



# IT Automation Conductor 【実習編】

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

Exastro IT Automation Version 1.9  
Exastro developer

# 目次

## 1. はじめに

### 1. 本書について

## 2. Conductor

### 1. シナリオ

### 2. 事前準備

## 3. 実習

### 1. 作業対象ホストの登録

### 2. オペレーションの登録

### 3. IaCの登録

### 4. Movementの登録

### 5. Movement-Playbook紐付の登録

### 6. オペレーションに関連付くMovementとホストの登録

### 7. 代入値管理

### 8. Conductorの登録


### 9. Conductorの実行

### 10. Conductorの完了確認

# 1. はじめに

# 1.1 本書について

- 本書では、メニューグループの「**Conductor**」について解説しています。

 **Exastro**  
IT Automation

管理コンソール

ようこそ[システム管理者]さん  
ログインID [administrator]

[ロール](#) [パスワード変更](#) [ログアウト](#)

Menu

DASHBOARD

編集

メインメニュー

システム設定

メニューグループ管理

メニュー管理

ロール管理

ユーザ管理

ロール・メニュー紐付管理

ロール・ユーザ紐付管理

シーケンス管理

SSO基本情報管理

SSO属性情報管理


バージョン確認

メニューグループ


 管理コンソ...


 基本コンソ...


 エクスポー...


 Symphony


 Conductor


 メニュー作成


 入力用


 代入値自動...

 参照用

 比較


 ホストグル...


 Ansible共通

 Ansible-Le...

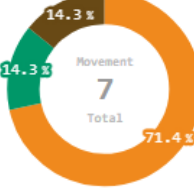
 Ansible-Pi...

 Ansible-Le...

 Terraform

 CI/CD for ...

Movement



Movement	7
Total	

Movement	SUM
Ansible Legacy	5
Ansible Pioneer	1
Ansible Legacy Role	1
Terraform	0

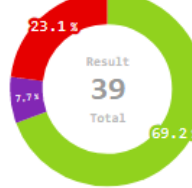
作業状況



Status	0
Total	

Status	CON	SYM	SUM
実行中	0	0	0
未実行(予約)	0	0	0
未実行	0	0	0

作業結果



Result	39
Total	

Result	CON	SYM	SUM
正常終了	27	0	27
異常終了	3	0	3
想定外エラー	9	0	9
緊急停止	0	0	0
予約取消	0	0	0

作業履歴

## 2. Conductorについての説明

## 2.1 シナリオ (1/2)

- 本シナリオは以下の流れとなります。
- Ansible driverはAnsible-Legacyを使用しています。

①機器情報の登録

基本コンソールメニュー

②オペレーションの登録

③Movementの登録

各種Driverメニュー

④Movementの確認

⑤インターフェース情報を登録

Conductorメニュー

⑥Conductorの登録

⑦Conductorの確認

⑧Conductorの実行

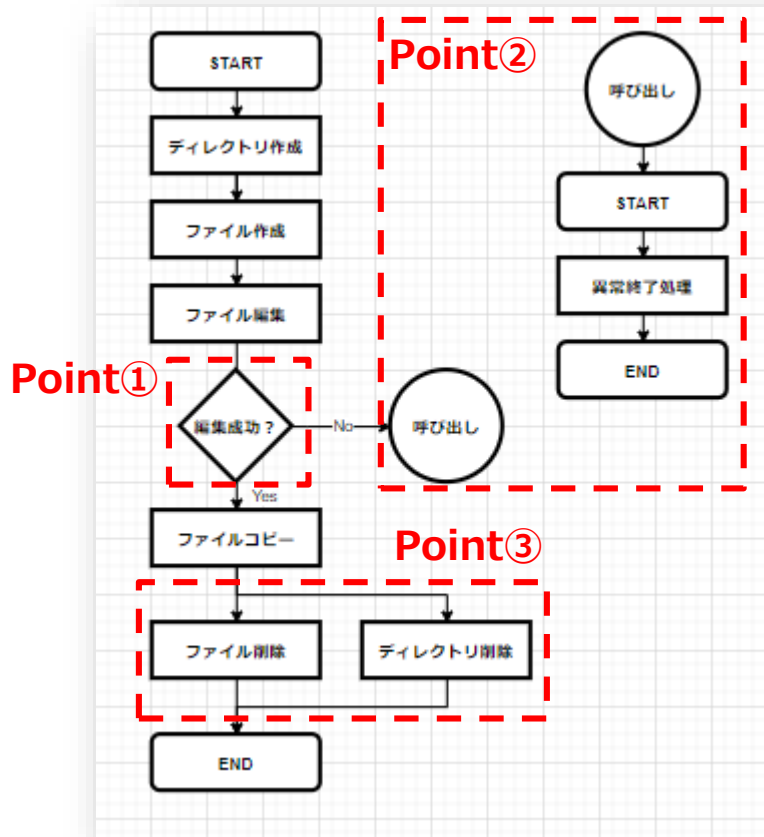
⑨実行結果確認

⑩実行履歴の確認

## 2.1 シナリオ (2/2)

- 本編ではConductor機能を体感いただくために、以下のフローチャートと同様のConductorを作成します。

- フローチャート



### □Conductor機能の特徴

#### □Point①

前処理の成功/終了判断による条件分岐機能

#### □Point②

登録済のOperation/Conductorの呼び出し機能

#### □Point③

Movementの並行処理機能

## 2.2 事前準備

### ●IaCの作成(1/2)

本シナリオでは、Ansible-Legacyを例にご説明します。

下記のIaCをモジュールごとにymlファイルとして保存してください。

※文字コードは"UTF-8"、改行コードは"LF"、拡張子は"yml"形式。  
また、インデントにご注意下さい。

```
- name: create directory
  file:
    path: /tmp/{{ item.dir }}
    state: directory
    mode: 0755
  with_items:
    - { dir: "{{ VAR_dir_name_1 }}" }
    - { dir: "{{ VAR_dir_name_2 }}" }

- name: remove directory
  file:
    path=/tmp/{{ VAR_dir_name_1 }}
    state=absent

- name: create file
  file:
    path=/tmp/{{ VAR_dir_name_1 }}/{{ VAR_file_name }}
    state=touch
    mode=0755
```



## 2.2 事前準備

### ●IaCの作成(2/2)

以下も同様にモジュールごとにymlファイルとして保存してください。

- name: remove file  
file:  
  path=/tmp/{{ VAR\_dir\_name\_1 }}/{{ VAR\_file\_name }}  
  state=absent
- name: copy file  
copy:  
  src: /tmp/{{ VAR\_dir\_name\_1 }}/{{ VAR\_file\_name }}  
  dest: /tmp/{{ VAR\_dir\_name\_2 }}/{{ VAR\_file\_name }}  
  owner: root  
  group: root  
  mode: 0644  
  remote\_src: yes
- name: edit file  
copy:  
  dest: /tmp/{{ VAR\_dir\_name\_1 }}/{{ VAR\_file\_name }}  
  content: "{{ VAR\_edit\_param\_1 }}"
- name: forced termination  
fail: msg={{ VAR\_message\_text }}

#### ・作成後イメージ

名前	更新日時	種類	サイズ
 copy_file.yml	2021/09/01 16:57	YML ファイル	1 KB
 create_directory.yml	2021/09/01 16:58	YML ファイル	1 KB
 create_file.yml	2021/09/01 16:58	YML ファイル	1 KB
 edit_file.yml	2021/09/01 16:58	YML ファイル	1 KB
 forced_termination.yml	2021/09/01 16:58	YML ファイル	1 KB
 remove_directory.yml	2021/09/01 16:58	YML ファイル	1 KB
 remove_file.yml	2021/09/01 16:58	YML ファイル	1 KB

### 3. 実習

## 3.1 作業対象ホストの登録

### ●作業対象ホストの登録

「基本コンソール」メニューグループ >> 「機器一覧」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「ホスト名」「IPアドレス」「ログインユーザID」「管理」「ログインパスワード」「認証方式」を入力
- ② 「登録」ボタンを押下

1

項目へ値を入力する

項目	値
ホスト名	testserver
IPアドレス	(任意の値)
ログインユーザID	(任意の値)
管理	●
ログインパスワード	(任意の値)
認証方式	パスワード認証

#### POINT

本シナリオでは、作業対象ホストにsshのパスワード接続を行う場合を想定しています。  
「IPアドレス」「ログインユーザID」「ログインパスワード」については  
ユーザ様のご利用環境に適した設定をご入力ください。

## 3.2 オペレーションの登録

### ●オペレーションの登録

「基本コンソール」メニューグループ >> 「オペレーション一覧」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「オペレーション名」「実施予定日時」を入力
- ② 「登録」ボタンを押下

登録

△開じる

No.	オペレーションID	オペレーション名*	実施予定日時*	アクセス権	設定
自動入力	自動入力			アクセス許可ロール	

※\*は必須項目です。

戻る 登録

1 項目へ値を入力する

項目	値
オペレーション名	operation1
実施予定日時	(任意の日時)

全件ダウンロードとファイルアップロード編集 ▽開く

変更履歴 ▽開く

#### POINT

ここで指定した日時に  
処理が実行されるわけではありません

## 3.3 IaCの登録 (1/2)

### ●IaCの登録

「Ansible-Legacy」メニューグループ >> 「Playbook素材集」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「Playbook素材名」を入力、  
「Playbook素材」欄の「ファイルを選択」ボタンを押下し  
**事前に作成したyamlファイルをすべてをアップロード**  
(「事前アップロード」ボタン押下)
- ② 「登録」ボタンを押下

ようこそ[システム管理者]さん  
ログインID [administrator]  
パスワード変更 ログアウト

Menu  
メインメニュー  
Movement一覧  
Playbook素材集  
Movement-Playbook追加  
代入値自動登録設定  
作業対象ホスト  
代入値管理  
作業実行  
作業状態確認  
作業管理

説明  
表示フィルタ  
一覧/更新  
登録

Playbook素材名  
Playbook素材  
備考  
最終更新日時  
最終更新者

自動入力  
ファイルを選択 選択されていません  
事前アップロード  
アップロード状況:

\*は必須項目です。

戻る 登録

全件ダウンロードとファイルアップロード編集

項目へ値を入力する

項目	値
Playbook素材名	<任意>
Playbook素材	<任意> .yaml

POINT

IaCの作成手順につきましては、  
「2.2事前準備」をご参照下さい

## 3.3 IaCの登録 (2/2)

### ●IaCの登録

作成後のイメージは以下ようになります。

一覧/更新

履歴	複製	更新	廃止	素材ID	Playbook素材名	Playbook素材	Movement-Playbook紐付	アクセス権	備考	最終更新日時	最終更新者
								アクセス許可ロール			
履歴	複製	更新	廃止	1	copy_file	<a href="#">copy_file.yml</a>	Movement-Playbook紐付			2021/09/09 14:37:04	システム管理者
履歴	複製	更新	廃止	2	create_directory	<a href="#">create_directory.yml</a>	Movement-Playbook紐付			2021/09/09 14:37:10	システム管理者
履歴	複製	更新	廃止	3	create_file	<a href="#">create_file.yml</a>	Movement-Playbook紐付			2021/09/09 14:37:16	システム管理者
履歴	複製	更新	廃止	4	edit_file	<a href="#">edit_file.yml</a>	Movement-Playbook紐付			2021/09/09 14:37:24	システム管理者
履歴	複製	更新	廃止	5	forced_termination	<a href="#">forced_termination.yml</a>	Movement-Playbook紐付			2021/09/09 14:37:34	システム管理者
履歴	複製	更新	廃止	6	remove_directory	<a href="#">remove_directory.yml</a>	Movement-Playbook紐付			2021/09/09 14:37:41	システム管理者
履歴	複製	更新	廃止	7	remove_file	<a href="#">remove_file.yml</a>	Movement-Playbook紐付			2021/09/09 14:36:56	システム管理者

フィルタ結果件数: 7

Excel出力

## 3.4 Movementの登録 (1/2)

### ●Movementの登録

「Ansible-Legacy」メニューグループ >> 「Movement一覧」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「Movement名」「ホスト指定形式」を入力
- ② 「登録」ボタンを押下

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu

メインメニュー

Movement一覧

Playbook素材集

Movement-Playbook紐付

代入値自動登録設定

作業対象ホスト

代入値管理

作業実行

作業状態確認

作業管理

説明

表示フィルタ

一覧/更新

登録

MovementID Movement名\* 遅延タイマー Ansible利用情報

自動入力

ホスト指定形式\*

WinRM接続

username

※\*は必須項目です。

戻る 登録

全件ダウンロードとファイルアップロード編集

変更履歴

項目	値
Movement名	<任意>
ホスト指定形式	IP

POINT 作成するMovementは ymlファイルと同数を作成して下さい

## 3.4 Movementの登録 (2/2)

### ●Movementの登録

登録後のイメージは以下ようになります。

一覧/更新									
履歴	複製	更新	廃止	MovementID	Movement名	オーケストレータ	遅延タイマー	ホスト指定形式	
								WinRM接続	
履歴	複製	更新	廃止	1	copy_file	Ansible Legacy		IP	
履歴	複製	更新	廃止	2	create_directory	Ansible Legacy		IP	
履歴	複製	更新	廃止	3	create_file	Ansible Legacy		IP	
履歴	複製	更新	廃止	4	edit_file	Ansible Legacy		IP	
履歴	複製	更新	廃止	5	forced_termination	Ansible Legacy		IP	
履歴	複製	更新	廃止	6	remove_directory	Ansible Legacy		IP	
履歴	複製	更新	廃止	7	remove_file	Ansible Legacy		IP	

フィルタ結果件数: 7

Excel出力



## 3.5 Movement-Playbook紐付の登録 (1/2)

### ●Movement-Playbook紐付の登録

「Ansible-Legacy」メニューグループ >> 「Movement-Playbook紐付」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「Movement」「Playbook素材」「インクルード順序」を入力
- ② 「登録」ボタンを押下

Exastro IT Automation Ansible-Legacy

Menu

- メインメニュー
- Movement一覧
- Playbook素材集
- Movement-Playbook紐付
- 代入値自動登録設定
- 作業対象ホスト
- 代入値管理
- 作業実行
- 作業状態確認
- 作業管理

説明

表示フィルタ

一覧/更新

登録

項目	値
Movement	作成した Movementを選択
Playbook素材	登録した Playbookを選択
インクルード順序	1

紐付項番

Movement

Playbook素材

インクルード順序

自動入力

設定

最終更新日時

最終更新者

自動入力

自動入力

※\*は必須項目です。

戻る

登録

POINT

登録するMovement詳細は  
yamlファイルと同数を作成して下さい

## 3.5 Movement-Playbook紐付の登録 (2/2)

### ●Movement-Playbook紐付の登録

登録後のイメージは以下になります。

一覧/更新

△閉じる

履歴	複製	更新	廃止	紐付項番	Movement	Playbook素材	インクルード順序	アクセス権	備考	最終更新日時	最終更新者
								アクセス許可ロール			
履歴	複製	更新	廃止	1	<a href="#">1:copy_file</a>	<a href="#">copy_file</a>	1			2021/09/08 15:49:19	システム管理者
履歴	複製	更新	廃止	2	<a href="#">2:create_directory</a>	<a href="#">create_directory</a>	1			2021/09/09 14:51:31	システム管理者
履歴	複製	更新	廃止	3	<a href="#">3:create_file</a>	<a href="#">create_file</a>	1			2021/09/09 14:51:39	システム管理者
履歴	複製	更新	廃止	4	<a href="#">4:edit_file</a>	<a href="#">edit_file</a>	1			2021/09/09 14:51:49	システム管理者
履歴	複製	更新	廃止	5	<a href="#">5:forced_termination</a>	<a href="#">forced_termination</a>	1			2021/09/09 14:51:58	システム管理者
履歴	複製	更新	廃止	6	<a href="#">6:remove_directory</a>	<a href="#">remove_directory</a>	1			2021/09/09 14:52:07	システム管理者
履歴	複製	更新	廃止	7	<a href="#">7:remove_file</a>	<a href="#">remove_file</a>	1			2021/09/09 14:52:15	システム管理者

フィルタ結果件数: 7

Excel出力

## 3.6 オペレーションに関連付くMovementとホストの登録

### ●オペレーションに関連付くMovementとホストの登録

「Ansible-Legacy」メニューグループ >> 「作業対象ホスト」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「オペレーション」「Movement」「ホスト」を入力
- ② 「登録」ボタンを押下

**Exastro IT Automation Ansible-Legacy**

Menu

- メインメニュー
- Movement一覧
- Playbook素材集
- Movement-Playbook紐付
- 代入値自動登録設定
- 作業対象ホスト
- 代入値管理
- 作業実行
- 作業状態確認
- 作業管理

説明

表示フィルタ

一覧/更新

登録

項目	値
オペレーション	operation1
Movement	作成したMovement
ホスト	testserver

項番

オペレーション\*

Movement\*

ホスト\*

自動入力

最終更新日時

最終更新者

自動入力

自動入力

※\*は必須項目です。

戻る

登録

POINT

作成したMovementは  
すべて登録を行ってください

## 3.7 代入値管理 (1/2)

### ●代入値管理

「Ansible-Legacy」メニューグループ >> 「代入値管理」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「オペレーション」「Movement」「ホスト」「変数名」「具体値」を入力
- ② 「登録」ボタンを押下

Exastro IT Automation Ansible-Legacy

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu

メインメニュー

Movement一覧

Playbook素材集

Movement-Playbook紐付

代入値自動登録設定

作業対象ホスト

代入値管理

作業実行

作業状態確認

作業管理

説明

表示フィルタ

一覧/更新

登録

項目	オペレーション*	Movement*	ホスト*	変数名*
自動入力	オペレーションを選択して下さい	オペレーションを選択して下さい	Movementを選択して下さい	OFF

※\*は必須項目です。

戻る 登録

項目へ値を設定する

項目	値
オペレーション	operation1
Movement	次頁参照
ホスト	testserver
変数名	次頁参照

## 3.7 代入値管理 (2/2)

### ●代入値管理

代入値の登録は以下を参考に行ってください。

オペレーション	ホスト	Movement : 変数	具体値	代入順序
1:operation1	1:Testserver	1:copy_file:1:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	1:copy_file:2:VAR_file_name	testfile	
1:operation1	1:Testserver	1:copy_file:3:VAR_dir_name_2	dir2	
1:operation1	1:Testserver	2:create_directory:4:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	2:create_directory:5:VAR_dir_name_2	dir2	
1:operation1	1:Testserver	3:create_file:6:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	3:create_file:7:VAR_file_name	testfile	
1:operation1	1:Testserver	4:edit_file:8:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	4:edit_file:9:VAR_file_name	testfile	
1:operation1	1:Testserver	4:edit_file:10:VAR_edit_param_1	param1	
1:operation1	1:Testserver	5:forced_termination:11:VAR_message_text	testmsg_fail	
1:operation1	1:Testserver	6:remove_directory:12:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	7:remove_file:13:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	7:remove_file:14:VAR_file_name_1	testfile	

## 3.8 Conductorの登録 (1/7)

### ●Conductorの登録

「Conductor」メニューグループ >> 「Conductorクラス編集」

- ① 「Conductor名称」を入力
- ② 画面右側に表示されている「Movement」「Function」を画面中央にドラッグ&ドロップ
- ③ 「登録」ボタンを押下

The screenshot displays the 'Conductor Class Edit' interface. On the left, a flowchart is built on a grid, starting with a 'Start' node, followed by a sequence of 'Ansible Legacy' tasks (create\_directory, create\_file, edit\_file), then a 'copy\_file' task, and finally a 'Conductor call' node. On the right, a configuration panel shows 'Conductor名称' (Conductor Name) set to 'Conductor\_01'. Below this, a table lists available movements and functions. A red arrow points from the 'Conductor call' node in the flowchart to the table, labeled 'ドラッグ&ドロップ' (Drag & Drop). At the bottom, a '登録' (Register) button is highlighted with a red circle and the number 3.

Conductor名称

ID : Auto numbering

Name : Conductor\_01

Permission role

1 項目へ値を入力する

項目	値
Name	任意の名称

2

ドラッグ&ドロップ

3

登録

※今回作成するConductorの登録手順は次ページ以降を参照してください

## 3.8 Conductorの登録 (2/7)

### ●Conductorの登録

以下のようにConductorを作成してください

項目へ値を入力する

項目	値
Name	Conductor_2

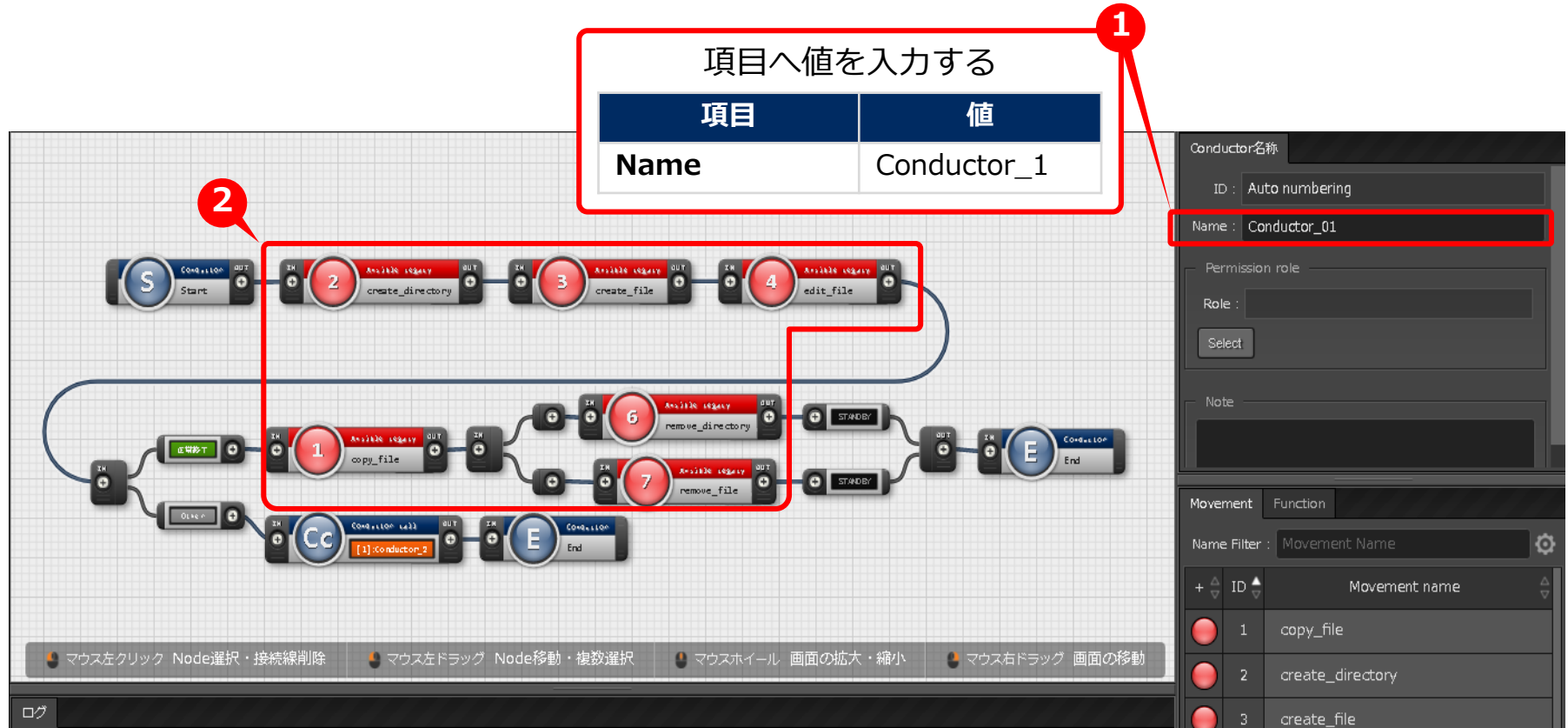
①Conductor Nameに「Conductor\_2」と入力  
②Movementから「forced\_termination」をドラッグアンドドロップ  
③図のように「OUT」と「IN」をつなぐ  
④画面下の「登録」を押下

登録

## 3.8 Conductorの登録 (3/7)

### ●Conductorの登録

「Conductor\_1」の全体図は以下のようになります。  
次ページ以降で細部を説明します。



- ①Conductor Nameに「Conductor\_1」と入力
- ②各種Movementは図を参考に配置してください。



## 3.8 Conductorの登録 (4/7)

### ●Conductorの登録

以下のようにConductorを作成してください

The screenshot displays the Conductor registration interface. On the left, a workflow diagram is shown on a grid background. It starts with a 'Start' node (S) leading to a 'create\_directory' node (2), then a 'create\_file' node (3), and an 'edit\_file' node (4). These nodes are connected by a line that loops back to a 'Conditional branch' node (1). The 'Conditional branch' node has two output paths: one leading to a 'copy\_file' node (1) and another leading to a 'Conductor call' node (Cc). The 'copy\_file' node leads to a 'Parallel branch' node (6), which then leads to a 'Parallel merge' node (7). The 'Conductor call' node leads to another 'Conductor call' node (E). A red box highlights the 'Conditional branch' node, and a red arrow points from it to the 'Conditional branch' configuration panel at the bottom left. The configuration panel shows 'Case 1' set to '正常終了' (Normal End) and 'Other' cases set to '異常終了' (Abnormal End), '緊急停止' (Emergency Stop), '準備エラー' (Preparation Error), '想定外エラー' (Unexpected Error), 'Skip完了' (Skip Complete), and '警告終了' (Warning End). On the right, the 'Conductor名称' (Conductor Name) panel shows 'ID: Auto numbering', 'Name: Conductor\_01', and 'Permission role'. Below this, the 'Function' tab is selected, showing a list of function types: 'Conductor pause', 'Conductor call', 'Symphony call', 'Conditional branch', 'Parallel branch', and 'Parallel merge'. The 'Conditional branch' option is highlighted with a red box. A text box in the center of the diagram reads: 「Function」タブから Conductor branchを ドラッグアンドドロップして配置 (Drag and drop Conductor branch from the 'Function' tab and place it).

「Function」タブから  
Conductor branchを  
ドラッグアンドドロップして配置

Conditional branch

Case : Add Delete

Case 1 : 正常終了

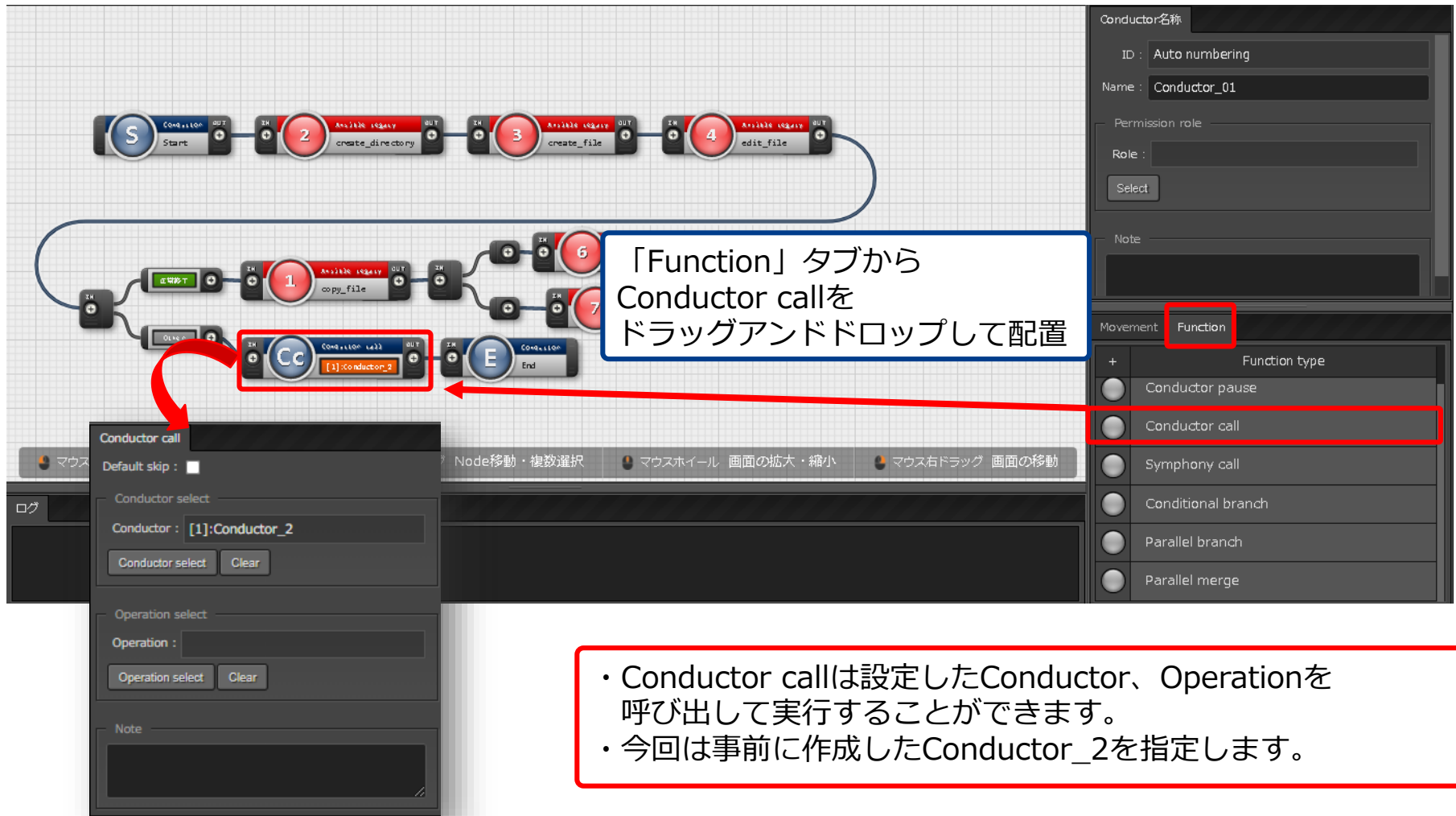
Other : 異常終了 緊急停止 準備エラー  
想定外エラー Skip完了 警告終了

- ・ Conductor branchは配置された直前のMovementの終了結果に応じて次の処理を分岐します。
- ・ 今回は「正常終了」の場合のみ後続処理に続く設定にします。

### 3.8 Conductorの登録 (5/7)

## ●Conductorの登録

以下のようにConductorを作成してください



## 3.8 Conductorの登録 (6/7)

### ●Conductorの登録

以下のようにConductorを作成してください

The screenshot displays the Conductor registration interface. On the left, a workflow diagram is shown on a grid background. It starts with a 'Start' node (S) connected to a 'create\_directory' node (2), which then connects to a 'create\_file' node (3). Below this, there is a 'copy\_file' node (1) and a 'remove\_file' node (7). A 'Conductor call' node (Cc) is connected to a 'Conductor end' node (E). A text box with a blue border and white background points to the 'Conductor end' node, stating: 「Function」タブから Conductor endをドラッグアンドドロップして配置. On the right, a panel shows the 'Conductor名称' (Conductor Name) section with fields for 'ID' (Auto numbering) and 'Name' (Conductor\_01). Below this is a 'Function' tab with a list of function types: 'Conductor end', 'Conductor pause', 'Conductor call', and 'Symphony call'. The 'Conductor end' option is selected and highlighted with a red box. At the bottom, a toolbar contains icons for node selection, movement, and zooming.

「Function」タブから  
Conductor endを  
ドラッグアンドドロップして配置

Conductor名称  
ID : Auto numbering  
Name : Conductor\_01  
Permission role  
Role :  
Select  
Note

Movement Function  
+ Function type  
Conductor end  
Conductor pause  
Conductor call  
Symphony call

マウス左クリック Node選択・接続線削除  
マウス左ドラッグ Node移動・複数選択  
マウスホイール 画面の拡大・縮小  
マウス右ドラッグ 画面の移動

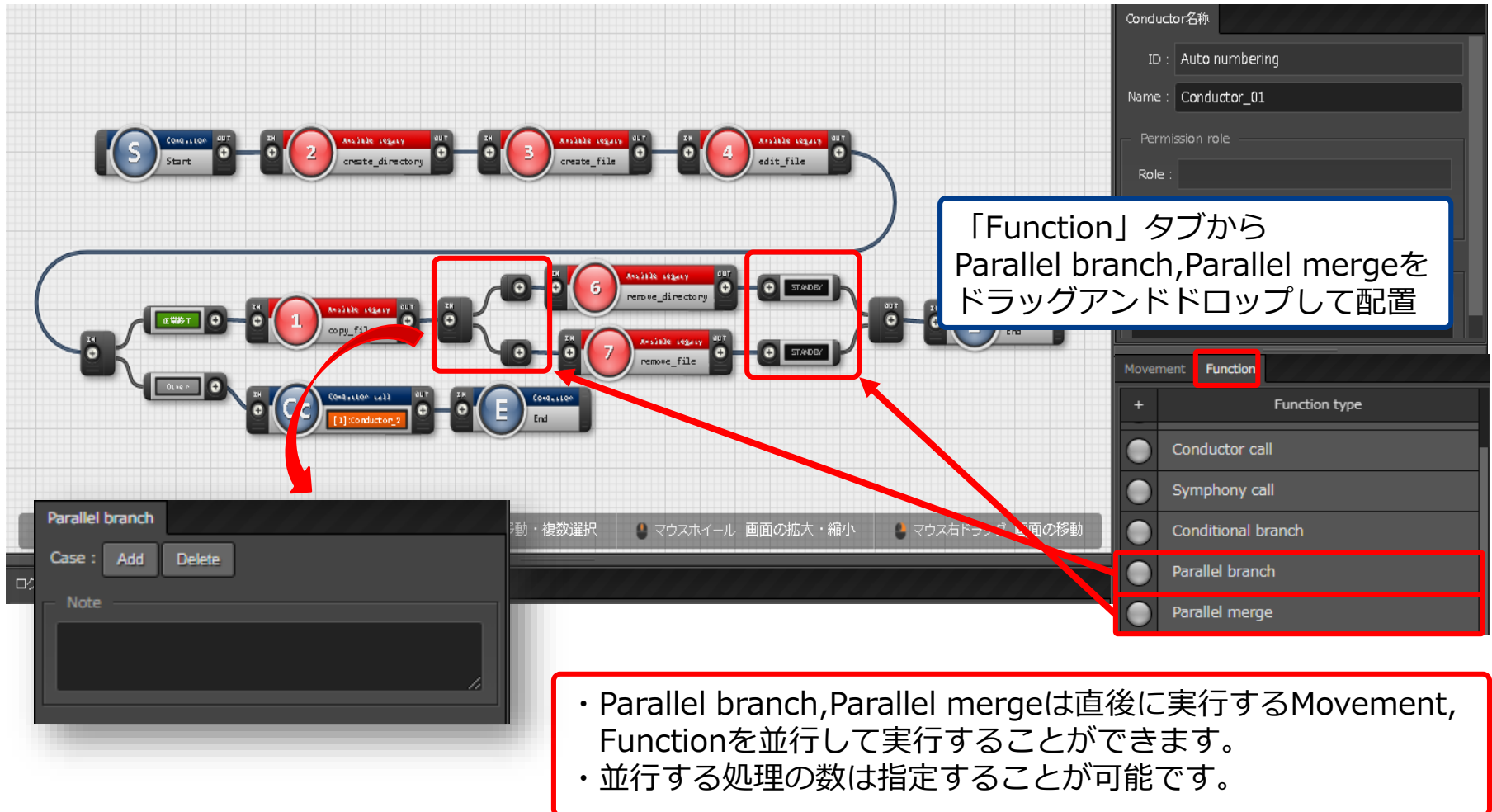
ログ

- Conductor endは処理の終了時に配置するfunctionです。
- (5/7)にてご紹介した分岐処理の終了時にも配置しています

## 3.8 Conductorの登録 (7/7)

### ●Conductorの登録

以下のようにConductorを作成してください



## 3.9 Conductorの実行

### ●Conductorの実行

「Conductor」メニューグループ >> 「Conductor作業実行」メニュー

- ① 「Conductor[一覧]」サブメニュー「予約日時」項目内のから実行日時を決定
- ② 「Conductor[一覧]」サブメニュー「Conductor名称」項目内の「Conductor\_1」を選択
- ③ 「オペレーション[一覧]」サブメニュー「オペレーション名」項目内の「operation1」を選択
- ④ 「実行」ボタンを押下

The screenshot displays the Conductor execution interface. It includes sections for scheduling, conductor selection, operation selection, and a workflow diagram.

**1** 下記の値を選択する

項目	値
予約日時	任意

**2** 下記の値を選択する

項目	値
Conductor名称	Conductor_1

**3** 下記の値を選択する

項目	値
オペレーション名	operation1

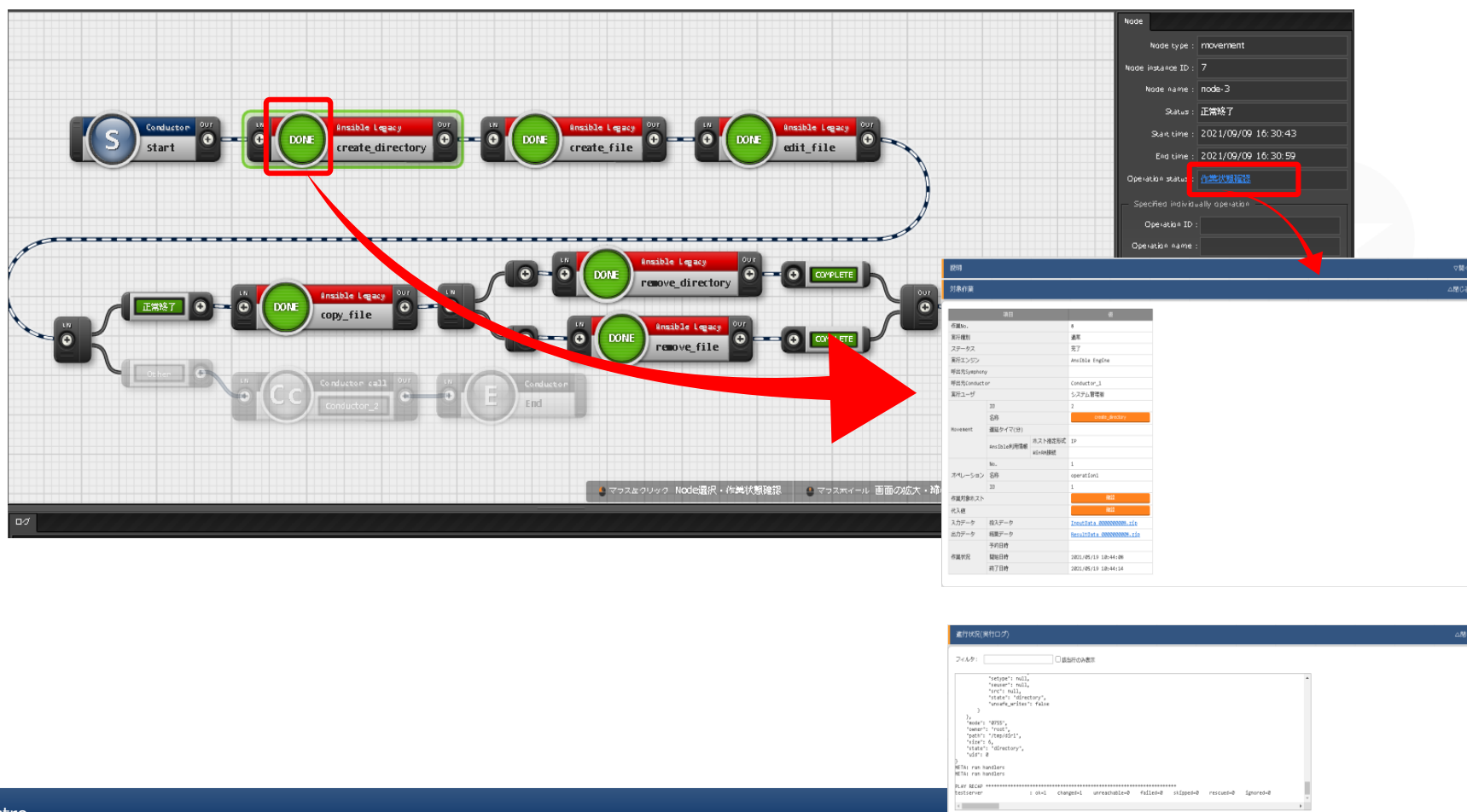
**4** 実行ボタンを押下

The workflow diagram on the right shows a sequence of steps: Start, create\_directory, create\_file, write\_file, read\_file, delete\_file, and End. The 'Conductor' and 'Operation' tabs are visible on the right side of the interface.

### 3.10 Conductor完了確認

## ●Conductor完了確認

実行中または実行完了したMovementを選択し、Doneのアイコンまたは右側のOperation statusをクリックすると対象作業ステータスや、ログを確認できる画面に遷移します。





**Exastro**