



# IT Automation Conductor 【座学編】

※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についての説明

### 1. Conductorとは

### 2. Conductorの特徴

### 3. Conductorメニューの機能説明

### 4. Conductorの作業フロー

# 1. はじめに

# 1.1 本書について

## ■ メインメニュー

- 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...

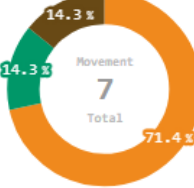
 Ansible-Pi...

 Ansible-Le...

 Terraform

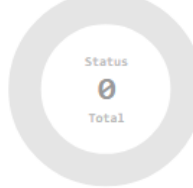
 CI/CD for ...

Movement



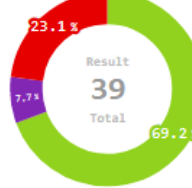
Movement	SUM
Ansible Legacy	5
Ansible Pioneer	1
Ansible Legacy Role	1
Terraform	0

作業状況



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



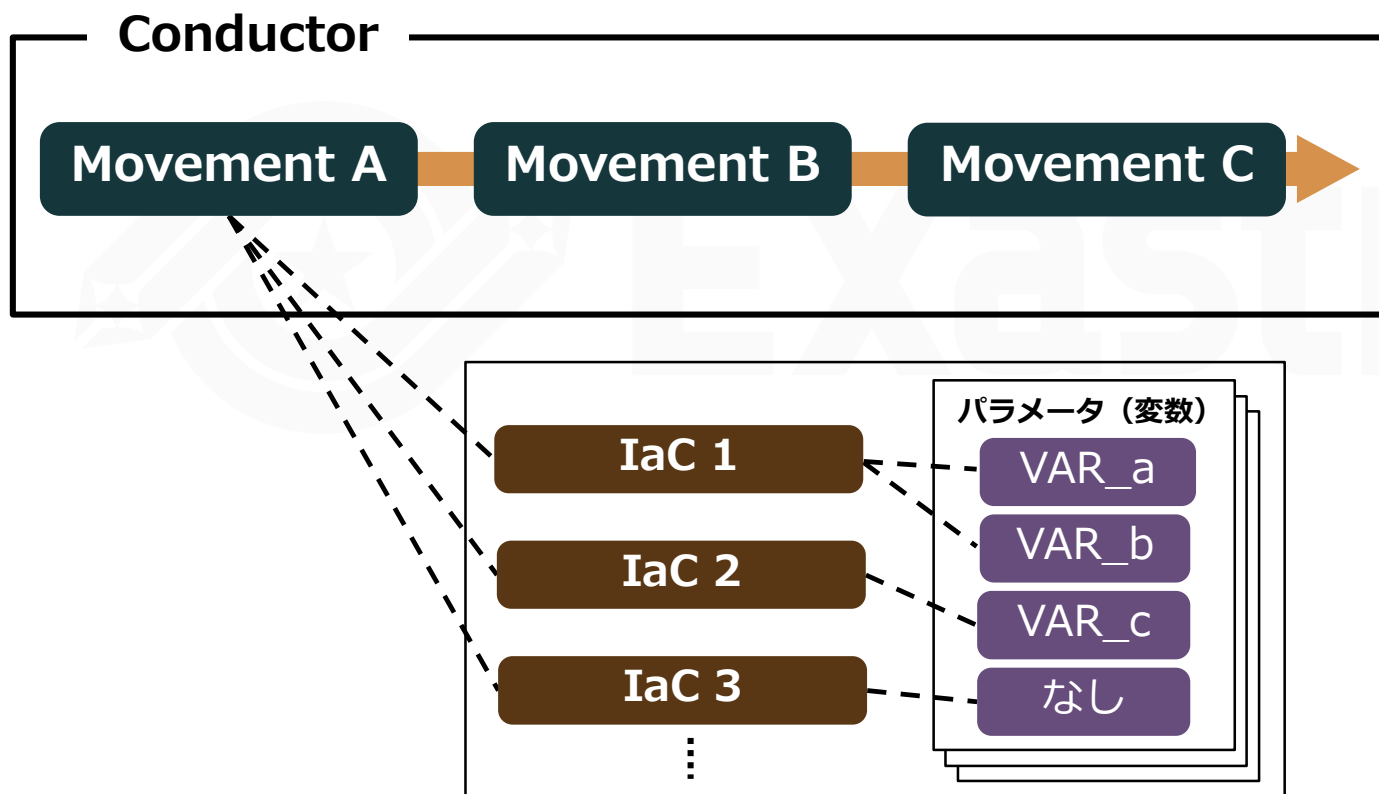
Result	CON	SYM	SUM
正常終了	27	0	27
異常終了	3	0	3
想定外エラー	9	0	9
緊急停止	0	0	0
予約取消	0	0	0

作業履歴

## 2. Conductorについての説明

## 2.1 Conductorとは

- Conductorは、ITAにver1.5.0より追加された機能です。
- ITA における一連の作業の単位を指し、オペレーション名と関連付けて実行します。（ジョブフロー）



## 2.2 Conductorの特徴

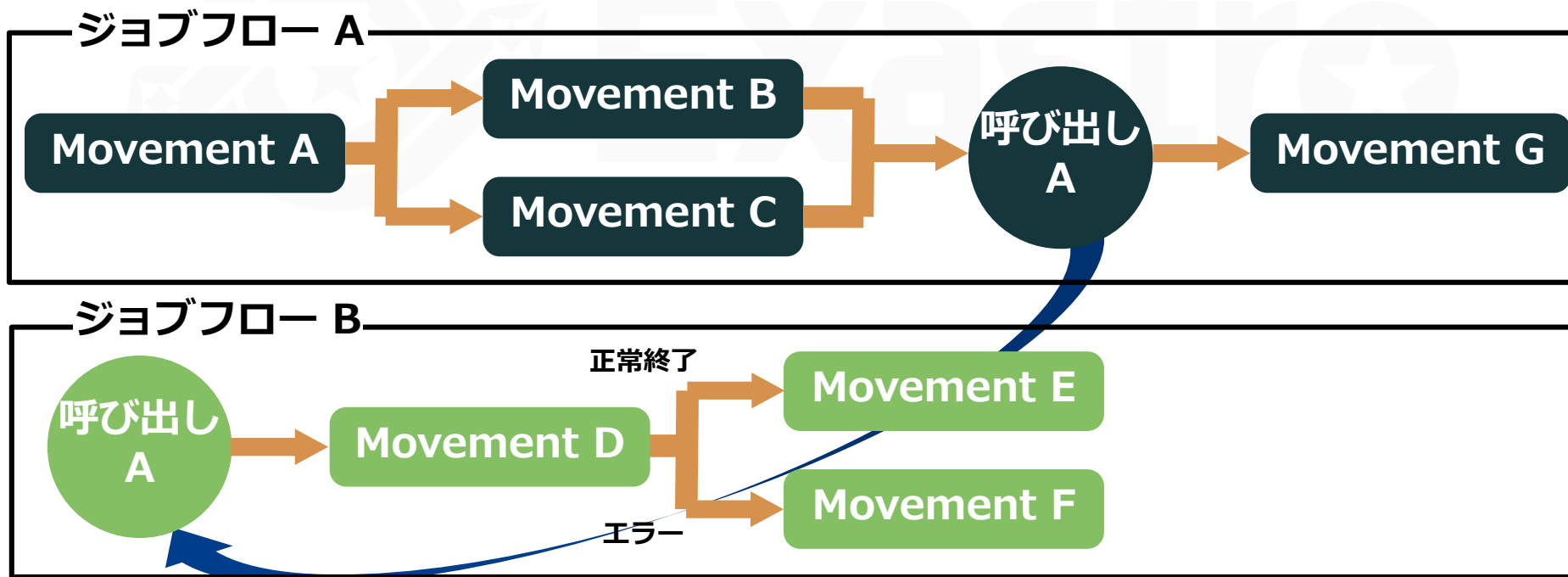
- ConductorではSymphony機能と同様の作業実行機能に加え、以下の機能を備えています。

これらによって、Conductorでは**より高度なジョブフローを実行できます。**

- Movement の並列実行**

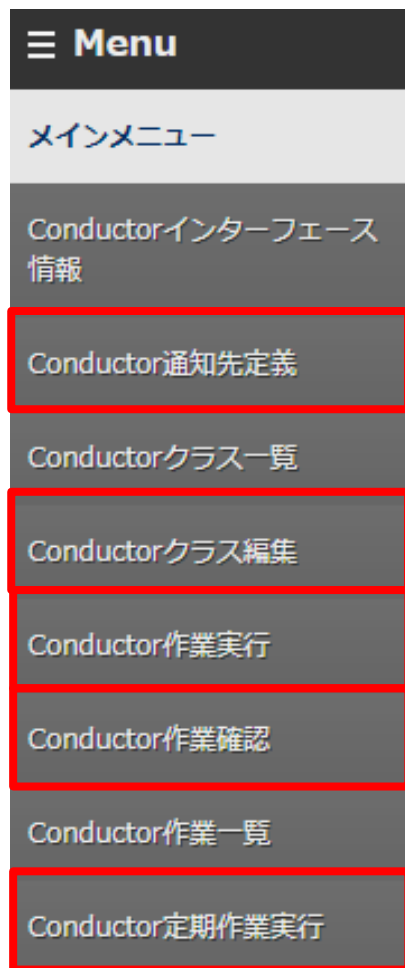
- 別のジョブフローの呼び出し**

- Movement の実行結果、または返り値による条件分岐**



## 2.3 Conductorメニューの機能説明 (1/14)

- Conductorにおける主なメニュー機能を紹介します。



### ①Conductorクラス編集

作成したMovementを使用してオペレーションを作成します。

### ②Conductor作業実行

オペレーションを実行します。

### ③Conductor作業確認

作成したオペレーションの確認できます。

### ④Conductor定期作業実行

オペレーションを登録し、定期実行をおこないます。

### ⑤Conductor通知先定義

Conductorで作業時に実行される通知に関する定義を設定します。



## 2.3 Conductorメニューの機能説明 (2/14)

### ●Conductorクラス編集 (1/4)

- 「Conductorクラス編集」メニューではMovement、各種制御を行うFunctionの追加、削除が可能です。

The screenshot displays the Conductor editor interface. The main workspace shows a workflow with three nodes: a 'Start' node (blue circle with 'S'), an 'Ansible Legacy' node (red circle with '1'), and an 'End' node (blue circle with 'E'). A red double-headed arrow connects the 'Ansible Legacy' node to the 'End' node. The right sidebar shows a list of movements with columns for ID and Movement name. The first movement is '1' with name 'Monement\_1'. A red box highlights the 'Function' button above the list. A red arrow points from the 'Function' button to the 'Monement\_1' row. Another red arrow points from the 'Monement\_1' row to the 'Ansible Legacy' node in the workflow.

Movementごとのin/outをドラッグ&ドロップでMovementの結合が可能です

ドラッグ&ドロップすることでMovementの配置ができます

各種Functionが選択可能です

## 2.3 Conductorメニューの機能説明 (3/14)

### ●Conductorクラス編集 (2/4)

- 画面右中央付近のタブから、オペレーションの条件分岐を制御する **Function**を選択、使用することが可能です。

The screenshot displays the Conductor menu interface. On the left, a workflow diagram is shown on a grid background. It starts with a 'Start' node (blue circle with 'S'), followed by a split into two parallel paths. Each path contains a red circle with a number (1 and 2) and a box labeled 'Ansible Legacy' and 'Movement\_1'/'Movement\_2'. These paths merge and lead to an 'End' node (blue circle with 'E'). A red arrow points from the 'Function' tab in the right panel to the 'Movement\_2' node in the diagram.

Movementと同様にin/outを  
ドラッグ&ドロップで結合が可能です

ドラッグ&ドロップすることで  
Functionの配置ができます

各種Functionが選択可能です

Function type
Conductor end
Conductor pause
Conductor call
Symphony call
Conditional branch
Parallel branch
Parallel merge

### ●Conductorクラス編集 (3/4)

- Nodeを複数選択することで、オブジェクトの整列機能を使用することが可能です。Nodeを複数選択する方法については、ドラッグアンドドロップでの範囲選択の他「shift」キーを押下しながらの選択が可能です。以下のようにNodeタブの使用により、オブジェクトを整列することが可能です。Nodeタブの詳細は[マニュアル](#)をご覧ください。

The screenshot shows the Conductor menu interface. On the left, a workflow diagram is displayed with three nodes: 'Start' (Conductor), 'Movement\_1' (Ansible Legacy), and 'End' (Conductor). A blue callout box points to the 'Start' and 'Movement\_1' nodes, stating: 'Nodeを複数選択することによってオブジェクトの整列機能が表示されます' (By selecting multiple nodes, the alignment function is displayed). A red arrow points from this callout to the 'Node' panel on the right. The 'Node' panel has a '整列' (Align) section with several icons. A red box highlights the 'Align' icons, and a blue callout box points to it, stating: 'タブを選択することでオブジェクトの整列ができます' (By selecting the tab, you can align the objects). Below the 'Node' panel, a red arrow points to the final state of the workflow diagram, where the three nodes are now aligned horizontally.

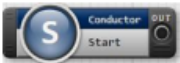





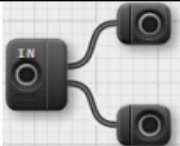
Nodeを複数選択することによって  
オブジェクトの整列機能が表示されます

タブを選択することで  
オブジェクトの整列ができます

## 2.3 Conductorメニューの機能説明 (5/14)

### ●Conductorクラス編集 (4/4)

- 使用可能なFunctionを記載します。詳細は[マニュアル](#)をご覧ください。

画像	名称	動作説明
	Conductor start	Conductorを開始します
	Conductor end	Conductor 終了します。 ※複数の Conductor end がある場合、全ての Conductor end が終了を待ちます。
	Conductor pause	ワークフローを一時停止します。 一時停止を解除すると、次の処理へ進みます。
	Conductor call	別の登録済みの Conductor クラスを呼び出し実行します。 ※呼び出し先の Conductor が警告終了で終了した場合、正常終了と同じように後続の処理を実行して、呼び出し元のステータスには影響しません。
	Symphony call	登録済みの Symphony クラスを呼び出し実行します。
	Conditional branch	接続された、「Movement」、「Conductor call」、「Symphony call」の結果によって、後続の処理を分岐させます。 指定可能なステータスは、以下になります。 <ul style="list-style-type: none"><li>・正常終了</li><li>・異常終了</li><li>・緊急停止</li><li>・準備エラー</li><li>・想定外エラー</li><li>・SKIP 完了</li><li>・警告終了</li></ul>
	Parallel branch	並列して、「Movement」、「Conductor call」、「Symphony call」を実行します。 ※並列可能な実行数は、ITA の構成やサーバースペックに依存します。

画像	名称	動作説明
	Parallel merge	接続している全 Node の処理が完了後、次の処理を実行します。
	Status file branch	接続された、「Movement」の作業結果ディレクトリ内のステータスファイルの内容によって、後続の処理を分岐させます。
	Movement 各種	Movement を実行します。

## 2.3 Conductorメニューの機能説明 (6/14)

### ●Conductor作業実行(1/2)

- 「Conductor作業実行」メニューでは作成したConductor、オペレーションを選択し実行します。

説明

スケジューリング

予約日時を指定する場合、日時フォーマット(YYYY/MM/DD HH:II)で入力して下さい。ブランクの場合は即時実行

予約日時

**実行タイミングの  
スケジューリングが可能です**

Conductor[フィルタ]

Conductor[一覧]

選択	ConductorクラスID	Conductor名称	説明	アクセス種 アクセス許可ロール	備考	最終更新日時	最終更新者
<input type="radio"/>	1	Conductor_01				2021/08/31 17:38:23	システム管理者
<input type="radio"/>	2	Conductor_02				2021/08/31 17:17:14	システム管理者

フィルタ結果件数: 2

**実行するConductor、  
オペレーションを選択します**

オペレーション[フィルタ]

オペレーション[一覧]

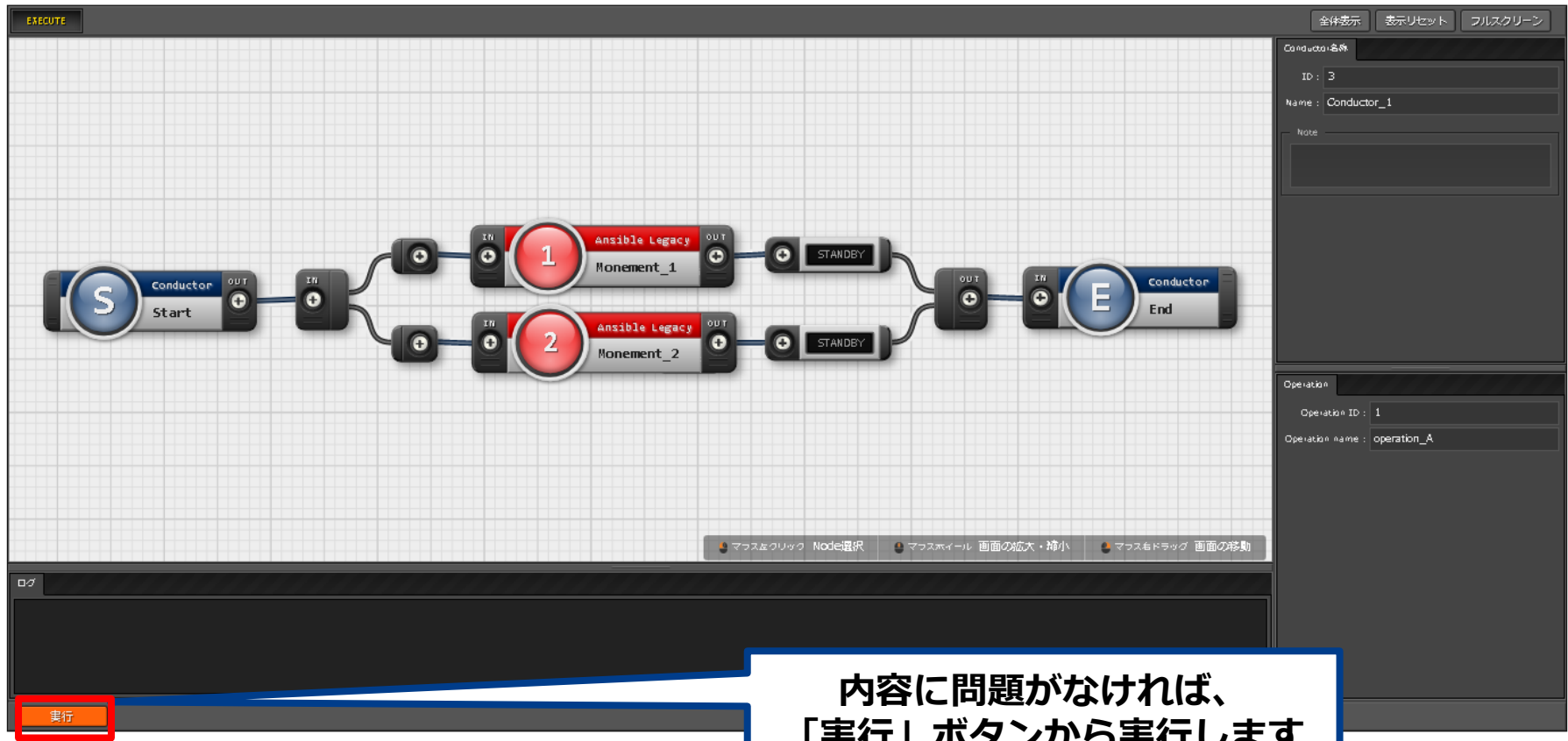
選択	No.	オペレーションID	オペレーション名	実施予定日時	最終実行日時	アクセス種 アクセス許可ロール	備考	最終更新日時	最終更新者
<input type="radio"/>	1	1	operation_A	2021/08/31 17:14	2021/08/31 17:46			2021/08/31 17:46:51	legacy作業実行プロセス
<input type="radio"/>	2	2	operation_B	2021/08/31 17:14				2021/08/31 17:14:42	システム管理者

フィルタ結果件数: 2

## 2.3 Conductorメニューの機能説明 (7/14)

### ●Conductor作業実行(2/2)

- ページ上部で選択したConductor、オペレーションが表示されます。



### ●Conductor作業確認 (1)

- 「Conductor作業確認」メニューから、Conductorの実行状態が確認可能です。

The screenshot displays the Conductor interface. At the top, a yellow 'CHECKING' status bar is visible. The main workspace shows a workflow diagram with three nodes: 'Start' (blue circle with 'S'), 'Monement\_1' (green circle with 'DONE'), and 'End' (blue circle with 'DONE'). The 'Monement\_1' node is highlighted with a red box. A blue callout bubble points to this node, containing the text: 'Movementをクリックすると、実行の詳細結果を確認可能です 詳細は[こちら](#)'. To the right, a details panel shows the following information:

全体表示	
Node type :	movement
Instance ID :	12
Node name :	node-3
Status :	正常終了
Start time :	2021/09/01 09:47:44
End time :	2021/09/01 09:48:03
Operation status :	<a href="#">作業状況確認</a>
Specified individually operation	
Operation ID :	
Operation name :	
Note	
Operation	
Operation ID :	1
Operation name :	operation_A

At the bottom left, a red box highlights a button labeled '緊急停止'. A blue callout bubble points to this button, containing the text: 'Conductorに予約日時が設定されていてかつ未実行の場合、「予約取り消し」ボタンから予約を取り消すことが可能です'.

## 2.3 Conductorメニューの機能説明 (9/14)

### ●Conductor作業確認 (1)

- 「Conductor」メニューグループ >> 「Conductor作業一覧」メニュー>> 「一覧」サブメニューにて、各Conductorの投入データと結果データを取得することができます。

The screenshot displays the Exastro Conductor web application interface. The top navigation bar includes the Exastro logo, the title 'Conductor', and user information: 'ようこそ[システム管理者]さん' (Welcome [System Administrator]), 'ログインID [administrator]' (Login ID [administrator]), and buttons for 'ロール' (Role), 'パスワード変更' (Change Password), and 'ログアウト' (Logout).

The left sidebar menu is titled 'Menu' and lists several options. The 'Conductor作業一覧' (Conductor Job List) option is highlighted with a red box.

The main content area shows the 'Conductor作業一覧' (Conductor Job List) page. It features a '説明' (Description) section with a '▽開く' (Expand) button, a '表示フィルタ' (Display Filter) section with an '△閉じる' (Close) button, and a table of Conductor jobs. The table has columns for '廃止' (Cancellation), 'ConductorインスタンスID' (Conductor Instance ID), 'Conductor名称' (Conductor Name), 'オペレーション名' (Operation Name), 'ステータス' (Status), '最終更新日時' (Last Update Date/Time), and '最終更新者' (Last Updated By).

Below the table, there are buttons for 'フィルタ' (Filter) and 'フィルタクリア' (Clear Filter), and a checkbox for 'オートフィルタ' (Auto Filter).

The '一覧' (List) section shows a table of Conductor jobs. The table has columns for '履歴' (History), 'ConductorインスタンスID' (Conductor Instance ID), '実行ユーザ' (Executing User), '緊急停止発令フラグ' (Emergency Stop Issuance Flag), '投入データ式 (zip)' (Input Data Format (zip)), '結果データ式 (zip)' (Result Data Format (zip)), '最終更新日時' (Last Update Date/Time), and '最終更新者' (Last Updated By). The '投入データ式 (zip)' and '結果データ式 (zip)' columns are highlighted with a red box, showing 'download(.zip)' buttons.

履歴	ConductorインスタンスID	実行ユーザ	緊急停止発令フラグ	投入データ式 (zip)	結果データ式 (zip)	最終更新日時	最終更新者
履歴	39	システム管理者	未発令	download(.zip)	download(.zip)	2022/01/07 15:20:43	コンダクター管理プロシージャ
履歴	38	システム管理者	未発令	download(.zip)	download(.zip)	2022/01/07 14:53:32	コンダクター管理プロシージャ
履歴	37	システム管理者	未発令	download(.zip)	download(.zip)	2021/12/16 17:26:32	コンダクター管理プロシージャ



## 2.3 Conductorメニューの機能説明 (10/14)

### ●Conductor定期作業実行 (1/3)

- 「Conductor定期作業実行」メニューでは、スケジュールに従って定期的に行う作業を管理します。

表示フィルタ

△閉じる

廃止	定期作業実行ID	Conductor名称	オペレーション名	ステータス	実行ユーザ		最終更新日時	最終更新者
廃止含まず ▼	~						~	
	▼ プルダウン検索	▼ プルダウン検索	▼ プルダウン検索	▼ プルダウン検索	▼ プルダウン検索			▼ プルダウン検索

フィルタ

フィルタクリア

☒ オートフィルタ

一覧/更新

レコードはありません。  
新規登録は下記より可能です。

登録

△閉じる

定期作業実行ID	Conductor名称 *	オペレーション名 *	ステータス	実行ユーザ	スケジュール設定	次回実行日付	開始日付 *	終了日付	曜日	最終更新日時	最終更新者
自動入力	Conductor_001 ▼	Operation01 ▼	自動入力	自動入力	スケジュール設定	自動入力	2021/05/10 17:00	2021/06/09 17:00	時	自動入力	自動入力

※\*は必須項目です。

戻る

登録

「スケジュール設定」から詳細なスケジュールが設定できます

## 2.3 Conductorメニューの機能説明 (11/14)

### ●Conductor定期作業実行 (2/3)

- 「スケジュール設定」は以下のように実行期間や、作業を停止する期間などの設定を細かく行うことが可能です。

スケジュールを設定する

作業期間

\* 開始日付: 2021/09/01 00:00 終了日付: 2021/09/03 00:00

スケジュール

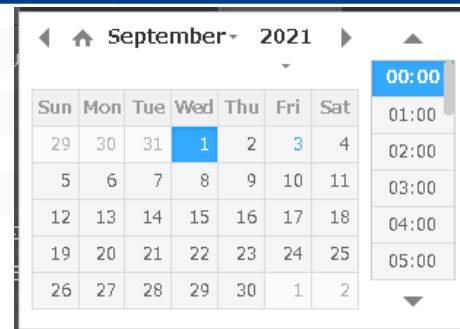
☒ 時 ☐ 日 ☐ 週 ☐ 月(日付指定) ☐ 月(曜日指定) ☐ 月末

\* 間隔: 8 時間ごと

作業停止期間

備考

クリックすると  
カレンダーが表示されます



現在日時「2021/8/31 00:00」に  
左図のようにスケジュール設定した場合  
Conductorの実行日時は以下になります。

初回(=次回実行日付) : 2021/09/01 00:00  
2回目 : 2021/09/01 08:00  
3回目 : 2021/09/01 16:00  
{

## 2.3 Conductorメニューの機能説明 (12/14)

### ●Conductor定期作業実行 (3/3)

- 定期作業実行に登録したConductor作業は、次回実行日付のインターバル時間前になると未実行（予約）ステータスに遷移します。（インターバル時間の初期値は3分）  
インターバル時間はユーザ側で「管理コンソール」メニューグループの「システム設定」メニューより設定可能です。変更手順は[管理コンソールの利用手順マニュアル](#)をご参照ください。
- ステータス 未実行（予約）のConductor/Symphonyは「Conductor作業一覧」メニューとダッシュボードの「作業状況」で確認できます。

【例】 次回実行日付が「2021/9/1 18:00」で インターバル時間が3分の場合、  
「2021/9/1 17:57」 に未実行（予約）ステータスに遷移します。



次回実行日付のインターバル時間前になると表示されます

## 2.3 Conductorメニューの機能説明 (13/14)

### ●Conductor通知先定義 (1/2)

- 指定のメッセージツール（Teams、Slack）にConductorの実行結果をhttpリクエストで送ることができます。「Conductor通知先定義」メニューで通知先や通知内容の定義を行います。詳細は[マニュアル](#)をご覧ください。
- 実行時間が長い処理や、予約実行の結果確認などでご活用いただけます。

Exastro IT Automation Conductor

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

Conductorインターフェース情報

Conductor通知先定義

Conductorクラス一覧

Conductorクラス編集

Conductor作業実行

Conductor作業確認

Conductor作業一覧

Conductor定期作業実行

説明

表示フィルタ

廃止	No	通知名称	HTTPリクエストオプション			PROXY		作業	最終更新日時	最終更新者
			通知先(CURLLOPT_URL)	ヘッダー(CURLLOPT_HTTPHEADER)	メッセージ(CURLLOPT_POSTFIELDS)	URL(CURLLOPT_PROXY)	PORT(CURLLOPT_PROXYPORT)			
廃止含まず ▼	～	▼ フルダウン検索	▼ フルダウン検索	▼ フルダウン検索	▼ フルダウン検索	▼ フルダウン検索	▼ フルダウン検索	▼ フルダウン検索	▼ フルダウン検索	▼ フルダウン検索

フィルタ フィルタクリア

☒ オートフィルタ

一覧/更新

登録

No	通知名称	通知先(CURLLOPT_URL)	ヘッダー(CURLLOPT_HTTPHEADER)	メッセージ(CURLLOPT_POSTFIELDS)	URL(CURLLOPT_PROXY)	PORT(CURLLOPT_PROXYPORT)	最終更新日時	最終更新者
自動入力							自動入力	自動入力

\*は必須項目です。

戻る 登録

通知先・通知内容の定義をします。

## 2.3 Conductorメニューの機能説明 (14/14)

### ●Conductor通知先定義 (2/2)

- 「Conductor クラス編集」の「Notice」から登録したConductor通知先定義を指定して、通知を発出するConductor作業ステータスを設定します。

The screenshot shows the Conductor interface with a workflow diagram on the left and a 'Notice list' dialog on the right. The workflow diagram consists of several nodes: a 'Start' node, followed by three 'Ansible Legacy' nodes (labeled 3, 4, and 5) with sub-nodes 'Install\_Packages', 'Open\_Ports', and 'Start\_Service', and finally an 'End' node. A callout box points to the 'Select' button in the 'Notice list' dialog, indicating that clicking it will display the 'Notice list'.

**Notice list**

通知名称	実行中	実行中(遅延)	正常終了	警告終了	緊急停止	異常終了	想定外エラー
通知サンプル	■	■	✓	✓	■	✓	✓

決定 取消

## 2.4 Conductorの作業フロー

- Conductorの作業フローは以下の通りです。  
実際の操作は実習編にて記載しています。

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



**Exastro**