



# IT Automation Conductor 【Tutorial】

※In this document, “Exastro IT Automation” is described as “ITA”.

# Table of contents

## 1. Introduction

### 1. About this document

## 2. Conductor

### 1. About Conductor

### 2. Conductor feature

### 3. Conductor Function Description

### 4. Conductor workflow

# 1. Introduction

## Main menu

- User name [System Administrator]  
Login ID [administrator]
Change password
Logout

Menu
Edit

Main menu

---

Menu group

Managemen...

Basic Console

Export/Import

Symphony

Conductor

File control ...

File control c...

Input

Substitution ...

Reference

Contrast

HostGroup ...

Ansible-Leg...

Ansible-Pion...

Ansible-Leg...

Cobbler

Terraform

Menu set

Ansible-Leg...

Ansible-Pion...

Ansible-Leg...

Cobbler

Terraform

### Movement

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

### Work status

Status	CON	SYM	SUM
Executing	0	0	0
Unexecuted (schedule)	0	0	0
Unexecuted	0	0	0

### Work result

Result	CON	SYM	SUM
Normal end	0	0	0
Abnormal end	0	0	0
Unexpected error	0	0	0
Emergency stop	0	0	0
Schedule cancellation	0	0	0

### Work history

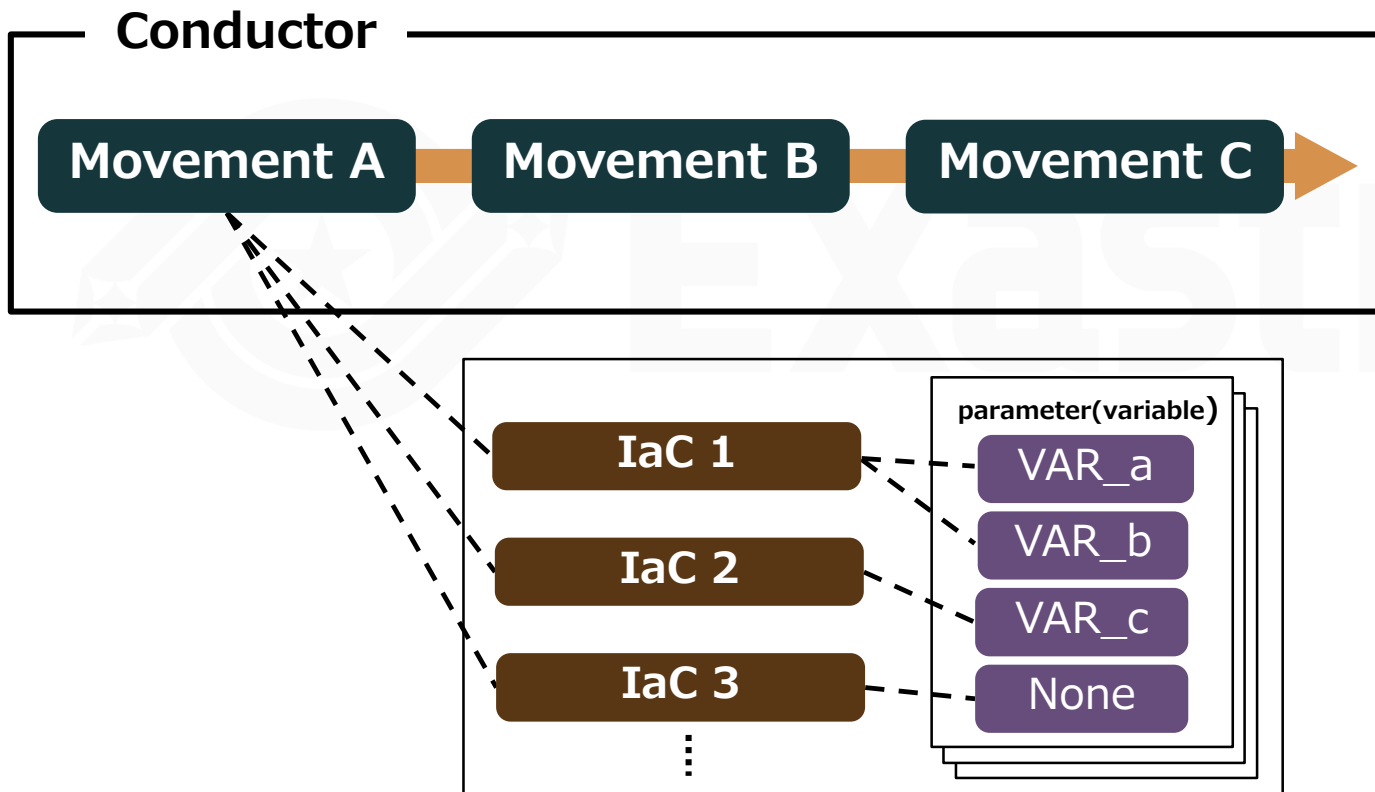

[Contact administrator](#)

## 2. About Conductor



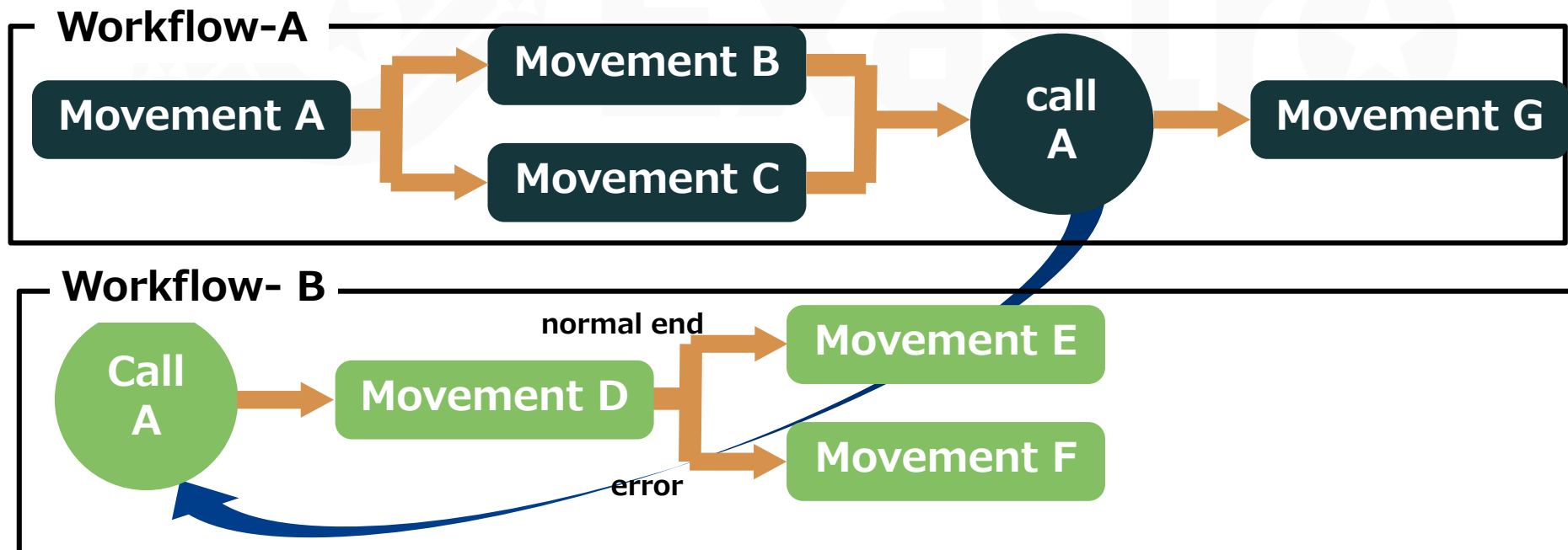
## 2.1 About Conductor

- Conductor is a function added in ITA Ver. 1.5.0
- Conductor allows users to specify multiple Movements into one sequence and links it to an operation before executing it.



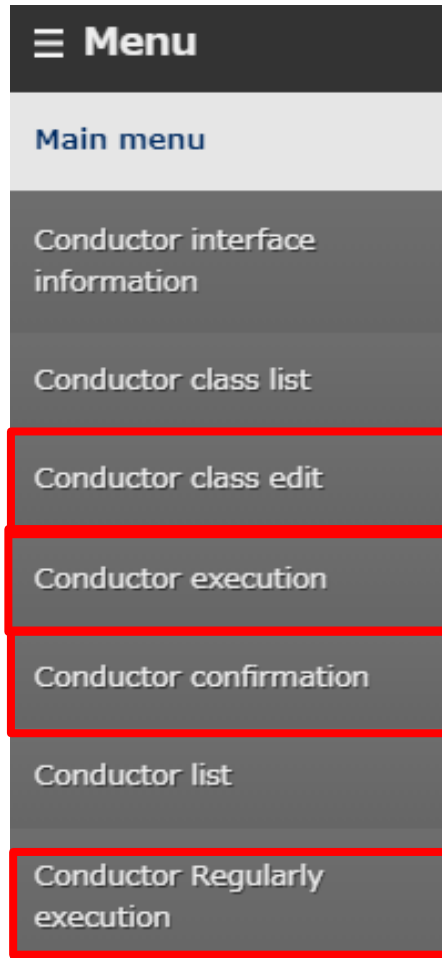
## 2.2 Conductor features

- While the Conductor Function has ,in similarity to the Symphony function, an execution function, it also contains the following functions.
- Hence, Conductor allows for execution of more complicated jobflows.
- **Parallel movement executions**
  - **Ability to call other jobflows**
  - **Conditional branching according to the execution result of movement**



## 2.3 Conductor Function Description (1/14)

- The main menus the Conductor menu group and their functions are as following



### ① Conductor class edit

Create an operation using previously created movement.

### ② Conductor execution

Execute operations.

### ③ Conductor confirmation

Confirm previously created operations.

### ④ Conductor Regularly execution

Register operations and configure regularly executed jobflows.



## 2.3 Conductor Function Description (2/14)

### ●Conductor class edit(1/4)

- In the "Conductor class edit" menu, Movements and different functions can be added and deleted.

The screenshot displays the 'Conductor class edit' interface. At the top, a menu bar includes 'EDIT', 'New', 'Save', 'Read', 'Cancel', 'Redo', and 'Delete node'. The main workspace features a grid with three components: a 'Start' node (blue circle with 'S'), a central 'Legacy1' node (red circle with '1' and 'Ansible Legacy' text), and an 'End' node (blue circle with 'E'). Red arrows indicate connections between these nodes. A right-hand panel shows a 'Movement' selection interface with a 'Filter' field and a list of movements. A red box highlights the 'Movement' column header, and another red box highlights a specific movement entry in the list. A third red box highlights the 'Function' column header. A large blue star is visible in the background of the right panel.

**Movements can be linked by dragging a line between the "in"/"out" circles.**

**Choose between various Functions.**

**Arrange Movement by dragging and dropping.**

## 2.3 Conductor Function Description (3/14)

### ●Conductor class edit(2/4)

- Users can use the Conditional branch function by selecting it from the “Function” tab on the right side of the screen.

The screenshot displays a software interface for editing conductor functions. On the left, a workflow diagram is shown on a grid background. It starts with a blue circular node labeled 'S' (Start) connected to a black rectangular node labeled 'Conductor Start'. This is followed by a series of black rectangular nodes with '+' signs, leading to two red circular nodes labeled '1' and '3'. Node '1' is labeled 'Ansible Legacy' and 'Legacy1'. Node '3' is labeled 'Ansible Legacy' and 'Legacy\_movement'. A red arrow points from node '3' to the 'Function' tab in the right-hand panel. The right-hand panel has a 'Function' tab selected, showing a list of functions: 'Conductor end', 'Conductor pause', 'Conductor call', 'Conditional branch', 'Parallel branch', and 'Parallel merge'. A blue callout box points to the 'Function' tab, stating: 'The arrangement of the “Function” tab can be changed by dragging and dropping them'. Another blue callout box points to the list of functions, stating: 'Users can choose between different Functions.' A third blue callout box points to the workflow diagram, stating: 'Similar to Movements, Operations can also be linked by dragging a line between them.'

Similar to Movements, **Operations** can also be linked by dragging a line between them.

The arrangement of the “Function” tab can be changed by dragging and dropping them

Users can choose between different Functions.

## 2.3 Conductor function description (4/14)

### ●Conductor class edit(3/4)

- By selecting multiple Nodes, users can use the buttons in the menu on the left to align the items. To select multiple Nodes, drag the mouse while holding the mouse button and drag over the Nodes you want to select or hold the shift key and press the Nodes you want selected individually.

For more information about the Node tab, please refer to [this manual](#).

**Selecting multiple Nodes will display the Object alignment function.**

**Use this tab to align the selected objects to your liking.**

## 2.3 Conductor menu functions (5/14)

### ●Conductor class edit(4/4)

- The following explains the different functions available.  
For more details, please refer to [this manual](#).









Figure <sup>↗</sup>	Name <sup>↗</sup>	Description <sup>↗</sup>
	Conductor start <sup>↗</sup>	Start of Conductor <sup>↗</sup>
	Conductor end <sup>↗</sup>	End of Conductor <sup>↗</sup> ※ If there are multiple Conductor end, the operation will end until all Conductor end are complete. <sup>↗</sup>
	Conductor pause <sup>↗</sup>	Pauses the Jobflow temporarily. <sup>↗</sup> Cancel the pause to move on to next step. <sup>↗</sup>
	Conductor call <sup>↗</sup>	Calls another registered Conductor class and executes it. <sup>↗</sup>
	Conditional branch <sup>↗</sup>	Branch process according to the result of "Movement" and "Conductor call" that the Node connects to. <sup>↗</sup> Status that can be specified is as follows. <sup>↗</sup> <ul style="list-style-type: none"><li>• Normal end<sup>↗</sup></li><li>• Abnormal end<sup>↗</sup></li><li>• Emergency stop<sup>↗</sup></li><li>• Preparation error<sup>↗</sup></li><li>• Unexpected error<sup>↗</sup></li><li>• SKIP complete<sup>↗</sup></li></ul>
	Parallel branch <sup>↗</sup>	Execute "Movement" or "Conductor call" in parallel. <sup>↗</sup> ※ The maximum parallel process number depends on the configuration and server spec of ITA. <sup>↗</sup>

Figure <sup>↗</sup>	Name <sup>↗</sup>	Description <sup>↗</sup>
	Parallel merge <sup>↗</sup>	Execute all process when all Nodes connected to this Node are finished. <sup>↗</sup>
	Movement <sup>↗</sup>	Execute Movement <sup>↗</sup>

## 2.3 Conductor Function Description(6/14)

### ●Conductor execution(1/2)

- Choose and execute the created Conductor in the "Conductor execution" menu.

Description

Scheduling

Specify the scheduled date/time in (YYYY/MM/DD HH:MM) Immediately execute when blank.

Scheduled date/time

Use this box to schedule when you want the Conductor to run.

Conductor [filter] ▾Open

Conductor [List] △Close

Select	Conductor class ID	Conductor name	Explanation	Remarks	Last update date/time	Last updated by
<input type="radio"/>	1	Sample1			2020/09/02 18:28:03	System Administrator
<input type="radio"/>	2	Test			2020/10/08 09:26:00	System Administrator

Filter result count: 2

Select Conductor and Operation.

Operation [Filter] ▾Open

Operation [List] △Close

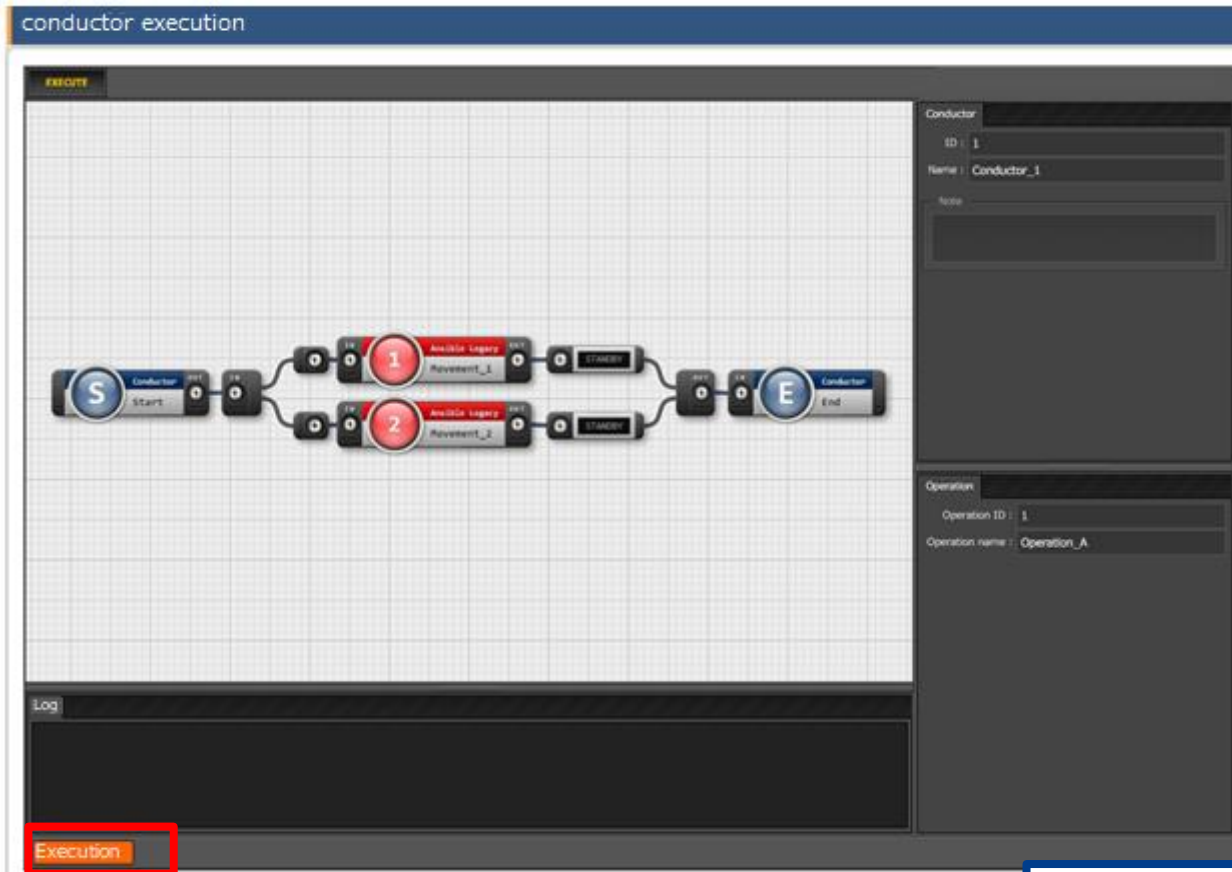
Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Remarks	Last update date/time	Last updated by
<input type="radio"/>	1	1	Operation1	2020/08/27 16:15	2020/10/15 10:31		2020/10/15 10:31:18	Legacy execution procedure
<input type="radio"/>	2	2	Test Operation	2020/10/08 10:00	2020/10/16 13:45		2020/10/16 13:45:19	Legacy execution procedure

Filter result count: 2

## 2.3 Conductor Function Description (7/14)

### ●Conductor execution (2/2)

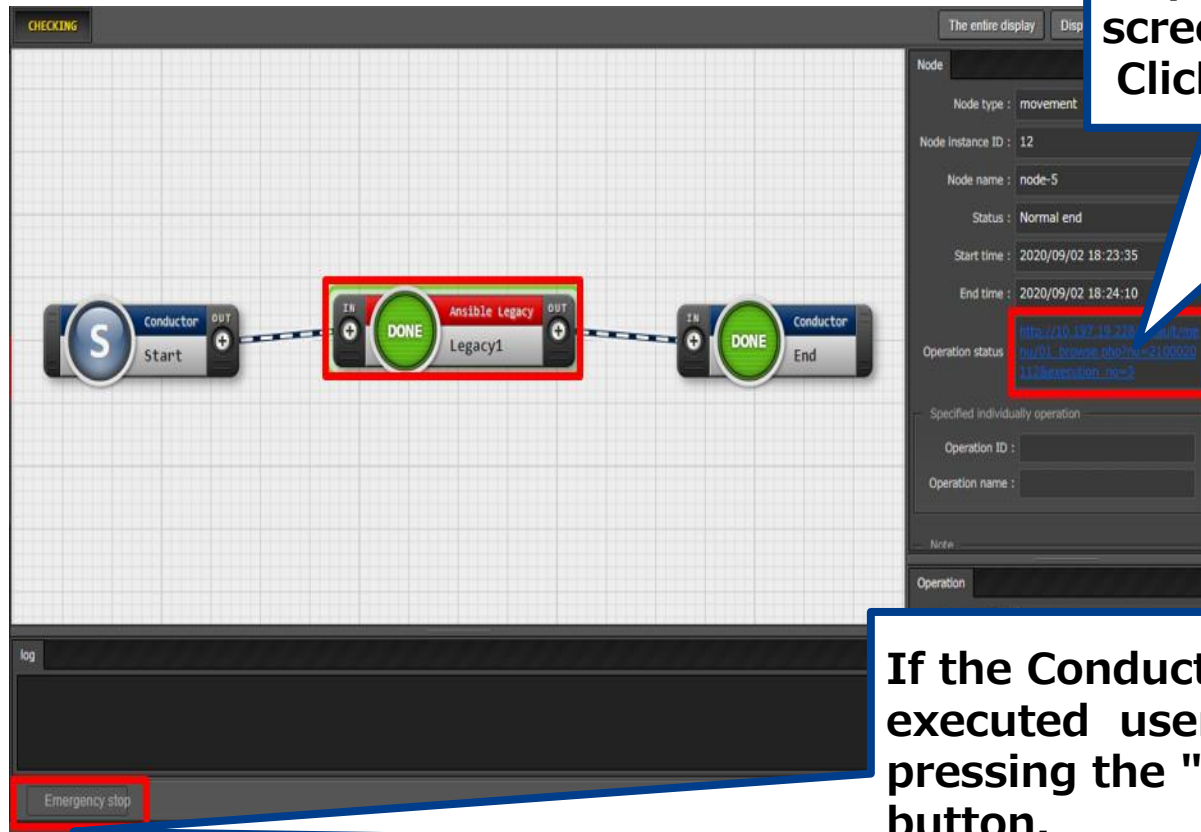
- The conductor and operation selected at the top of the page will be displayed.



If there are no problems with the contents, press the "Execute" button to execute.

## 2.3 Conductor (8/14)

- **Conductor confirmation**
- Check the execution status from the "Conductor Confirmation Menu".



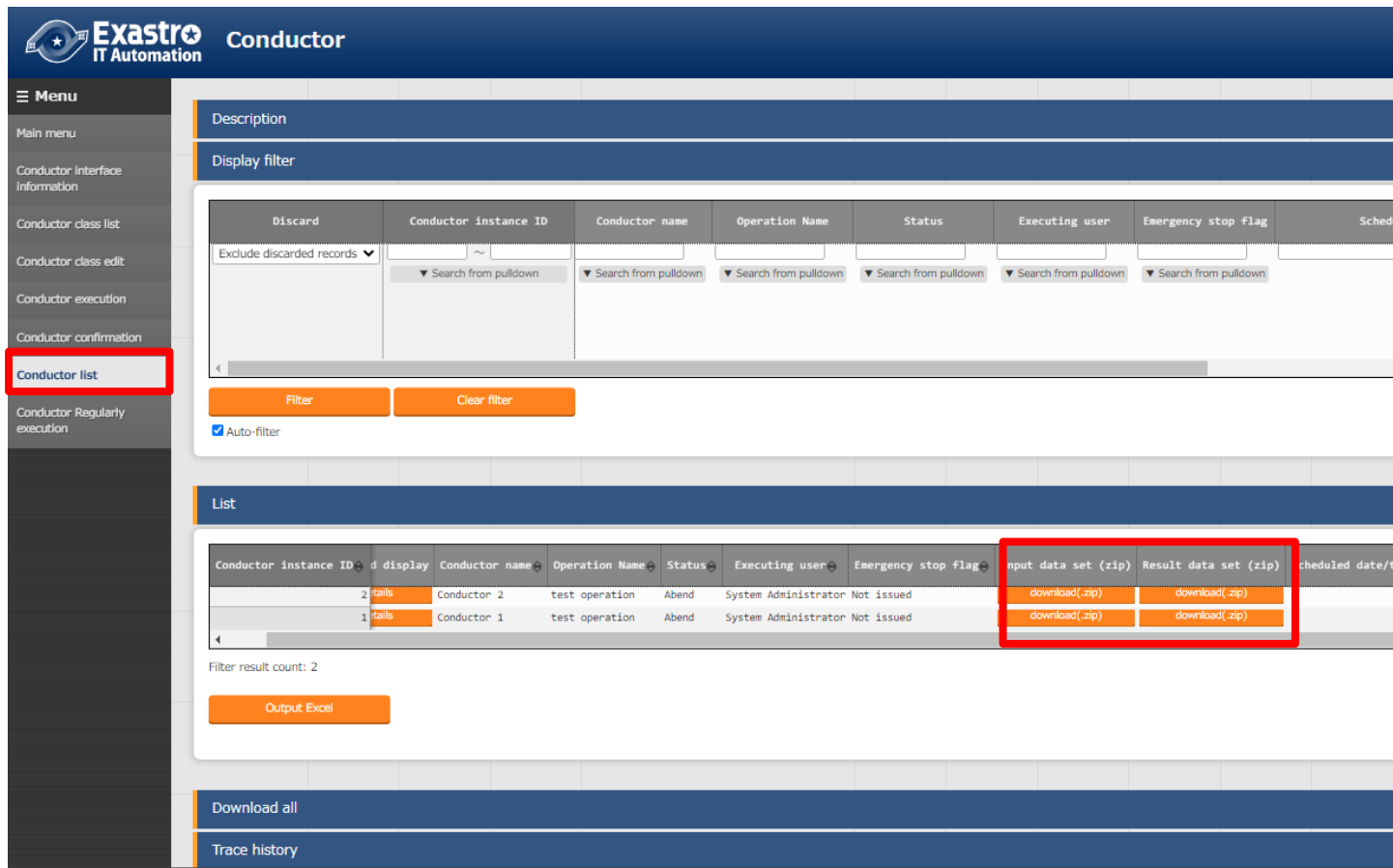
Clicking the movement will display a detailed results screen.  
Click [here](#) for more details.

**If the Conductor is scheduled to be executed users can cancel it pressing the " schedule cancelling" button.**

## 2.3 Conductor Function Description (9/14)

### ●Conductor check(1)

In the “Conductor” Menu group -> “Conductor list” menu -> “List” Sub-menu, users can download the input/results data for each Conductor.



The screenshot displays the Exastro IT Automation Conductor interface. The left sidebar shows the 'Conductor list' menu item highlighted with a red box. The main content area shows the 'List' sub-menu, which contains a table of conductor instances. The table has columns for Conductor instance ID, Display, Conductor name, Operation Name, Status, Executing user, Emergency stop flag, Input data set (zip), Result data set (zip), and Scheduled date/time. The 'Input data set (zip)' and 'Result data set (zip)' columns contain 'download(.zip)' buttons, which are highlighted with a red box. Below the table, there is a 'Filter result count: 2' and an 'Output Excel' button. At the bottom, there are 'Download all' and 'Trace history' buttons.

Conductor instance ID	Display	Conductor name	Operation Name	Status	Executing user	Emergency stop flag	Input data set (zip)	Result data set (zip)	Scheduled date/time
2	tails	Conductor 2	test operation	Abend	System Administrator	Not issued	download(.zip)	download(.zip)	
1	tails	Conductor 1	test operation	Abend	System Administrator	Not issued	download(.zip)	download(.zip)	



## 2.3 Conductor Function Description (10/14)

### ●Conductor Routine Executions(1/3)

- In the "Conductor Regularly execution" menu, users can manage regularly executed operations.

The screenshot displays the 'Conductor Routine Executions' interface. At the top is a 'Display filter' section with a table of filters and search buttons. Below this is a 'List/Update' section with a 'Register' button. The main table lists periodic work executions, and a 'Schedule settings' button is highlighted in the first row.

**Display filter**

Discard	Periodic work execution ID	Conductor class name	Operation name	status	Next execution date	Last update date/time	Last updated by
Exclude discarded records ▼	~				~	~	
	▼ Search from pulldown	▼ Search from pulldown	▼ Search from pulldown	▼ Search from pulldown			▼ Search from pulldown

**Filter** **Clear filter**

☒ Auto-filter

**List/Update** **Open**

**Register** **Close**

Periodic work execution ID	Conductor class name	Operation name	status	Schedule settings	Next execution date	Start date	End date	pe	Last update date/time	Last updated by
Auto-input			Auto-input	<b>Schedule settings</b>	Auto-input				Auto-input	Auto-input

※\*is a required item.

**Back** **Register**

**A detailed schedule can be set from the "Schedule Settings" button.**

## 2.3 Conductor Function Description (11/14)

### ●Conductor Routine Executions(2/3)

- "Schedule settings" allows user to set detailed settings such as the regular execution period and the period for stopping work.

**Set a schedule**

Work period  
\* Start date: 2020/10/19 11:00 End date: 2020/10/29 23:27

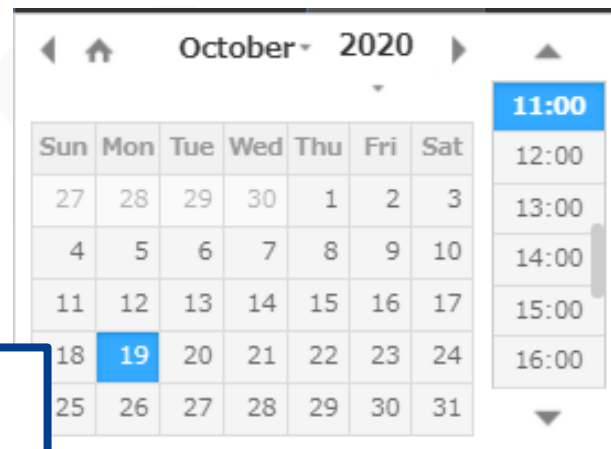
Schedule  
☒ Time  
☐ Day  
☐ Week  
☐ Month(Specify day)  
☐ Month(Specify day of week)  
☐ End of month

\* Interval: 5 Time

Work suspension period  
2020/10/06 23:00

Note

Clicking here will display a calendar.



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の実行日時は以下ようになります。

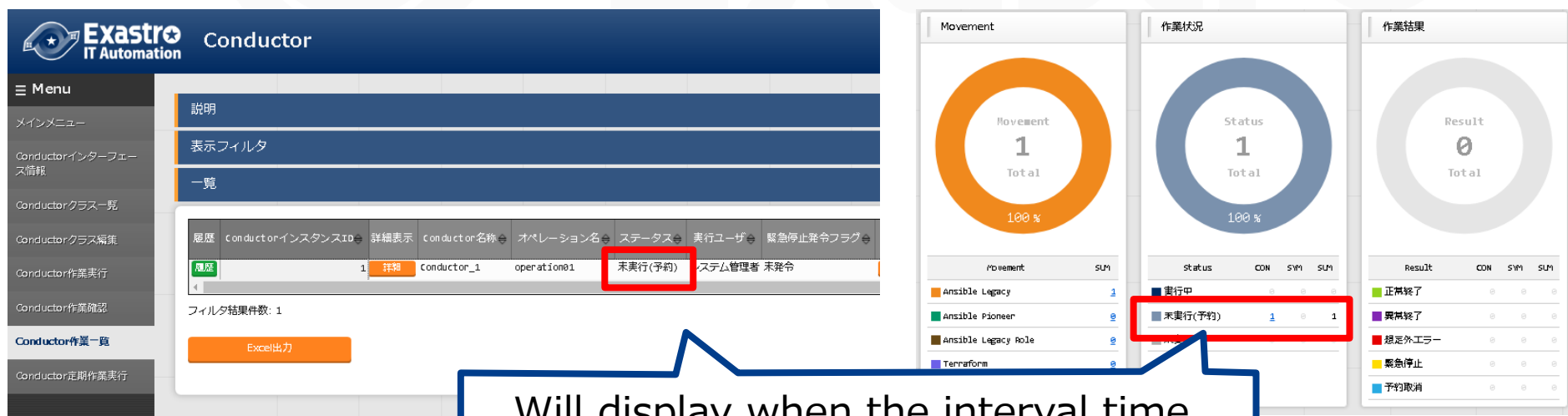
初回(=次回実行日付) : 2021/09/01 00:00  
2回目 : 2021/09/01 08:00  
3回目 : 2021/09/01 16:00

## 2.3 Conductor Function Description (12/14)

### ●Conductor Routine Executions (3/3)

- Any Conductor operations registered with a scheduled routine will display "Not executed (Reserved)" when the first interval time before the first scheduled execution date passes. (The default interval (time) value is 3 minutes). The Interval time can be changed by the user in the "Management console" > "System settings" menu. For more information, please see the [Management console user manual](#)
- Conductor/Symphonies with the status "Unexecuted (Schedule)" can be seen in the "Work status" field in the dashboard and in the "Conductor list" menu.

If the next execution date is set to "2021/9/1 18:00" and the interval is set to 3 minutes, the status will change to "Unexecuted (Schedule)" 17:57 2021/9/1.

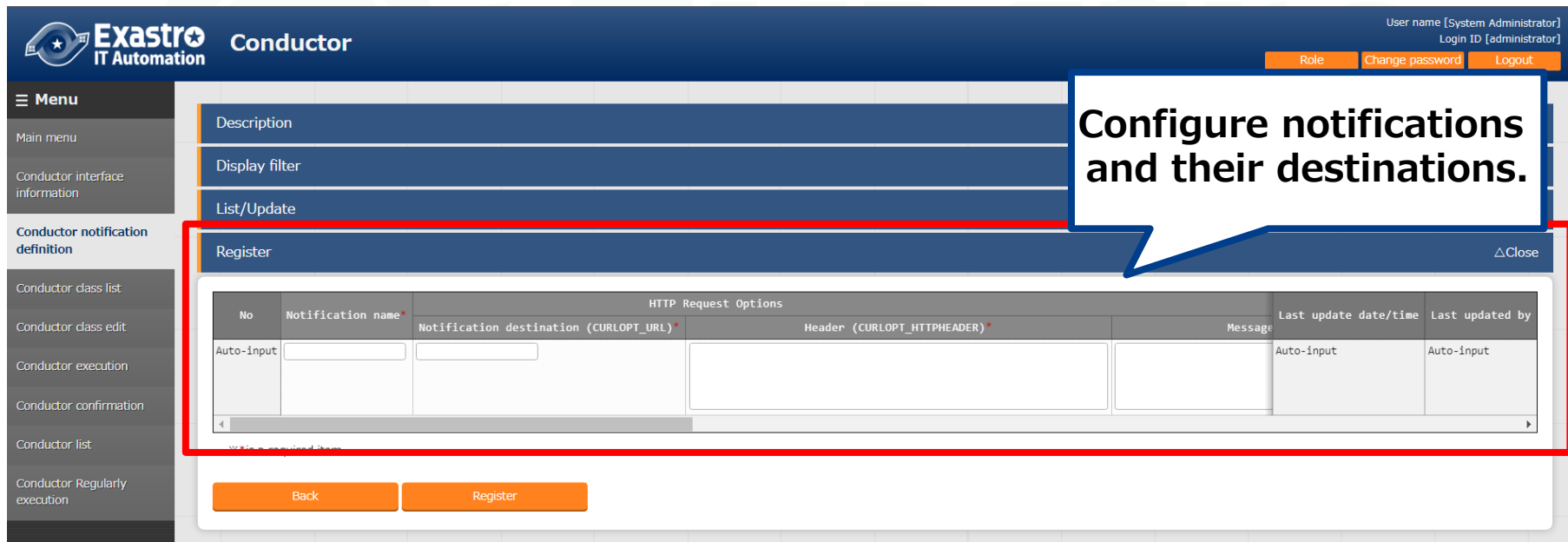


Will display when the interval time before the next execution date passes.

## 2.3 Conductor Function Description (13/14)

### ●Conductor Notification destination definition(1/2)

- This allows users to send Conductor execution results to specified message tools (Teams, Slack) through http requests. Use the "Conductor notification destination definition" menu to configure receiving messaging tools and define notification contents. For more information, see the [manual](#).
- This tool is especially helpful for executions with long processing times and for reserved/scheduled operations.



The screenshot displays the Exastro Conductor web interface. The top navigation bar includes the Exastro logo, the word 'Conductor', and user information (User name [System Administrator], Login ID [administrator]). On the right, there are links for 'Role', 'Change password', and 'Logout'. A left sidebar menu lists various options, with 'Conductor notification definition' highlighted. The main content area shows a 'Register' form for defining notification destinations. A red box highlights this form, and a blue callout bubble points to it with the text 'Configure notifications and their destinations.' The form contains a table with the following columns: No, Notification name, Notification destination (CURLOPT\_URL), HTTP Request Options (Header (CURLOPT\_HTTPHEADER), Message), Last update date/time, and Last updated by. The table has one row with the value 'Auto-input' in the 'No' column and 'Auto-input' in the 'Last update date/time' and 'Last updated by' columns. Below the table are 'Back' and 'Register' buttons.

No	Notification name	Notification destination (CURLOPT_URL)	HTTP Request Options Header (CURLOPT_HTTPHEADER)	Message	Last update date/time	Last updated by
Auto-input					Auto-input	Auto-input

## 2.3 Conductor Function Description (14/14)

### ●Conductor Notification destination definition(1/2)

- Users can specify Conductor notification destination definitions registered in the "Conductor class edit" > "Notice" menu, and select what Conductor status sends notifications.

The screenshot shows the 'Conductor class edit' interface. On the left is a 'Menu' sidebar with options: Main menu, Conductor interface information, Conductor notification definition, Conductor class list, **Conductor class edit**, Conductor execution, Conductor confirmation, Conductor list, and Conductor Regularly execution. The main area is a grid. A blue callout box points to a 'Select' button in the top right corner of the grid area, with the text 'Click this button to display the "Notice list".' A red arrow points from this button to the 'Notice list' dialog box shown below.

**Notice list**

Notification name	Executing	Executing (delay)	Normal end	Warning end	Emergency stop	Abend	Unexpected error
Notification sample	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Decision Cancel

## 2.4 Conductor workflow

- The Conductor workflow is as follows.  
Details can be found in the Practice document.

① Register device information

Basic console menu

② Register operation

③ Register Movement

Various driver menus

④ Check Movement

⑤ Register interface information

Conductor menu

⑥ Register Conductor

⑦ Check Conductor

⑧ Execution Conductor

⑨ Check execution result

⑩ Check execution history



**Exastro**