



IT Automation Conductor 【Tutorial】

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

Table of contents

1. Introduction

1. About this documents

2. About description of Conductor

1. About Conductor

2. Conductor feature

3. Description of the functions in Conductor Menu

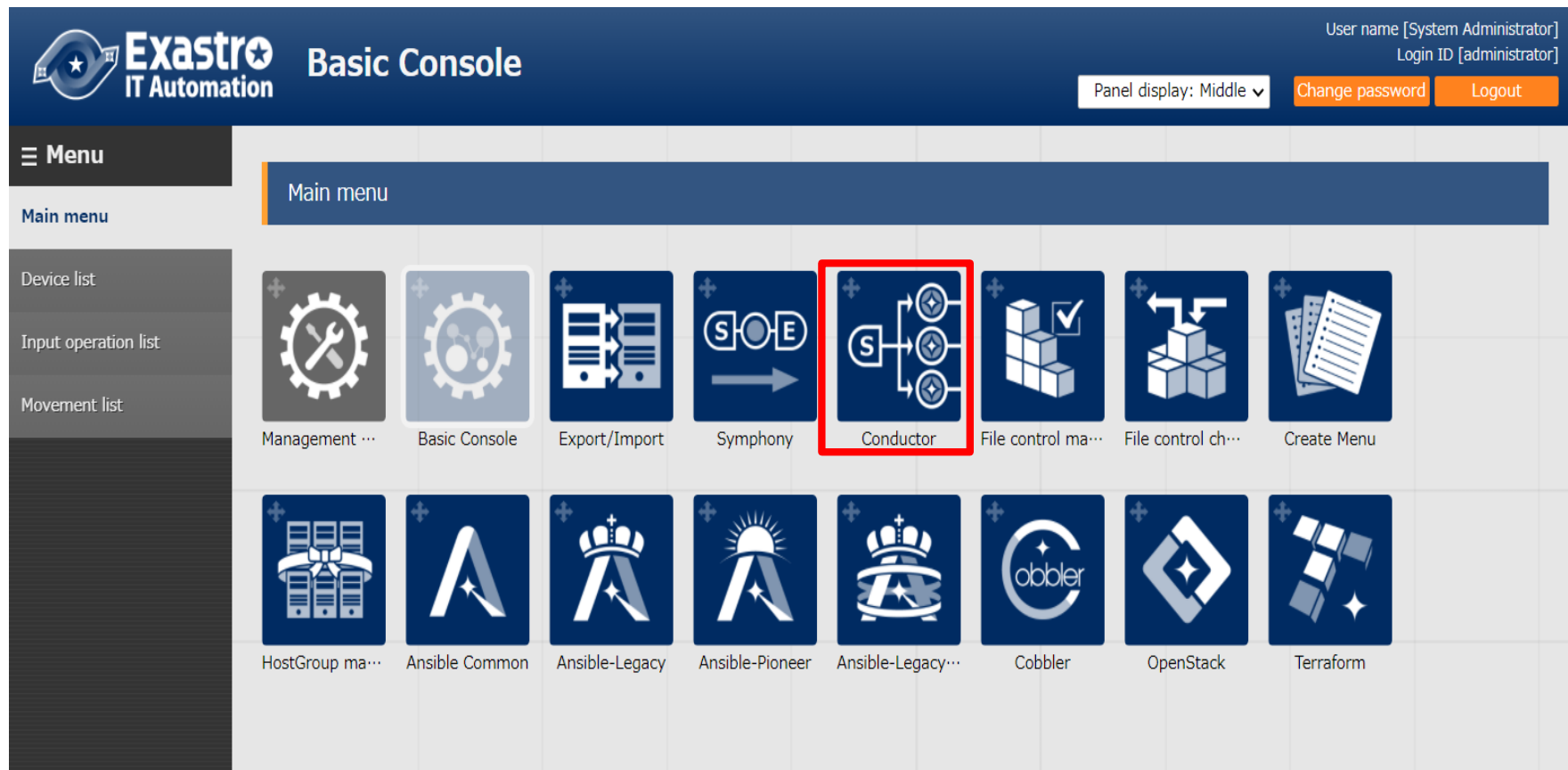
4. Conductor workflow

1. Introduction

1.1 About this document

Main menu

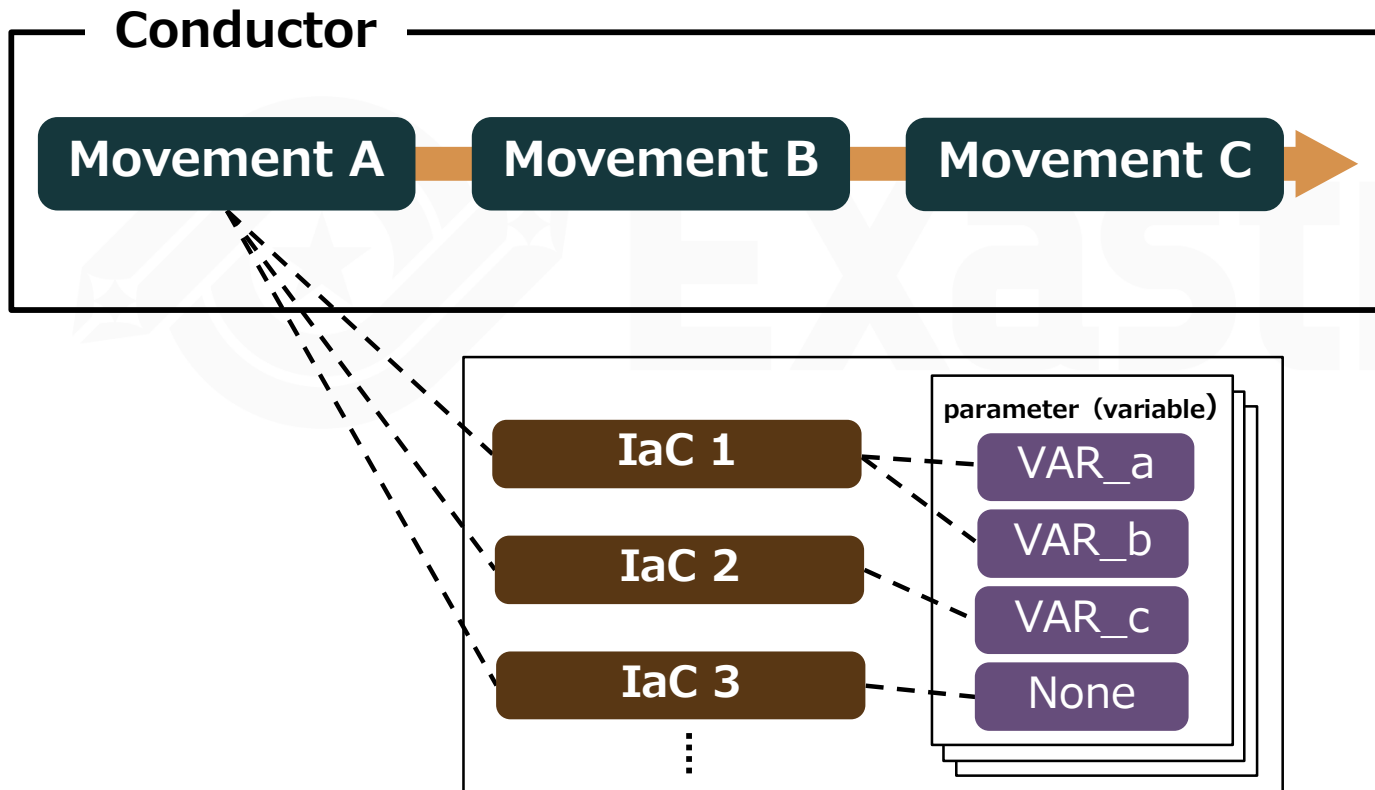
- About the "Conductor" of menu group are explained in this document.



2. Description about Conductor

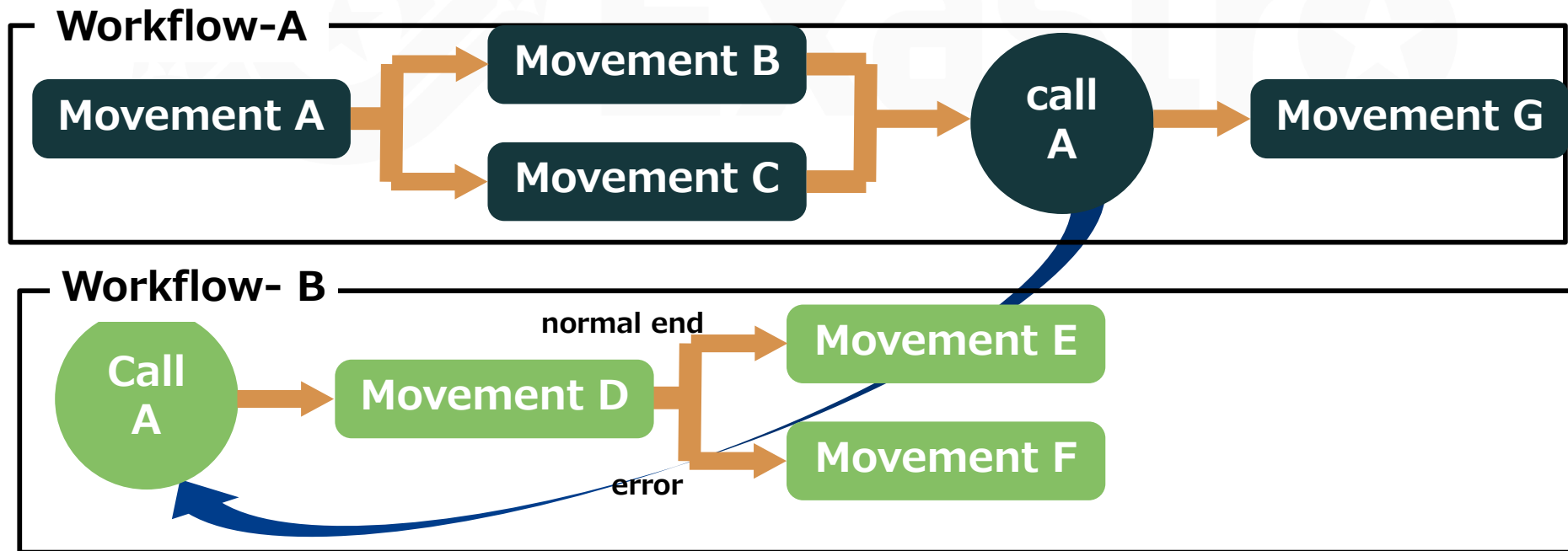
2.1 About Conductor

- Conductor is a function added to ITA from ver1.5.0.
- Conductor refers to a single unit of a series of work in ITA and execute in association with the operation name. (workflow)

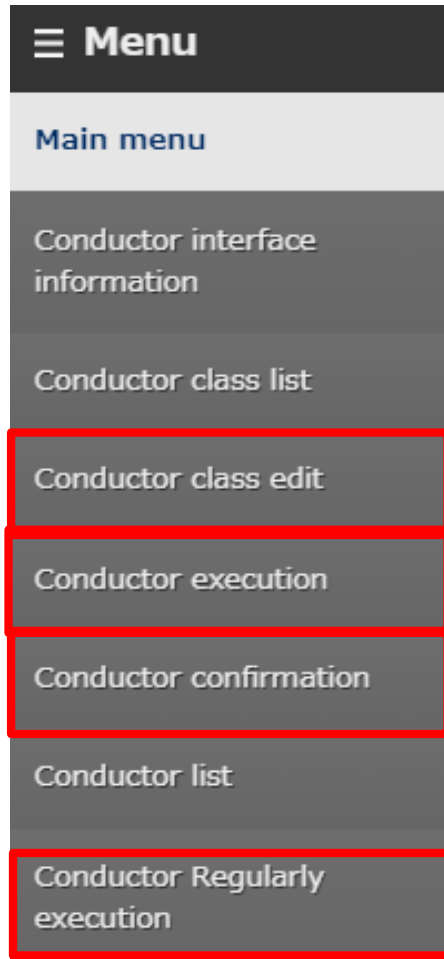


2.2 Conductor features

- In conductor, users can prepare work execution functions similar to the Symphony functions, as well as the functions listed below.
- According to this Conductor allows to perform more sophisticated job follows.
 - **Parallel execution of movement**
 - **Call another job flow**
 - **Conditional branching according to the execution result of movement**



● Introduction of main functions in the Conductor Menu



① **Conductor class edit**

Create an operation using the a created movement.

② **Conductor execution**

Executing the operation.

③ **Conductor confirmation**

Confirm the created operation.

④ **Conductor Regularly execution**

Register an operation and performs execution period.

●Conductor class edit (1/3)

- In the "Conductor class edit" menu Movement and Functions that perform various controls can be added and deleted.

The screenshot shows the 'Conductor class edit' interface. At the top, there is a menu bar with buttons: EDIT, New, Save, Read, Cancel, Redo, Delete node, The entire display, Display reset, and full screen. The main workspace is a grid where three components are connected: a 'Start' node (blue circle with 'S'), an 'Ansible Legacy' node (red circle with '1'), and an 'End' node (blue circle with 'E'). Red arrows indicate the flow between them. A callout box points to the 'Ansible Legacy' node with the text: 'Movements can be combined by dragging and dropping in/out for each movement.'

On the right side, there is a panel with a 'Movement' section. It has a 'Function' dropdown menu and a 'Filter' input field. Below this is a table with the following data:

| | ID | Movement name |
|---|----|-----------------|
| + | 1 | Legacy1 |
| | 3 | Legacy_movem... |
| | 5 | Move1 |
| | 4 | move |

A callout box points to the 'Function' dropdown and the table with the text: 'Choose between various Functions.'

At the bottom left, there is a 'log' section and a 'Registration' button. A callout box points to the 'Ansible Legacy' node with the text: 'Arrange Movement by dragging and dropping.'

●Conductor class edit (2/3)

- From the tab near the center right of the screen, select and use the function that controls the conditional branching of the operation.

The screenshot displays the Conductor Menu interface. At the top, a menu bar includes 'EDIT', 'New', 'Save', 'Read', 'Cancel', 'Redo', and 'Delete node'. The main workspace shows a workflow diagram on a grid. It starts with a 'Conductor Start' node (blue circle with 'S'), followed by a series of '+' nodes. A red circle with the number '1' is connected to an 'Ansible Legacy' node (labeled 'Legacy1'). A red arrow points from a red circle with the number '3' to a 'Legacy_movement' node. A blue callout box points to the '1' node with the text: 'Similar to Movement, **operation** can be combined in/out by dragging and dropping.' On the right, a 'Function' menu is open, listing several options: 'Conductor end', 'Conductor pause', 'Conductor call', 'Conditional branch', 'Parallel branch', and 'Parallel merge'. A red box highlights this menu, and a blue callout box points to it with the text: 'Choose between various Functions.' At the bottom left, a 'log' window and a 'Registration' button are visible. A blue callout box points to the workflow diagram with the text: 'The function arrangement can be changed by dragging and dropping.'









Similar to Movement, **operation** can be combined in/out by dragging and dropping.

The function arrangement can be changed by dragging and dropping.

Choose between various Functions.

●Conductor class edit (3/3)

- Possible to describes the functions. For details refer to menu.

| Figure | Name | Description |
|---|--------------------|--|
|  | Conductor start | Start of Conductor |
|  | Conductor end | End of Conductor ※If there are multiple Conductor end, the operation will end until all Conductor end is achieved. |
|  | Conductor pause | Pause the workflow temporary. Cancel the pause to move on to next step. |
|  | Conductor call | Call another register Conductor class and execute it. |
|  | Conditional branch | Branch process according to the result of "Movement" and "Conductor call" that the Node connects to. Status that can be specified is as follows. <ul style="list-style-type: none"> • Normal end • Abend • Emergency stop • Preparation error • Unexpected error • SKIP complete |
|  | Parallel branch | Execute "Movement" or "Conductor call" in parallel. ※The maximum parallel process number depends on the configuration and server spec of ITA. |
|  | Parallel merge | Execute all process when all Nodes connected to this Node are finished. |
|  | Movement | Execute Movement |

2.3 Description of the functions in Conductor Menu (5/9)

●Conductor execution(1/2)

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

Description

▽Open

Scheduling

Possible to schedule execution timing.

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

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 |

Select operation and executed Conductor.

Filter result count: 2

Operation [Filter]

▽Open

Operation [List]

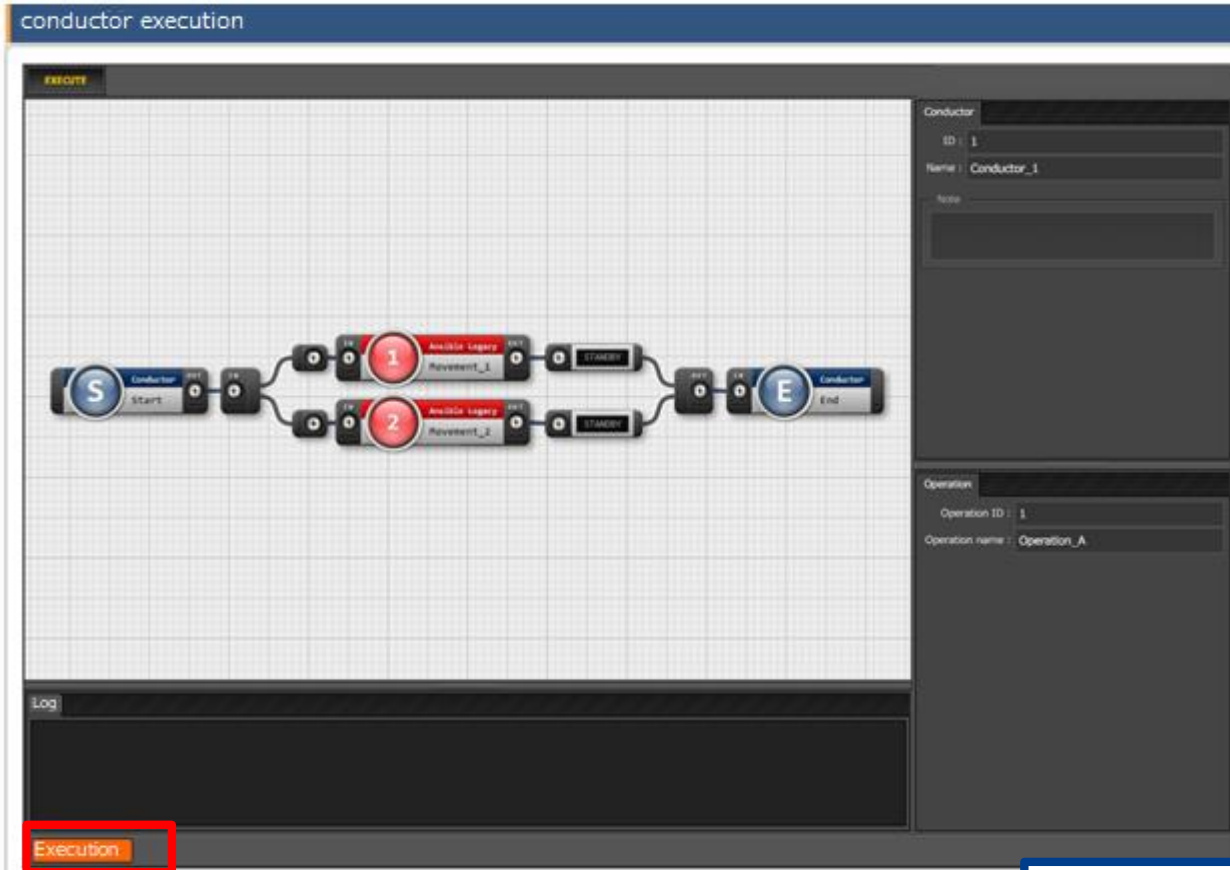
△Close

| 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

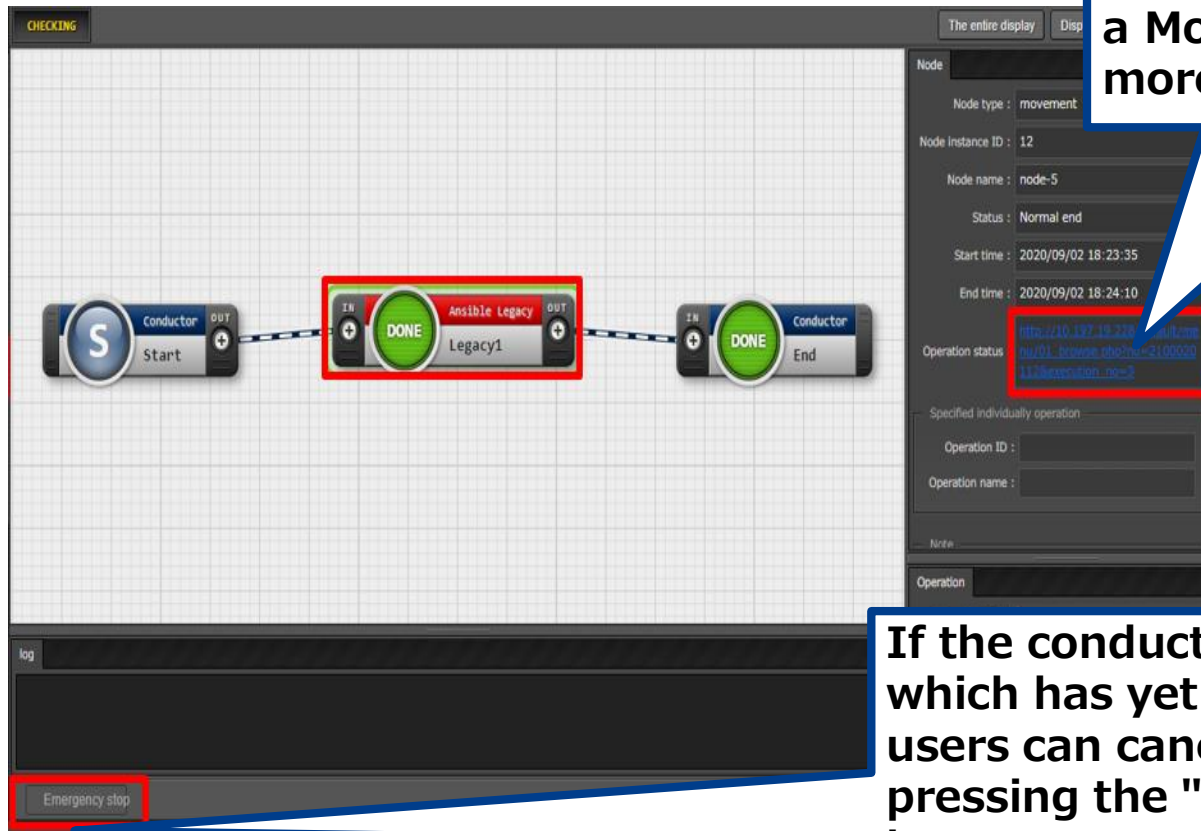
●Conductor execution (2/2)

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



If there is no problem with the contents, press the "Execute" button to execute.

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



Check a detailed status of the execution results by pressing a Movement. Click here for more details.

If the conductor has a set schedule which has yet to be executed, users can cancel the schedule by pressing the " Reservation cancelling" button.

●Conductor Regularly execution (1/2)

- In the "Conductor Regularly execution" menu, manage work schedule to be executed regularly.

The screenshot displays the 'Conductor Regularly execution' interface. At the top is a 'Display filter' panel with a 'Close' button. Below it is a table with columns: Discard, Periodic work execution ID, Conductor class name, Operation name, status, Next execution date, Last update date/time, and Last updated by. Each column has a search input field with a 'Search from pulldown' button. Below the table are 'Filter' and 'Clear filter' buttons, and a checked 'Auto-filter' checkbox. Below this is a 'List/Update' section with an 'Open' button, and a 'Register' section with a 'Close' button. At the bottom is a table with columns: Periodic work execution ID, Conductor class name, Operation name, status, Schedule settings, Next execution date, Start date, End date, pe, Last update date/time, and Last updated by. The 'Schedule settings' column has a red button labeled 'Schedule settings'. Below the table is a note '※*is a required item.' and 'Back' and 'Register' buttons.

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

●Conductor Regularly execution (2/2)

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

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 23:34 ~ 2020/10/14 23:34

Note

* is a required field

OK Close

Clicking it will display a calendar.

October 2020

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 27 | 28 | 29 | 30 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 |

11:00
12:00
13:00
14:00
15:00
16:00

2.4 Conductor workflow

- Conductor workflow is as follows.
Actual operation is described in the training.

① Register device information

Basic console menu

② Register operation

③ Register Movement

Various driver menu

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