



IT Automation Conductor 【実習編】

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

第1.0版

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詳細の登録

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、ご説明をしております。



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。
- また、シナリオを進めるにあたり、Ansible driverが必要となりますので、本シナリオでは、Ansible-Legacyを使用しご説明をいたします。

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

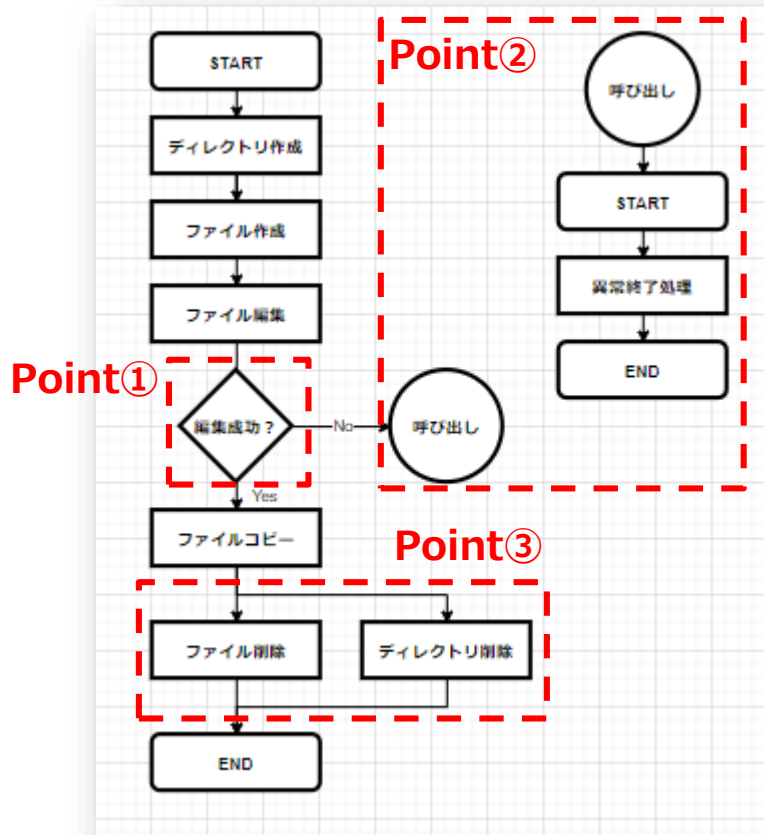
⑨実行結果確認

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/{{ VAR_dir_name_1 }}
    state=directory
    mode=0755

- 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

- 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	2020/10/30 9:55	YML ファイル	1 KB
 create_directory.yml	2020/10/30 9:55	YML ファイル	1 KB
 create_file.yml	2020/10/30 9:55	YML ファイル	1 KB
 edit_file.yml	2020/10/30 9:55	YML ファイル	1 KB
 forced_termination.yml	2020/10/30 9:55	YML ファイル	1 KB
 remove_directory.yml	2020/10/30 9:55	YML ファイル	1 KB
 remove_file.yml	2020/10/30 9:55	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

登録

管理システム項番 HW機器種別

自動入力 ▼

※*は必須項目です。

戻る 登録

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 >> 「投入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

No.	オペレーションID	オペレーション名*	実施予定日時*	備考	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

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
プレイブック素材名	operation1
実施予定日時	(任意の日時)

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 >> 「プレイブック素材集」メニュー >> 「登録」サブメニュー >> 「登録開始」ボタン

- ① 「プレイブック素材名」を入力、
「プレイブック素材」欄の「参照」ボタンを押下し
事前に作成したyml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

Exastro IT Automation Ansible-Legacy

Menu
メインメニュー
Movement一覧
プレイブック素材集
Movement詳細
変数一覧
Movement変数組付管理
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

素材ID
自動入力
プレイブック素材名*
プレイブック素材*
備考
最終更新日時
最終更新者

参照...
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
プレイブック素材名	<任意>
プレイブック素材	<任意> .yml

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							△閉じる
更新	廃止	素材ID	プレイブック素材名	プレイブック素材	備考	最終更新日時	最終更新者
更新	廃止	1	copy_file	copy_file.yml		2020/10/30 09:56:59	システム管理者
更新	廃止	2	create_directory	create_directory.yml		2020/10/30 09:57:25	システム管理者
更新	廃止	3	create_file	create_file.yml		2020/10/30 09:57:38	システム管理者
更新	廃止	4	edit_file	edit_file.yml		2020/10/30 09:57:53	システム管理者
更新	廃止	5	forced_termination	forced_termination.yml		2020/10/30 09:58:17	システム管理者
更新	廃止	6	remove_directory	remove_directory.yml		2020/10/30 09:58:35	システム管理者
更新	廃止	7	remove_file	remove_file.yml		2020/10/30 09:58:48	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

Exastro IT Automation Ansible-Legacy

Menu

メインメニュー

Movement一覧

ブロック素材集

Movement詳細

変数名一覧

Movement変数紐付管理

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

1 項目へ値を設定する

項目	値
Movement名	<任意>
ホスト指定形式	IP

MovementID

Movement名*

遅延タイマー

ホスト指定形式*

WinRM接続

Ansible利用情報

自動入力

戻る

登録

2

※*は必須項目です。

POINT

作成するMovementは
yaml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										△閉じる
更新	廃止	MovementID	Movement名	オーケストレータ	遅延タイマー	Ansible利用情報			最終更新日時	最終更新者
						ホスト指定形式	WinRM接続	ヘッダーセクション		
更新	廃止	3	copy_file	Ansible Legacy		IP			2020/10/30 11:03:52	システム管理者
更新	廃止	4	create_directory	Ansible Legacy		IP			2020/10/30 11:04:06	システム管理者
更新	廃止	5	create_file	Ansible Legacy		IP			2020/10/30 11:04:14	システム管理者
更新	廃止	6	edit_file	Ansible Legacy		IP			2020/10/30 11:04:25	システム管理者
更新	廃止	7	forced_termination	Ansible Legacy		IP			2020/10/30 11:04:34	システム管理者
更新	廃止	8	remove_directory	Ansible Legacy		IP			2020/10/30 11:04:44	システム管理者
更新	廃止	9	remove_file	Ansible Legacy		IP			2020/10/30 11:04:52	システム管理者

フィルタ結果件数: 7

Excel出力

3.5 Movement詳細の登録 (1/2)

Movement詳細の登録

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

- ① 「Movement」「プレイブック素材」「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

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
プレイブック素材集
Movement詳細

変数名一覧
Movement変数紐付管理
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

説明
表示フィルタ
一覧/更新
登録

紐付項目
自動入力

Movement*
プレイブック素材*
インクルード順序*

備考
最終更新日時
自動入力

※*は必須項目です。

戻る 登録

項目へ値を設定する

項目	値
Movement	作成した Movementを選択
プレイブック素材	登録したプレイブックを選択
インクルード順序	1

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詳細の登録 (2/2)

●Movement詳細の登録

登録後のイメージは以下になります。

一覧/更新								△閉じる
更新	廃止	紐付項番	Movement	プレイブック素材	インクルード順序	備考	最終更新日時	最終更新者
更新	廃止	13	copy_file	copy_file	1		2020/10/30 11:07:29	システム管理者
更新	廃止	24	create_directory	create_directory	1		2020/10/30 11:07:39	システム管理者
更新	廃止	35	create_file	create_file	1		2020/10/30 11:07:48	システム管理者
更新	廃止	46	edit_file	edit_file	1		2020/10/30 11:07:58	システム管理者
更新	廃止	57	forced_termination	forced_termination	1		2020/10/30 11:08:06	システム管理者
更新	廃止	68	remove_directory	remove_directory	1		2020/10/30 11:08:14	システム管理者
更新	廃止	79	remove_file	remove_file	1		2020/10/30 11:08:33	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

The screenshot shows the 'Ansible-Legacy' interface. On the left is a sidebar menu with options like 'Menu', 'メインメニュー', 'Movement一覧', 'ブロック素材集', 'Movement詳細', '変数名一覧', 'Movement変数紐付管理', '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', and '作業管理'. The main area has a '登録' (Registration) section with a table for inputting values. The table has three columns: '項目' (Item), 'オペレーション*' (Operation*), 'Movement*' (Movement*), and 'ホスト*' (Host*). The '項目' column has a dropdown menu with '自動入力' (Automatic Input) selected. Below the table, there are buttons for '戻る' (Back) and '登録' (Register). A red box highlights the table and the '登録' button. A red circle with the number '1' points to the table, and a red circle with the number '2' points to the '登録' button.

項目	オペレーション*	Movement*	ホスト*
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

1

項目へ値を設定する

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

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

Menu

メインメニュー

Movement一覧

プレイブック素材集

Movement詳細

変数名一覧

Movement変数紐付管理

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 具体値 代入

自動入力 1:ope1

※*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

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	3:copy_file:1:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	3:copy_file:2:VAR_file_name	dir2	
1:operation1	1:Testserver	3:copy_file:3:VAR_edit_param_1	testfile	
1:operation1	1:Testserver	4:create_directory:4:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	5:create_file:5:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	5:create_file:6:VAR_file_name	testfile	
1:operation1	1:Testserver	6:edit_file:7:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	6:edit_file:8:VAR_file_name	testfile	
1:operation1	1:Testserver	6:edit_file:9:VAR_edit_param_1	param1	
1:operation1	1:Testserver	7:forced_termination:10:VAR_message_text	testmsg_fail	
1:operation1	1:Testserver	8:remove_directory:11:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	9:remove_file:12:VAR_dir_name_1	dir1	
1:operation1	1:Testserver	9:remove_file:12:VAR_dir_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」 >>
「Conductor Name」を入力

- ① 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
- ② 「登録」ボタンを押下

項目へ値を入力する

項目	値
Name	Conductor_1

ドラッグ&ドロップ

※作成するConductorは次ページ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

The diagram illustrates a Conductor workflow on a grid background. It consists of three main components connected in a sequence:

- Start Conductor:** A blue box with a large 'S' on the left, 'Conductor' and 'Start' in the center, and an 'OUT' port with a plus sign on the right.
- Function Block:** A red box with a large '7' on the left, 'Ansible Legacy' and 'forced_termination' in the center, and 'IN' and 'OUT' ports with plus signs on either side.
- End Conductor:** A blue box with a large 'E' on the left, 'Conductor' and 'End' in the center, and an 'IN' port with a plus sign on the left.

Connections are shown as blue lines: one from the 'OUT' port of the Start Conductor to the 'IN' port of the Function Block, and another from the 'OUT' port of the Function Block to the 'IN' port of the End Conductor.

On the right side, the configuration panel for the Conductor is shown. It includes fields for 'ID' (set to 1) and 'Name' (set to 'Conductor_2'), and a 'Note' text area. Below this is a table listing available movements and functions.

Movement	Function	
+	ID	Movement name
3	copy_file	
4	create_directory	
5	create_file	
6	edit_file	
7	forced_termination	
8	remove_directory	
9	remove_file	

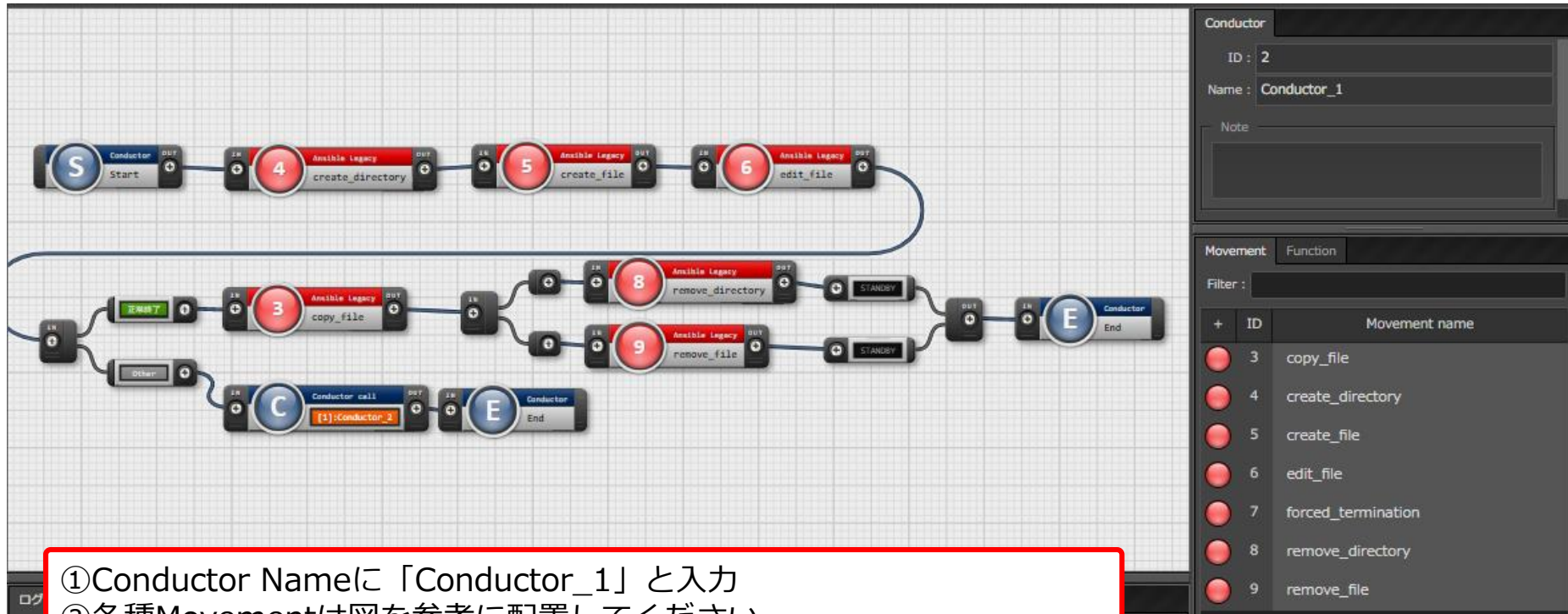
Below the diagram, a red-bordered box contains the following steps:

- 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
- ③Functionから「Conductor」をドラッグアンドドロップ
- ④図のように「OUT」と「IN」をつなぐ
- ⑤画面下の「登録」を押下

3.8 Conductorの登録 (3/7)

●Conductorの登録

作成Conductor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 Exastro Conductor configuration interface. On the left, a workflow diagram is shown on a grid background. It starts with a 'Conductor Start' node (S), followed by a sequence of 'Ansible legacy' tasks: 'create_directory' (4), 'create_file' (5), and 'edit_file' (6). These are connected to a 'Conductor call' node (C) which calls 'Conductor_2'. This is followed by an 'End' node (E). A 'Conductor Branch' node is placed before the 'Conductor call' node. A red box highlights this branch node, and a red arrow points from a text box to it. The text box contains the instruction: 「Function」タブから Conductor Branchを ドラッグアンドドロップして配置. Below the main diagram, a 'Conditional branch' configuration window is open. It shows 'Case 1' with the label '正常終了' (Normal End) and 'Other' with three options: '異常終了' (Abnormal End), '緊急停止' (Emergency Stop), and '準備エラー' (Preparation Error). The '正常終了' option is selected. On the right, the 'Conductor' configuration panel is visible, showing 'ID: 2' and 'Name: Conductor_1'. Below this, the 'Movement' tab is selected, showing a list of 'Function type' options: 'Conductor end', 'Conductor pause', 'Conductor call', 'Conditional branch', 'Parallel branch', and 'Parallel merge'. The 'Conditional branch' option is highlighted with a red box. A red arrow points from this option to the 'Conductor Branch' node in the workflow diagram.

「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

- 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

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

Conductor call configuration:

- Default skip: ☐
- Conductor select:
Conductor: [1]:Conductor_2
Buttons: Conductor select, Clear
- Operation select:
Operation:
Buttons: Operation select, Clear
- Note:

Function type list:

- Conductor end
- Conductor pause
- Conductor call
- Conditional branch
- Parallel branch
- Parallel merge

Conductor call configuration details:

Conductor: [1]:Conductor_2

Operation:

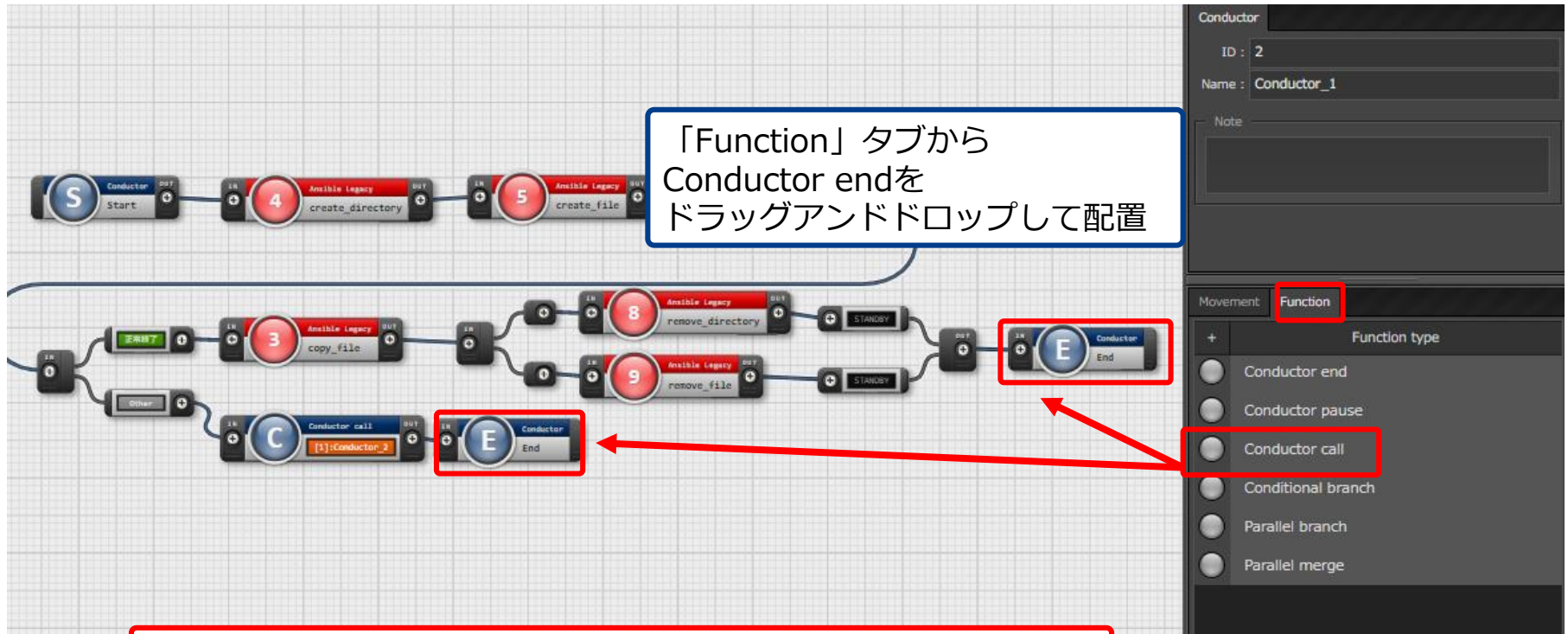
Note:

- Conductor callは設定したConductor、Operationを呼び出して実行することができます。
- 今回は事前に作成したConductor_2を指定します。

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

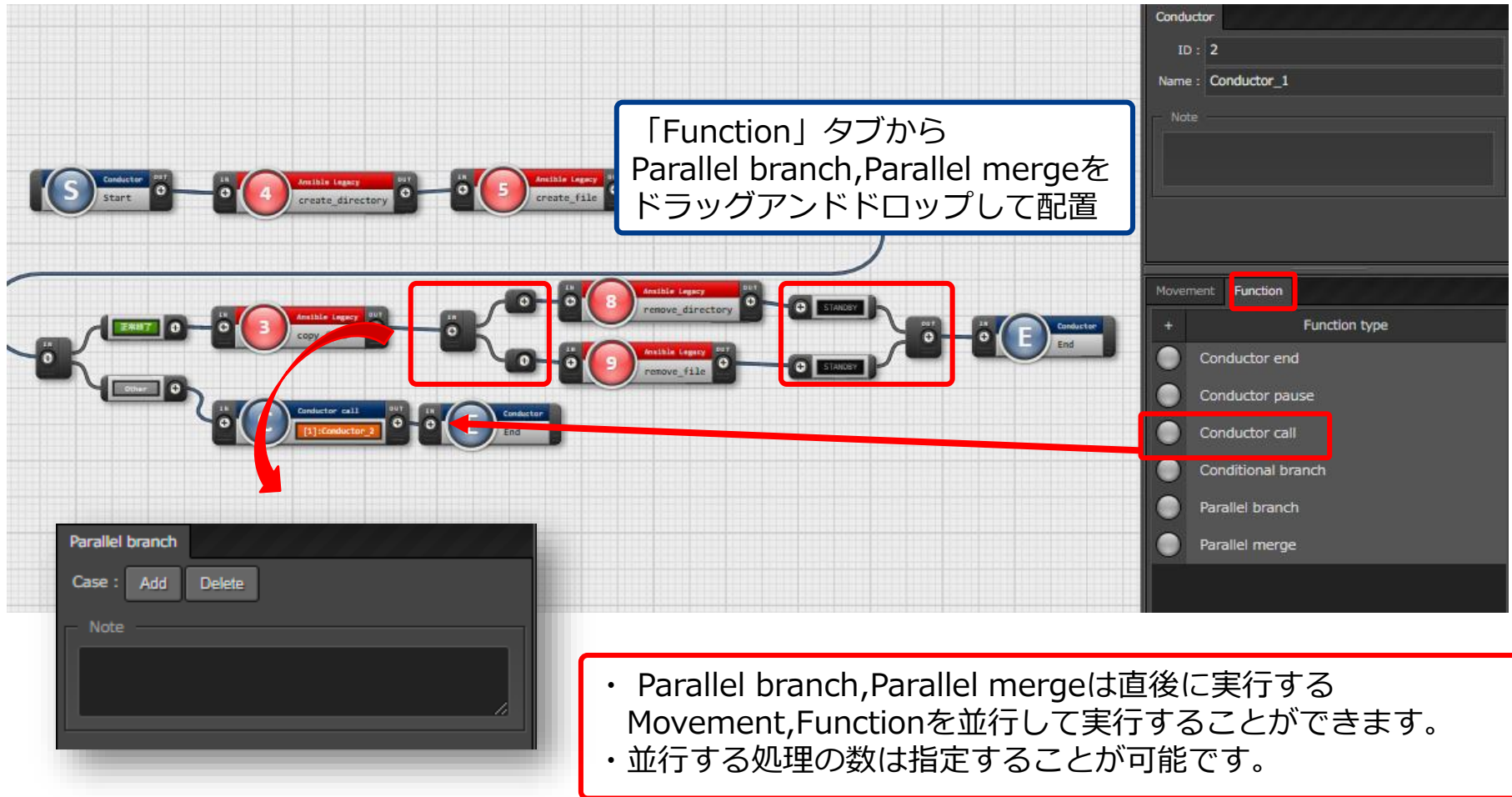


- Conductor endは処理の終了時に配置するfunctionです。
- (5/7)にてご紹介した分岐処理の終了時にも配置しています

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_2」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「operation」を選択
- ④ 「実行」ボタンを押下

The screenshot shows the 'Conductor' menu group with the 'Conductor作業実行' (Conductor Execution) menu selected. The interface is divided into several sections: '説明' (Description), 'スケジューリング' (Scheduling), 'Conductor[フィルタ]' (Conductor Filter), 'Conductor[一覧]' (Conductor List), 'オペレーション[フィルタ]' (Operation Filter), and 'オペレーション[一覧]' (Operation List). The 'Conductor[一覧]' section is highlighted, showing a table with columns '選択' (Select), 'ConductorクラスID' (Conductor Class ID), and 'Conductor'. The 'オペレーション[一覧]' section is also highlighted, showing a table with columns '選択' (Select), 'No.', 'オペレーションID' (Operation ID), and 'オペレーション名' (Operation Name). The '実行' (Execute) button is located at the bottom right of the interface.

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

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。

The screenshot displays the Conductor interface. On the left, a workflow diagram shows a sequence of operations. A red box highlights a 'DONE' status for the 'create_directory' operation. A large red arrow points from this box to the right-hand panel, which provides a detailed view of the selected operation.

Node Information:

- Node type: movement
- Node instance ID: 80
- Node name: node-7
- Status: 正常終了
- Start time: 2020/11/04 14:43:50
- End time: 2020/11/04 14:44:02
- Operation status: https://10.197.19.110:18431/default/menu/01_browser.php?no=2100020112&revision_no=12

Operation Information:

- Operation ID: 1
- Operation name: operation1

対象作業 (Target Task) Table:

項目	値
作業No.	12
実行権限	通常
ステータス	完了
実行エンジン	Ansible Engine
呼出元Symphony	
呼出元Conductor	Conductor_1
実行ユーザ	システム管理者
Movement	
ID	4
名称	create_directory
所要タイム(分)	
Ansible利用構成	IP
No.	1
オペレーション	
名称	operation1
ID	1
作業対象ホスト	
代入値	指定
入力データ	入力データ 0000000011.json
出力データ	結果データ 0000000011.json
作業状況	
開始日時	2020/11/04 14:43:50
終了日時	2020/11/04 14:43:59

実行状況(実行ログ) (Execution Status (Execution Log))

フィルタ: ☐ 該当行のみ表示

```

{
  "source": null,
  "setype": null,
  "source": null,
  "src": null,
  "status": "directory",
  "unsafe_writes": null
}
}
}
"node": "0755",
"user": "root",
"path": "/tmp/ds1",
"size": 6,
"state": "directory",
"uid": 0
}
}
META: ran handlers
META: ran handlers
PLAY RECAP *****
Taskrunner : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

```



Exastro