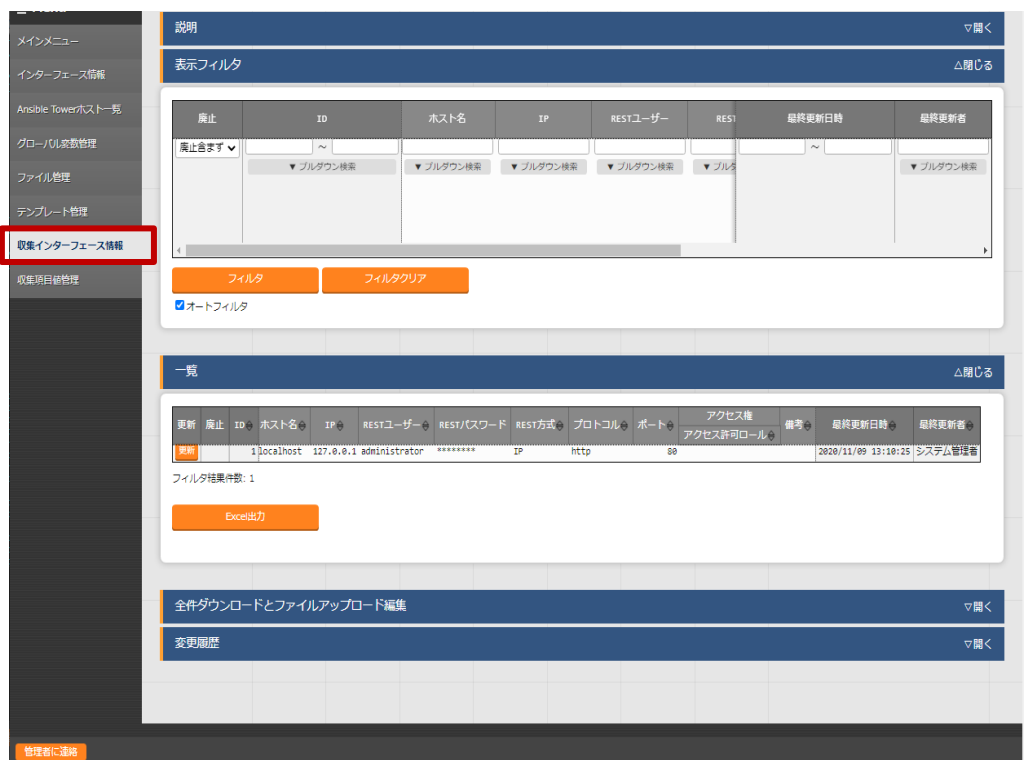


図 5.1-1 サブメニュー画面(収集インターフェース情報)

- (2) 「一覧」-「更新」ボタンより、収集インターフェース情報の登録を行います。



図 5.1-2 更新画面(収集インターフェース情報)

- (3) 収集インターフェース情報画面の項目一覧は以下のとおりです。
 収集インターフェース情報が未登録または、複数レコード登録されている状態で作業実行した場合、
収集機能によるパラメータシートへの登録はされません。

表 5.1-1 登録画面項目一覧(インタフェース情報)

項目	説明	入力 必須	入力形式	制約事項
ホスト名	使用するホスト名を入力 初期値:localhost	○	手動入力	
IP	使用する IP を入力 初期値:127.0.0.1	○	手動入力	
REST ユーザー	使用する ITA ユーザーのログイン ID を入力		手動入力	※1
REST パスワード	使用する ITA ユーザーのログインパスワードを入力		手動入力	
REST 方式	ホスト名、IP のどちらか使用するものを選択 <ul style="list-style-type: none"> ● IP ● ホスト名 	○	リスト選択	
プロトコル	使用するプロトコルを入力 初期値:http	○	手動入力	
ポート	使用するポートを入力 初期値:80	○	手動入力	
備考	自由記述欄です。	-	手動入力	

※1 REST ユーザーに入力されたユーザーは、以下が必須になります。

- ・ ユーザーが所属しているロールが、作成されたパラメータシートのメニューのアクセス権/アクセス許可ロールをもっている。
- ・ メニューのロール情報にて、ユーザーに紐づくロールがメンテナンス可である。

ユーザー、ロールの作成、紐付、について「利用手順マニュアル_管理コンソール」を参照してください。

5.1.2 収集項目値管理

(1) [収集項目値管理]では、収集項目とパラメータシートの項目の紐付設定を行います。

図 5.1-3 サブメニュー画面(収集項目値管理)

(2) 「一覧」-「登録開始」ボタンより、収集項目の登録を行います。

図 5.1-4 登録画面(収集項目値管理)

(3) 収集項目値管理画面の項目一覧は以下のとおりです。

表 5.1-1 登録画面項目一覧(収集項目値管理)

項目	説明	入力 必須	入力形式	制約事項
収集項目 (FROM)	パース形式	○	リスト選択	
	PREFIX(ファイル名)	○	手動入力	※1
	変数名	○	手動入力	※1
	メンバー変数		手動入力	※1
パラメータ シート (TO)	メニューグループ	○	リスト選択	
	メニュー			

項目		説明	入力 必須	入力形式	制約事項
	項目	項目を選択	○	リスト選択	

※1 ファイル名、変数、メンバー変数入力値の例

e.g.) 通常変数の構造の変数の場合

■ファイル名: SAMPLE.yml

■ファイルの内容

VAR_sample_config_1: yes

VAR_sample_config_2: test_parameter

■収集値項目管理の収集項目(FROM)の入力可能な値

PREFIX(ファイル名): SAMPLE

変数名: VAR_sample_config_1

VAR_sample_config_2

e.g.) 複数具体値の構造の変数の場合

■ファイル名: SAMPLE_2.yml

■ファイルの内容

VAR_sample2_conf:

SAMPLE1

SAMPLE2

SAMPLE3

■収集値項目管理の収集項目(FROM)の入力可能な値

PREFIX(ファイル名): SAMPLE_2

変数名: VAR_sample2_conf

メンバー変数: [0]

[1]

[2]

e.g.) 複数具体値の構造の変数の場合

■ファイル名: RH_sshd.yml

■ファイルの内容

VAR_RH_sshd_config:

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

■収集値項目管理の収集項目(FROM)の入力可能な値

PREFIX(ファイル名): RH_sshd

変数名: VAR_RH_sshd_config:

メンバー変数: [0].key
[0].value
[1].key
[1].value

e.g.) 複数具体値の構造の変数の場合 2

■ファイル名: RH_snmp.yml

■ファイルの内容

VAR_RH_snmpd_info:

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

■収集値項目管理の収集項目(FROM)の入力可能な値

PREFIX(ファイル名): RH_snmp

変数名: VAR_RH_snmp_config:

メンバー変数: 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 コンソール

5.2.1 収集状況の確認

各コンソール (Ansible-Legacy/Ansible-Pioneer/Ansible-LegacyRole) の作業管理の画面では、完了した作業の収集状態の確認、ログファイルが DL 可能です。

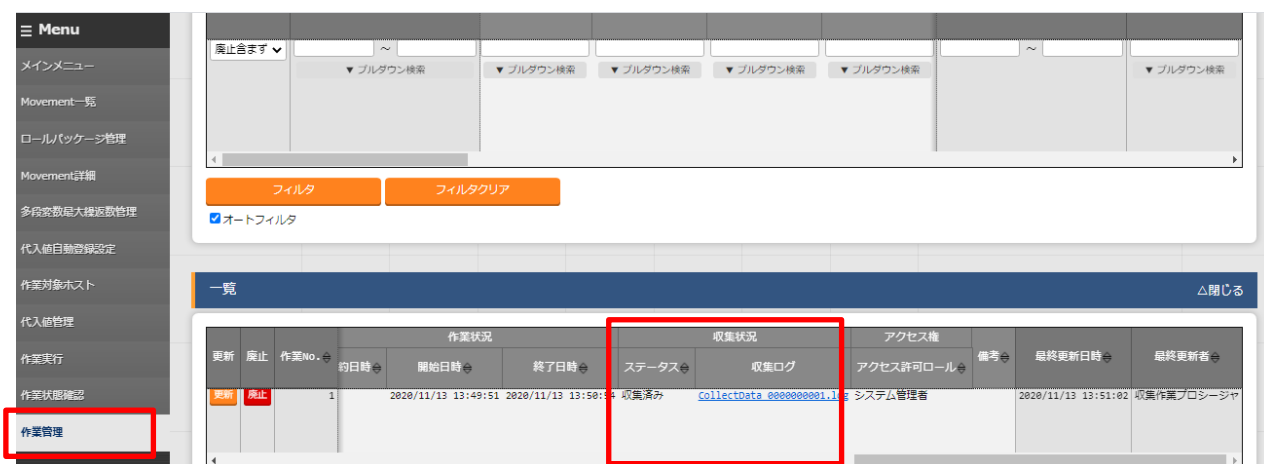


図 5.2-1 作業管理画面

表 5.2-1 作業管理画面収集状況詳細

項目	説明	備考
ステータス	収集機能の実行状況の表示 対象外: 収集機能対象外 (対象ファイルなし) 収集済み: 収集機能実施済み 収集済み (通知あり): 登録/更新中に不備があった場合	※
収集ログ	収集機能実行のログをダウンロード	

表 5.2-2 収集状況詳細

作業状態		収集機能対象	収集状況		備考
ステータス	対象ファイル		ステータス	収集ログ	
完了以外	なし	対象外	空	空	
完了以外	あり	対象外	空	空	
完了	なし	対象	対象外	空	
完了	あり	対象	収集済み	ログファイルあり	
完了	あり	対象	収集済み (通知あり)	ログファイルあり	

※ステータスの表記について

- 作業状態が完了でない場合、収集機能対象外の為、収集状況は更新されないため、空のままとなります。
- 作業状態が完了で、収集対象ファイルが存在しない場合、ステータスは収集済み、収集ログは空の状態となります。

- 収集インターフェース情報、設定項目値管理、メニューのアクセス権/アクセス許可ロールの不備により、RESTAPI の登録処理が失敗した場合でも収集済み(通知あり)となります。詳細は、収集ログ内を参照。

ログファイル出力内容例

e.g.) ログファイル出力内容例 (登録処理成功)

2020-11-13 13:51:02 Collect START (ホスト名:ita-sample ファイル名:RH_snmpd)

2020-11-13 13:51:02 REST DATA (ホスト名: ita-sample メニューID: 0000000004 オペレーション NO: 1)

Array

(

[0] => http://127.0.0.1:80/default/menu/07_rest_api_ver1.php?no=0000000004

[1] => [{"更新":"","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_20201111115557819037",""}]

[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¥u6d3b","ct":0},"error":{"name":"¥u30a8¥u30e9¥u30fc","ct":0},"RAW":["000","200",""]}}}}

)

2020-11-13 13:51:02 Collect END (ホスト名:ita-sample ファイル名:RH_snmpd)

e.g.) ログファイル出力内容例 (登録処理失敗)

2020-11-06 13:32:52 Collect START (ホスト名:ita-sample ファイル名:RH_snmpd)

2020-11-06 13:32:52 [処理]REST アクセスに失敗しました。

Array

(

[0] => http://127.0.0.1:80/default/menu/07_rest_api_ver1.php?no=0000000005

[1] => [{"登録":"","","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 (ホスト名:ita-sample ファイル名:RH_snmpd)

e.g.) ログファイル出力内容例 (対象外)

2020-11-05 16:55:31 [処理]対象機器が登録されていないか、廃止されているため、登録、更新処理をスキップ (ホスト名:ita-test)

5.3 BackYard コンテンツ

5.3.1 パラメータシートへの登録処理の概要

- (1) 収集インターフェース情報の取得
- (2) 正常に完了した作業の一覧を取得
収集対象ステータス: 完了
- (3) 収集対象作業 No から以下の情報を取得
 - オペレーション情報
 - 対象ホスト
 - 対象ソースファイル
- (4) 対象のホストが機器一覧に登録されているか問い合わせし
登録: 収集対象
未登録: 対象外
- (5) 対象ソースファイルと収集項目値管理から対象パラメータシートのメニューID を取得
- (6) (1)～(4)の情報から RESTAPI のパラメータを生成

対象のメニューID に対して、データの問い合わせを実施し、RESTAPI の実行種別を判定

登録: 一意のオペレーション、ホスト組み合わせのデータが登録されていない

更新: 一意のオペレーション、ホスト組み合わせのデータが登録されている

- (7) ITA 標準 RESTAPI 機能にて、データの登録/更新を実施します。
- (8) 作業 No に収集状況のステータスを更新

なお、パラメータシートへのデータ登録のタイミングは自動プロセスの起動周期に依存します。
起動周期の変更については、後述「[6.2 メンテナンス方法について](#)」を参照してください。

登録・更新されたレコードのアクセス権/アクセス許可ロールは、収集対象の作業結果のアクセス権/アクセス許可ロールを継承します。

対象の作業結果については、別紙「Exastro-ITA_利用手順マニュアル_Ansible-driver」を参照してください。

6 運用操作

本機能を活用する操作は、クライアント PC のブラウザ画面からのユーザー利用による入力だけでなく、システム運用・保守による操作もあります。用意している運用・保守の操作は次のとおりです。

6.1 メンテナンス

収集機能のプロセスの開始/停止/再起動に必要なファイルは以下となります。

説明	対象ファイル名
パラメータ自動登録 作業実行が完了した、作業結果から、設定項目値管理に登録した情報を元に、パラメータシートへの登録を行う。	ky_std_synchronize-Collector.service

対象ファイルは「/usr/lib/systemd/system」に格納されています。
プロセス起動/停止/再起動の方法は次の通りです。
root 権限でコマンドを実行してください。

① プロセス起動

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

① プロセス停止

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

② プロセス再起動

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

各対象ファイル名に置き換えて起動/停止/再起動を行ってください。

6.2 メンテナンス方法について

① NORMAL レベルへの変更

以下のファイルの 8 行目「DEBUG」を「NORMAL」に書き換えます。

ログレベル設定ファイル: <インストールディレクトリ>/ita-root/conf/backyardconfs/ita_env

② DEBUG レベルへの変更

以下のファイルの 8 行目「NORMAL」を「DEBUG」に書き換えます。

ログレベル設定ファイル: <インストールディレクトリ>/ita-root/conf/backyardconfs/ita_env

③ 起動周期の変更

各対象ファイルの ExecStart の 5 番目のパラメータを変更します。(単位:秒)

例外を除き起動周期はデフォルト値の使用をしてください。

```
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
```

書き換え後、プロセス再起動(restart)後に有効になります。

ログファイルの出力先: <インストールディレクトリ>/ita-root/logs/backyardlogs

7 付録

7.1 参考

以下、IaC(Playbook、Role)の参考例となります。

1. Ansible Playbook Collection (OS 設定収集)
<https://github.com/exastro-suite/playbook-collection-docs/blob/master/README.ja.md>
2. Ansible コンフィグ取得、パラメータ生成 Playbook

makeYml_Ansible.yml

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

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

※ makeYml_Ansible.yml 実行して、収集対象のソースファイル(yaml)を生成する場合、gather_facts を有効にする必要があります。
「Ansible-Legacy」-「Movement 一覧」編集時、「ヘッダーセクション」に以下を記載する。
設定変更については、「利用手順マニュアル Ansible-driver」を参照してください。

e.g) gather_facts 有効設定例

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