



ITA\_User Instruction Manual

Conductor

*－*Version 1.11*－*

Copyright © NEC Corporation 2020. All rights reserved.

Disclaimer

All the contents of this document are protected by copyright owned by NEC Corporation.

Unauthorized reproduction or copying of all or part of the contents of this document is prohibited.

The contents of this document are subject to change without prior notice in the future.

NEC Corporation is not responsible for any technical or editorial errors or omissions in this document.

NEC Corporation do not guarantee accuracy, usability, certainty of the content in this document.

Trademark

* Linux is registered trademark or trademark of　Linus Torvalds, registered in the U.S. and other countries.
* Red Hat is registered trademark or trademark of Red Hat, Inc. registered in the U.S. and other countries.
* Apache, Apache Tomcat, and Tomcat are registered trademarks or trademarks of the Apache Software Foundation.
* Ansible is registered trademark or trademark of Red Hat, Inc.
* AnsibleTower is registered trademark or trademark of Red Hat, Inc.
* Terraform is registered trademark or trademark of HashiCorp.

The names of other systems, company name and products mentioned in this document are registered trademarks or trademarks of their respective companies.

The ® mark and TM mark are not specified in this document.

※「Exastro IT Automation」 is written as 「ITA」 in this document.

**Table of Contents**

[Introduction 4](#_Toc93482339)

[1 Overview of ITA Conductor 5](#_Toc93482340)

[2 ITA Conductor menu screen configuration 6](#_Toc93482341)

[2.1 ITA Conductor menu list 6](#_Toc93482343)

[3 ITA Conductor user instruction procedure 7](#_Toc93482344)

[3.1 Jobflow 7](#_Toc93482345)

[4 Function and operation method description 9](#_Toc93482346)

[4.1 ITA Conductor 9](#_Toc93482347)

[4.1.1 Conductor interface information 9](#_Toc93482348)

[4.1.2 Conductor notification definition 10](#_Toc93482349)

[4.1.3 Conductor class list 12](#_Toc93482350)

[4.1.4 Conductor class edit 13](#_Toc93482351)

[4.1.5 Conductor execution 27](#_Toc93482352)

[4.1.6 Conductor confirmation 30](#_Toc93482353)

[4.1.7 Conductor list 35](#_Toc93482354)

[4.1.8 Conductor regularly execution 36](#_Toc93482355)

[5 Appendix 44](#_Toc93482356)

[5.1 Conductor notification destination definition 44](#_Toc93482357)

[5.1.1 Conductor notification destination definition setting 44](#_Toc93482358)

[5.1.2 Notification log output exmaple 47](#_Toc93482359)

Introduction

This document describes the functions and operation methods of the ITA Conductor function.

# Overview of ITA Conductor

This chapter explains the functions and operation methods of the Conductor menu.

Conductor menu provides the following functions that are commonly required to perform work using ITA.

# ITA Conductor menu screen configuration

This chapter explains the menu and screen configuration of the ITA Conductor menu.



## ITA Conductor menu list

The ITA common/Conductor menu is shown below

**Table 2.1-1 ITA　Conductor screen list**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Menu**  **Group** | **Menu / Screen** | **Description** |
| 1 | Conductor | Conductor interface information | Maintain (View/Register/Update/Discard) settings such as shared directory path of Movement when executing Conductor |
| 2 | Conductor class list | Maintain (View/Discard) Conductor class.  Click “Details” to move to Conductor class edit menu. |
| 3 | Conductor class edit | Edit Conductor class |
| 4 | Conductor execution | Execute Conductor operation |
| 5 | Conductor confirmation | Check the result of Conductor operation execution |
| 6 | Conductor list | View the list Conductor (execution history)  Click “Details” to move to Conductor confirmation |
| 7 | Conductor regularly execution | Manage Conductor operations that executes routinely. |

# ITA Conductor user instruction procedure

## Jobflow

The standard ITA Conductor jobflow is as follows.

More detailed information regarding each operation is listed in the next section.

* + For information regarding registering Device information and Operations, please refer to "Exastro-ITA\_User\_Instruction\_Manual\_Basic-Console".
  + For information regarding registering Movements, please refer to the different driver's manuals.
  + It is possible to use the Movement’s shared directory path, even when Conductor is running.

If you need information to be delivered between Movements, you can do so by using a shared directory path.

Ansible\_Driver and Terraform\_Driver and Terraform\_CLI\_Driver can use this function.

For more information, please see “Exastro-ITA\_User\_Instruction\_Manual\_Ansible-driver” and “Exastro-ITA\_User\_Instruction\_Manual\_Terraform-driver”

and “Exastro-ITA\_User\_Instruction\_Manual\_Terraform-CLI-driver”

The workflows done in both "Conductor call" and "Symphony call" have their own common directory.

（Movements that strides over the workflows are not shared)

**①Register / check device information**

**②Register / check Operation**

**⑥Register Conductor**

**⑦Check Conductor**

**⑧Execute Conductor**

**⑨Check Conductor execution result**

**⑩Check Conductor execution history**

**④Check Movement**

**③Registering Movement in each driver of ITA**

**⑤Register Conductor interface information**

Basic console

Conductor

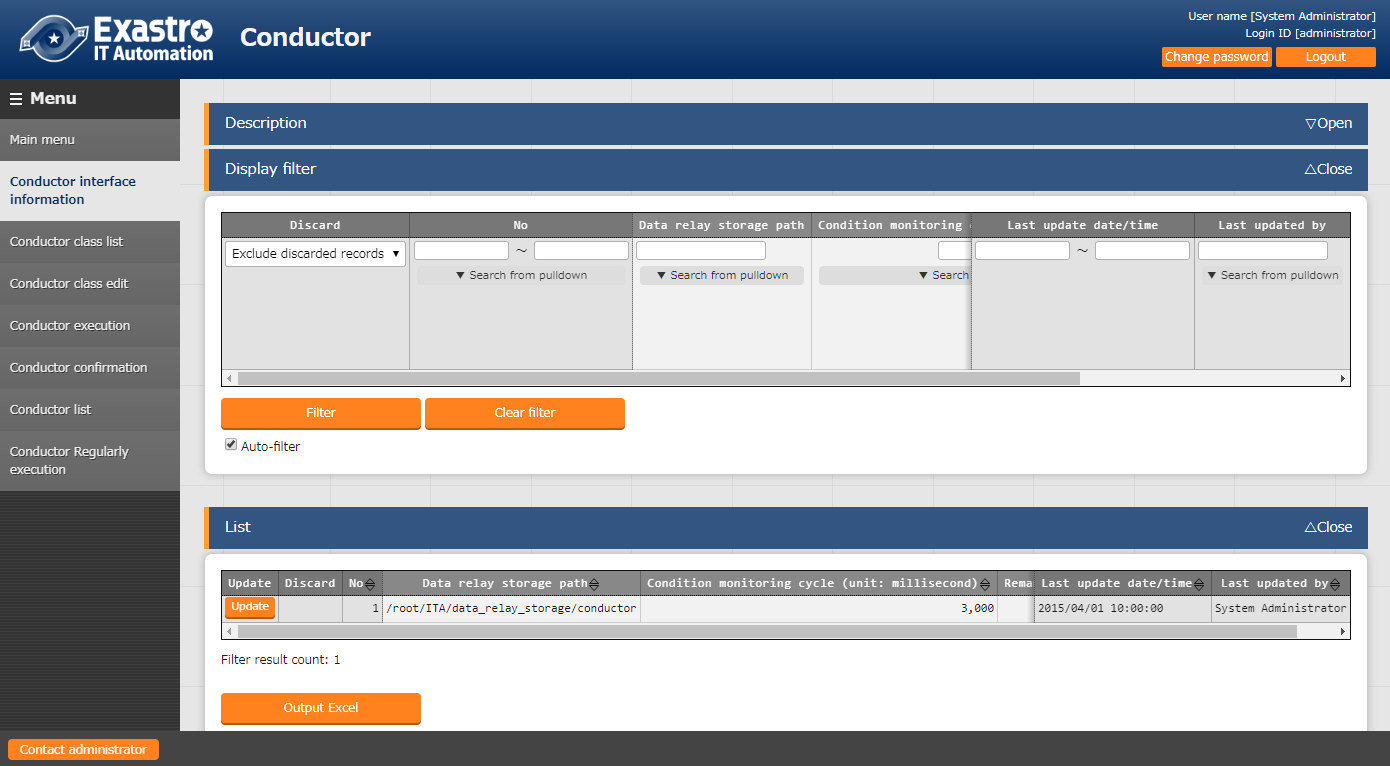
**Figure 3.1-1 Jobflow**

# Function and operation method description

## ITA Conductor

### Conductor interface information

1. In [Conductor interface information] screen, users can set the path of shared directory for each Movement executed by Conductor and the refresh interval for [Conductor confirmation] screen.



**Figure 4.1‑1 (Conductor interface information) Menu**

1. The list of common items on the registration screen is as follows.

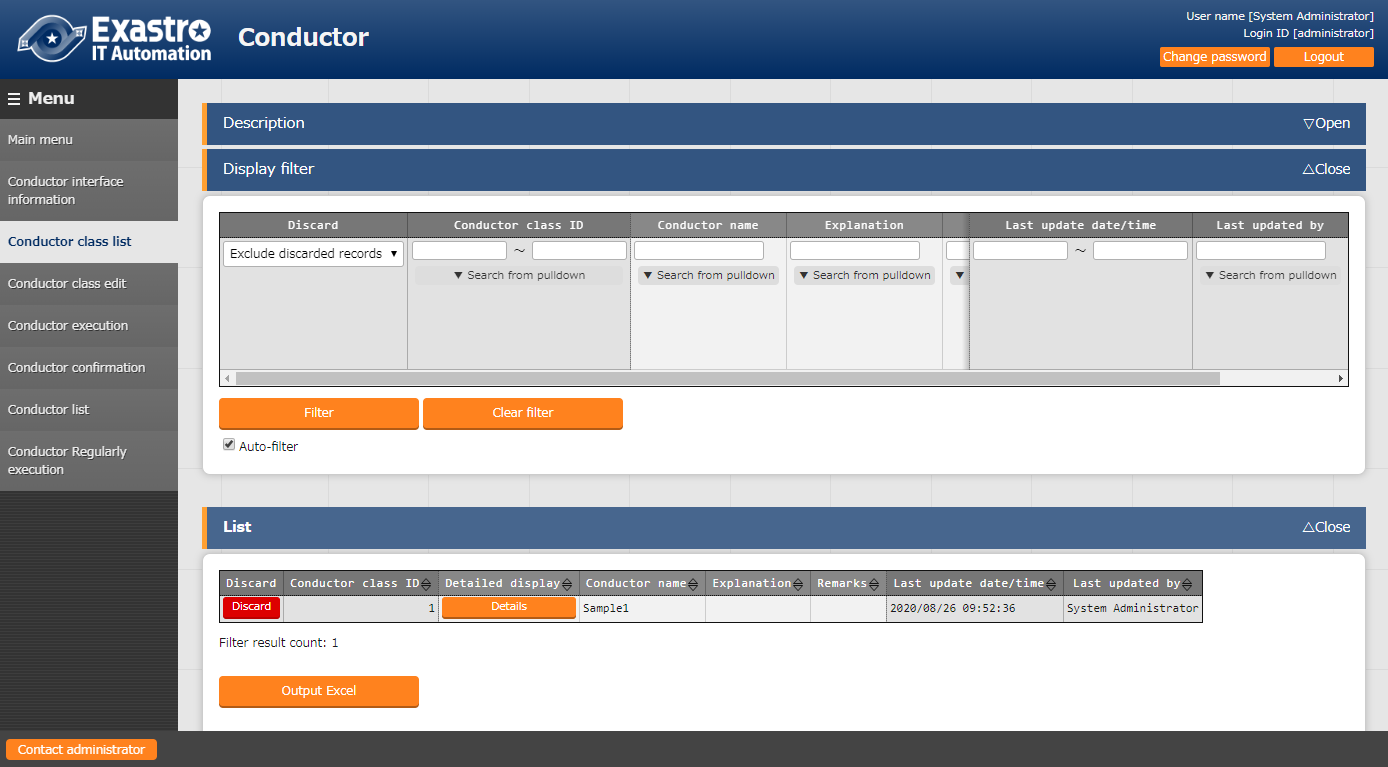
**Table 4.1-1　List of Registration Screen Items (Conductor interface information)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input**  **Required** | **Input** **type** | **Restrictions** |
| Data relay storage path | When executing Conductor, enter the directory shared by each Movement with the directory path viewed from the ITA server. For the path viewed from Ansible, and Ansible-Tower server, please refer to the interface information in the instruction manual for Ansible Driver. Connection between terraform is obtained from REST API without common directory. Therefore, Directory path is used in Terraform-Driver and Terraform-CLI-Driver. | ○ | Manual input | Maximum length 128 bytes |
| Status monitoring cycle (unit: millisecond) | Enter the interval for refreshing the display of “4.1.5 Conductor execution”. Generally, it is recommended to set the number to 3000 millisecond. | ○ | Manual input | Minimum value 1000ms |
| Remarks | Free description field | - | Manual input | - |

Conductor notification definition

In “Conductor notification definition” menu sets definition for notification performed in Conductor. Notification is sent using Webhook.

Registered Conductor notification definition sets notification status in “Notice” within “Conductor class list” during process.



admin@nec.amc.jp

~

~

~

~

:set bin noeol

**Figure 4.1‑2 (Conductor notification definition) Menu**

1. Details of (Conductor notification definition) Menu > (List) Sub menu are listed below

Table 4.1- 1「List」 Sub menu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input Required** | **Input type** | **Restrictions** |
| Notification name | Enter notification name | ○ | Manual input |  |
| Class(CURLOPT\_URL) | Enter class URL | ○ | Manual input |  |
| Header (CURLOPT\_HTTPHEADER) | Enter HTTP header in JSON format | ○ | Manual input |  |
| Message (CURLOPT\_POSTFIELDS) | Enter message according to service specification. | ○ | Manual input | ※ |
| PROXY / URL (CURLOPT\_PROXY) | Enter URL if PROXY setting is needed. |  | Manual input |  |
| PROXY / PORT (CURLOPT\_PROXYPORT) | Enter PORT if PROXY setting is needed. |  | Manual input |  |
| Confirmation URL(FQDN) | Enter FQDN used in input variable for confirmation URL |  | Manual input |  |
| Other | Enter in JSON format All option corresponding、curl\_setopt() are available.  Refer to cURL function for PHP |  | Manual input |  |
| Start time | Enter to stop notification |  | Manual input |  |
| End time | Enter to stop notification |  | Manual input |  |
| Remarks | Free description field |  |  |  |

※ For available ITA variables, see the following table.

Refer to 「5.1.1Conductor」 for Teams, Slack input.

Table 4.1- 2 Conductor class list ITA variables

|  |  |  |
| --- | --- | --- |
| **ITA variables** | **Selected variable** | **Remarks** |
| \_\_CONDUCTOR\_INSTANCE\_ID\_\_ | Conductor instance ID |  |
| \_\_CONDUCTOR\_NAME\_\_ | Conductor name |  |
| \_\_OPERATION\_ID\_\_ | Operation ID |  |
| \_\_OPERATION\_NAME\_\_ | Operation name |  |
| \_\_STATUS\_ID\_\_ | Status ID |  |
| \_\_STATUS\_NAME\_\_ | Status name |  |
| \_\_EXECUTION\_USER\_\_ | Execution user |  |
| \_\_TIME\_BOOK\_\_ | Book time |  |
| \_\_TIME\_START\_\_ | Start time |  |
| \_\_TIME\_END\_\_ | End time |  |
| \_\_JUMP\_URL\_\_ | Jump URL | ※ |

※”Operation confirmation URL(FQDN)” input is used as output for work confirmation URL as following.

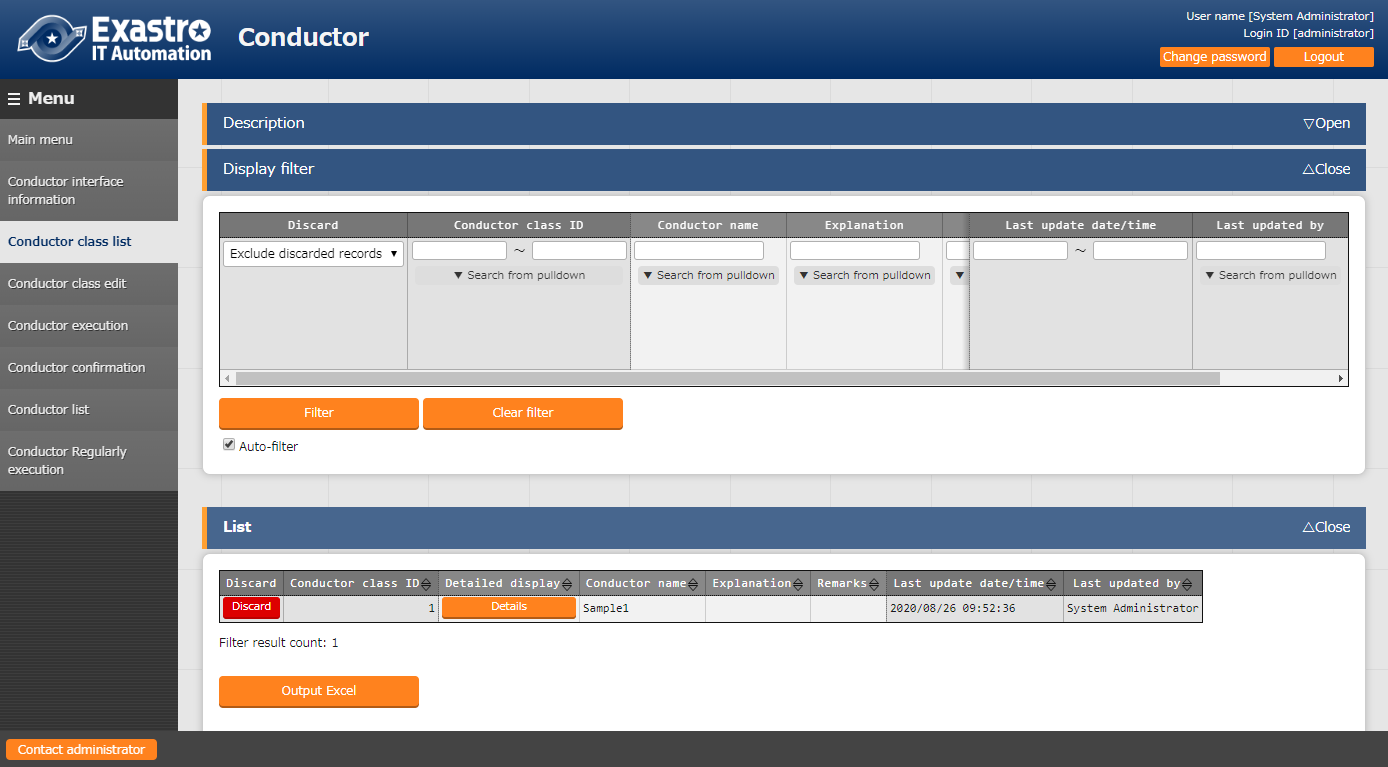
<Work confirmation URL(FQDN)>/default/menu/01\_browse.php?no=2100180005&conductor\_instance\_id=X

|  |
| --- |
| Ex: When work confirmation URL(FQDN) value is 「http://exastro-it-automation.local」  http://exastro-it-automation.local/default/menu/01\_browse.php?no=2100180005&conductor\_instance\_id=X |

Conductor class list

1. In the [Conductor class list] screen, users can view or discard registered Conductor classes.

Click the “Details” button to move the edit screen “4.1.4 Conductor class edit”.



admin@nec.amc.jp

~

~

~

~

:set bin noeol

**Figure 4.1‑3 (Conductor class list) Menu**

Conductor class edit

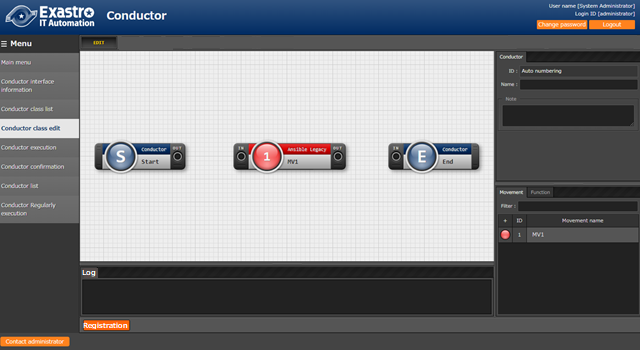
1. “Conductor class edit”

* Users can register Conductor names and parts that makes up the work flow (also called Nodes)
* This screen has two different modes. Please see the table below for more information.

**Table 4.1-4** **Conductor class edit screen mode list**

|  |  |
| --- | --- |
| **Mode** | **Description** |
| EDIT | ・The mode that users can edit Conductor class  ・Default mode of Conductor class edit screen  ・Switch to VIEW mode by clicking register/update button in EDIT mode |
| VIEW | ・The mode that users can only view Conductor class  ・The mode that is displayed when clicking the “Details” button in Conductor class list  ・Switch to EDIT mode by clicking the edit button in VIEW mode |

For more information about operating the different modes, please refer to **"Table 4.1-18 "Conductor class edit"** Menu Operation list.

*  The contents displayed in the "Detailed Information "section changes depending on the selected Node

**Drag and drop**

Node

**B)　Detailed information**

1. **Node list**

**Detailed**

**Information**

**Node list**

* **Figure 4.1‑4 Submenu screen (Conductor class edit：EDIT)**
* Node list  
  The Nodes available are displayed in the bottom right area of the screen  
  The Nodes can be created from the following tabs.

* 1. Movement tab
     + List of ID and name of registered Movement.
  2. Function tab
     + Conductor end
     + Conductor pause
     + Conductor call
     + Symphony call
     + Conditional branch
     + Parallel branch
     + Parallel merge
     + Status File branch

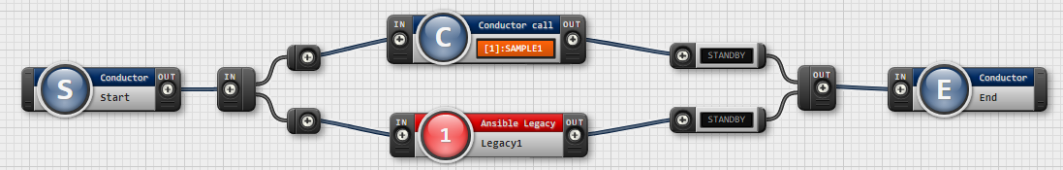
1. The following table shows the different conductors and their function.

**Table 4.1-5　Node list**

|  |  |  |
| --- | --- | --- |
| **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 are complete. |
|  | Conductor pause | Pauses the workflow temporarily.  Cancel the pause to move on to next step. |
|  | Conductor call | Calls another registered Conductor class and executes it.  ※When original Conductor shuts down with error, Conductor executes process as normal call and does not have effect on original status. |
|  | 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.  ・Normal end  ・Abnormal end  ・Emergency stop  ・Preparation error  ・Unexpected error  ・SKIP complete  ・Warning end |
|  | 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. |
|  | Status file branch | Branches off process based on results of work directory of Movement file. |
|  | Movement | Execute Movement |

The following section lists the different restrictions for using Nodes.

IN/OUT of all Nodes have to be connected.



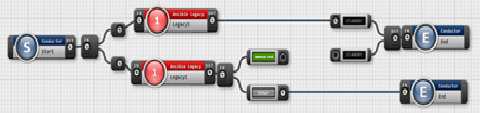
**Figure 4.1‑5 Node restriction (Correct example：Parallel branch)**

If you want to use a Parallel merge node, you must also use a Parallel branch



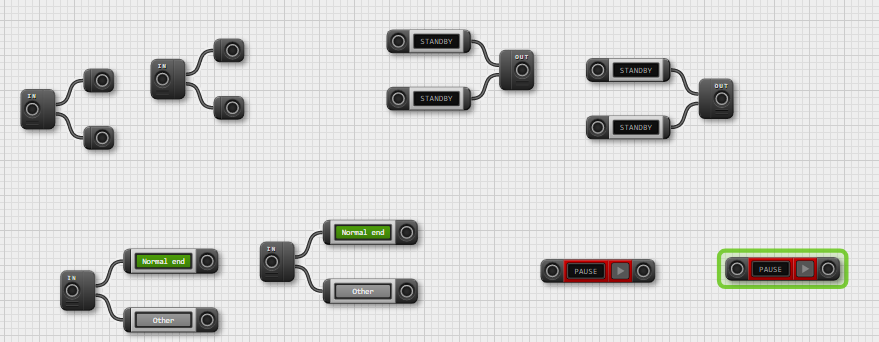
**Figure 4.1‑6 Node restriction (Bad example：Parallel branch)**

・Flow that is branched by Conditional branch can’t be merged to Parallel merge.



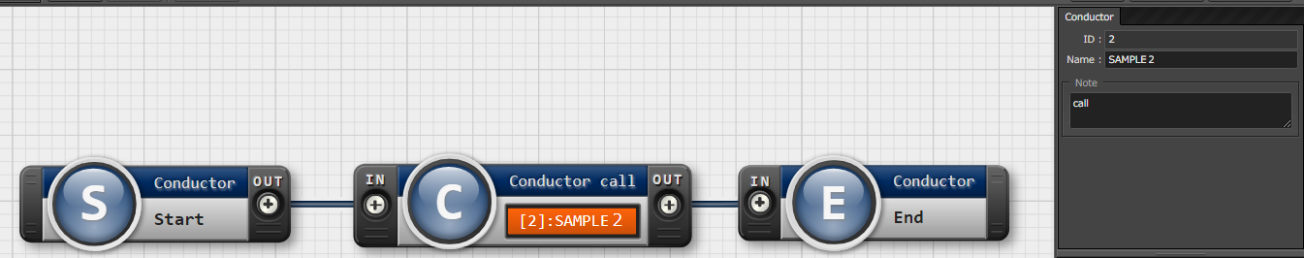
**Figure 4.1‑7 Node restriction （Bad example：Conditional branch）**

・For Parallel branch, Conditional branch, Parallel merge, and Conductor pause, it is invalid to connect them to same type of Node.



**Figure 4.1‑8 Node Restriction (Bad example： Successive use）**

・It is invalid to assign the Conductor that is currently begin updated to Conductor Call.



**Figure 4.1‑9 Node restriction (Bad example：Conductor call)**

・User can set Nodes by drag and drop the Nodes on the bottom-right side of screen

・Users can memo the description of operation or comment in the Note column of each node.

・The column is only for reference on the web, it doesn’t affect operation execution.

・Click the “Register” button after setting up Nodes to register the Conductor class.

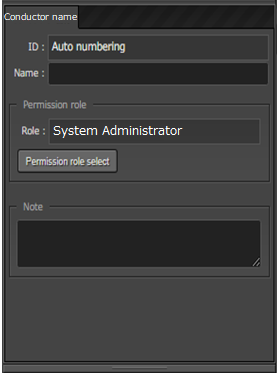
Detailed Information  
・In the upper left area of the screen, users can see detailed information about the selected node.  
・The name of the tab changes depending on the node selected.

1. Conductor name tab  
   ・This tab is displayed when no node is selected  
   ・The items found in the tab are as following.

**Table 4.1-6　 “Conductor name” Tab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input Required** | **Input type** | **Restriction** |
| ID | Unique ID for Conductor is auto-numbered | - | Auto input | - |
| Name | Enter any desired name for Conductor class | 〇 | Manual input | - |
| Notice | Select notification to perform  Select multiple notification for each status. | - | Select | ※ |
| Role | Select the role that have access to this Conductor.  If no role is selected, all role will be have access. |  | Select |  |
| Note | Enter description and comment for Conductor class | - | Manual  input | - |

※selectable notifications are registered in “**エラー! 参照元が見つかりません。** Conductor notification definition”



**Figure 4.1‑10 “Conductor” name tab**

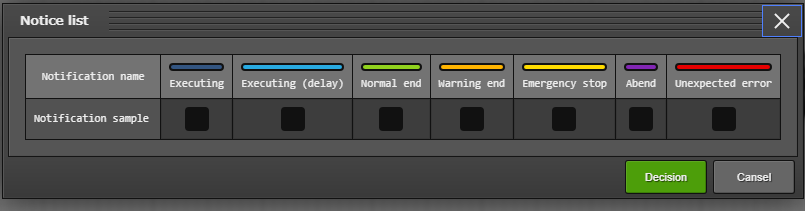
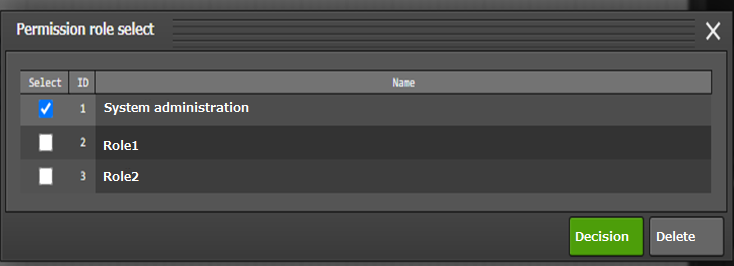


Figure 4.1- 1 “Notice” popup

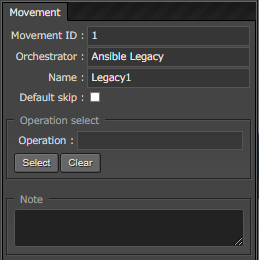


**Figure 4.1‑12 “Permission role select” popup**

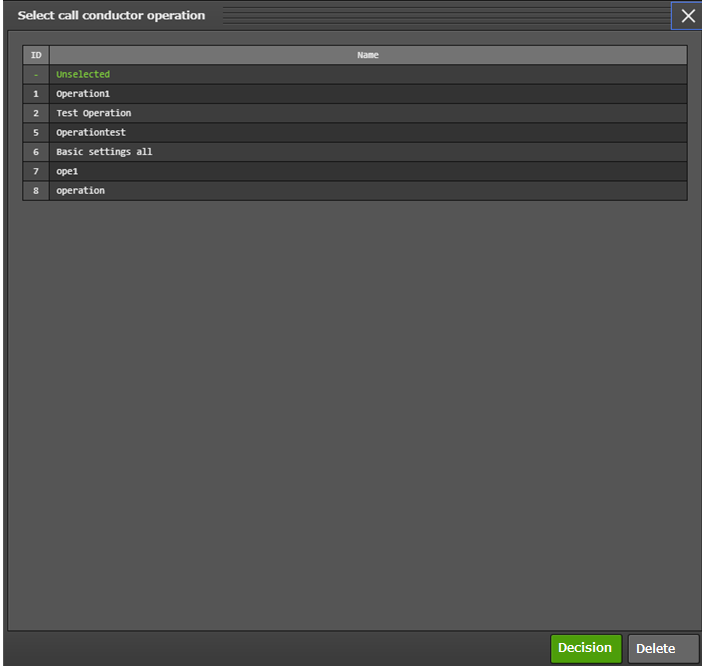
1. Movement details and input items  
   ・This tab is displayed if a Node is selected in the Node list Movement tab.  
   ・The items found in the tab are as following.

**Table 4.1-7　“Movement” tab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| Movement ID | ID of the selected Movement is displayed. | - | Auto input | - |
| Orchestrator | Orchestrator name of the selected Movement is displayed. | - | Auto input |  |
| Name | Name of the selected Movement is displayed. | - | Auto input | - |
| Default skip | Target operation will be skipped if checked.  This is a parameter that can be changed in Conductor execute screen. | - | Manual input |  |
| Operation | ・Click the Select button to select Operation from the displayed list.  ・The name of the Operation class will be displayed. | - | Select | - |
| Note | Enter a comment or a description regarding the Node. |  | Manual input |  |



**Figure 4.1‑13 “Movement” tab**

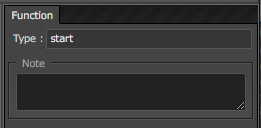


**Figure 4.1‑14 “Operation select” popup**

1. Node (common) details and input items  
   ・This tab is displayed if “Conductor start”, ”Conductor end” or “Conductor pause” is selected in the Node list’s Function tab.  
   ・The items found in the tab are as following.

**Table 4.1-8　 “Function” tab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| Type | Type of selected Node is displayed | - | Auto input | - |
| Note | Enter a comment or a description regarding the Node. | - | Manual input | - |

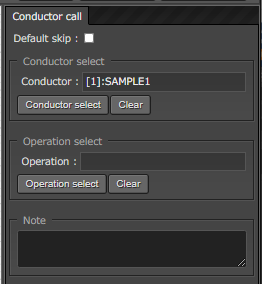


**Figure 4.1‑15 “Function” tab**

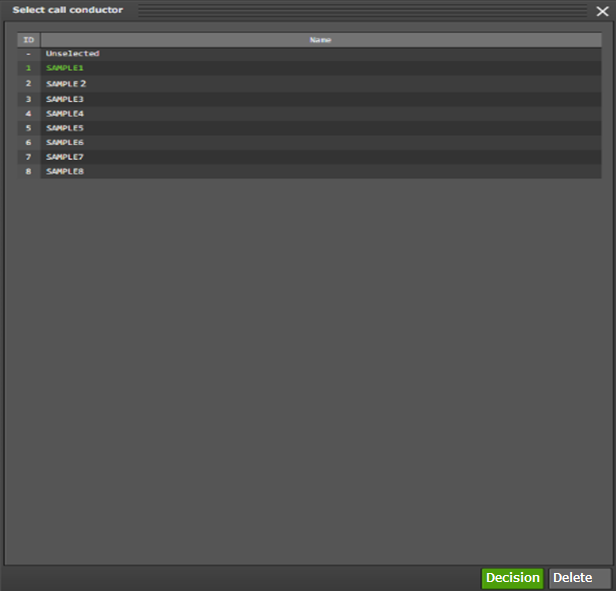
1. Conductor call details and input items  
   ・This tab is displayed if “Conductor call” is selected in the Node list’s Function tab.  
   ・The items found in the tab are as following.

**Table 4.1-9　 “Conductor call” tab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| Default skip | Target operation will be skipped if checked.  This is a parameter that can be changed in Conductor execution screen. | - | Manual | - |
| Conductor | ・Click the Select button to select Conductor class from the displayed list.  ・The name of the Conductor class will be displayed. | 〇 | Select | - |
| Operation | ・Click the Select button to select Operation from the displayed list.  ・The name of the Operation class will be displayed. | - | Select | - |
| Note | Enter a comment or a description regarding the Node. |  | Manual input |  |



**Figure 4.1‑16 “Conductor call” tab**



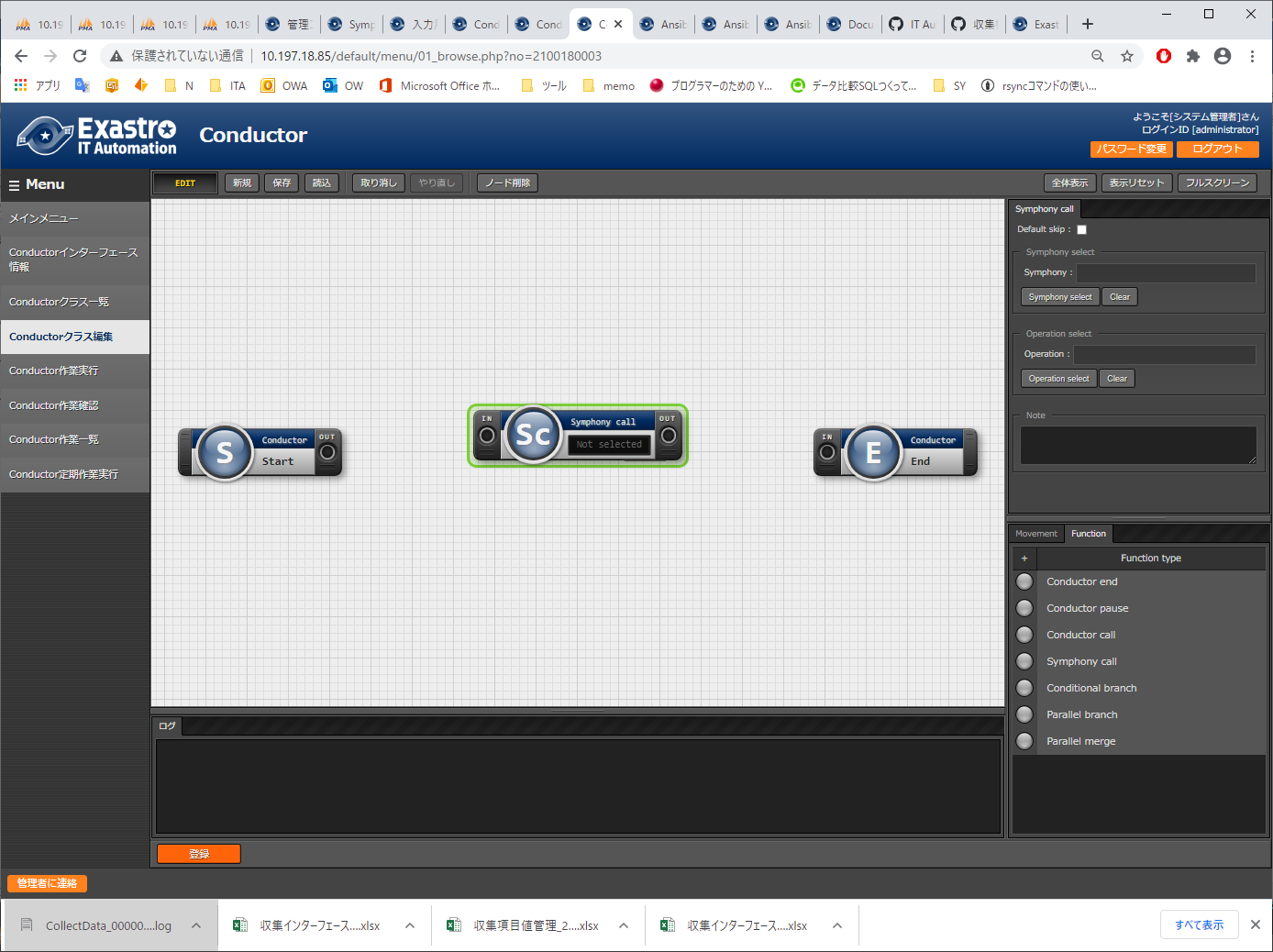
**Figure 4.1‑17 “Conductor select” popup**

1. “Symphony call” tab.

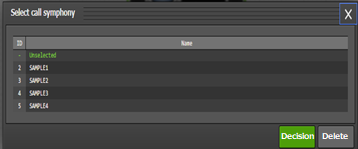
・This tab is displayed if “Symphony call” is selected in the Node list’s Function tab.  
・The items found in the tab are as following.

**Table 4.1-10 “Symphony call” tab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| Default skip | Target operation will be skipped if checked.  This is a parameter that can be changed in Conductor execution screen. | - | Manual input | - |
| Symphony | ・Click the Select button to select Symphony class from the displayed list.  ・The name of the Symphony class will be displayed. | 〇 | Select | - |
| Operation | ・Click the Select button to select Operation from the displayed list.  ・The name of the Operation class will be displayed. | - | Select | - |
| Note | Enter a comment or a description regarding the Node. |  | Manual input |  |



**4.1‑18 “Symphony call” tab**

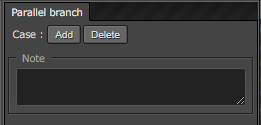


**4.1‑19 “Symphony select” Popup**

1. Parallel branch tab  
   ・This tab is displayed if “Parallel branch” is selected in the Node list’s Function tab.  
   ・The items found in the tab are as following.

**Table 4.1-11　 “Parallel branch” tab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| case | Set number of branches.  2 branches is set on default, click the following to add or delete branch.  ・Add  ・Delete | - | Select |  |
| Note | Enter a comment or a description regarding the Node. |  | Manual input |  |

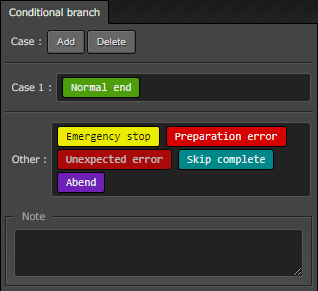


**Figure 4.1‑20** **“Parallel branch” tab**

1. Conditional branch tab  
   ・This tab is displayed if “Conditional branch” is selected in the Node list’s Function tab.  
   ・The items found in the tab are as following.

**Table 4.1-12　 Conductor class edit item list (Conditional branch)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| case  (1-6) | Set conditional branch according to the execution result of Movement and Conductor Call.  User can change the condition by drag and drop  The following is set by default   |  |  | | --- | --- | | **Case1** | Normal end | | **Other** | Abnormal end、Emergency stop, Preparation error, Unexpected error, Skip complete, Warning end | | - | Select | ※ |
| Note | Enter a comment or a description regarding the Node. |  | Manual input |  |

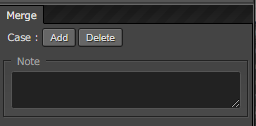


**Figure 4.1‑21** **“Conditional branch” tab**

1. Parallel merge tab  
   ・This tab is displayed if “Parallel merge” is selected in the Node list’s Function tab.  
   ・The items found in the tab are as following.

**Table 4.1-13　 “Merge” tab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| case | Select the number of parallel operation.  2 branches is set on default, click the following to add or delete branch.  ・Add  ・Delete | - | Select |  |
| Note | Enter a comment or a description regarding the Node. |  | Manual input |  |



**Figure 4.1‑22 “Merge” tab**

1. “End ”tab

・This tab is displayed if “End” is selected in the Node list’s Function tab.

・Item listed in tab are following

Table 4.1- 3　 “End” tab

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| End status | When End is processed, selected status is reflected on Conductor.  - Normal end (Default value)  - Warning end  - Error end  Priority of status when multiple End node is executed Priority：　　Normal < Warning < Error | - | Select |  |
| Note | Enter a comment or a description regarding the Node. | - | Manual input | - |

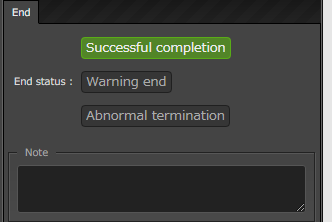


Figure 4.1- 2 “End” tab

1. “Status file branch” tab

・This tab is displayed if “End” is selected in the Node list’s Function tab.

・Item listed in tab are following

Table 4.1- 4　“Status file branch” tab

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| If  /  elseif | Set branch requirement for Movement status file.  Press “Add”/”Delete” button to add or delete branch.  Default branch is “if” and “else” | - | Manual input | ※ |
| Note | Enter a comment or a description regarding the Node. | - | Manual input | - |

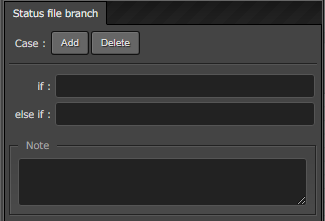


Figure 4.1- 3 “Status file branch” tab

※Reference status file

* Refer to “MOVEMENT\_STATUS\_FILE” under operation result directory in each Movement for status file
* “else” process is operated when status file does not exist.
* When status file has multiple line (including line feed), line after feed is excluded from evaluation subject.

Ex） Status file including line feed

|  |
| --- |
| 1  23  4 |

Evaluate “1” as status file content.

Table 4.1- 5　Status file ITA variable

|  |  |  |
| --- | --- | --- |
| **ITA variable** | **Variable content** | **Restriction** |
| \_\_movement\_status\_filepath\_\_ | Under operation result directory ”MOVEMENT\_STATUS\_FILE” path | ※ |

※compatible with “Ansible-Legacy”, ”Ansible-Pioneer” and ”Ansible-LegacyRole”

1. “Node” tab

* This will be displayed if there are multiple nodes selected in the Movement/Function tab in the Node list.
* You can either drag and drop nodes into the selection area, or click multiple nodes while holding the Shift key to select multiple Nodes.
* The items found in the tab are listed below.

Table 4.1- 6 “Node” tab

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input format** | **Restrictions** |
|  | Aligns the selected nodes to the left | - |  | - |
|  | Aligns the selected nodes to the left and right center | - | Select | - |
|  | Aligns the selected nodes to the right | - | Select | - |
|  | Aligns the selected nodes to the top | - | Select | - |
|  | Aligns the selected nodes to the top and bottom center | - | Select | - |
|  | Aligns the selected nodes to the bottom | - | Select | - |
|  | Aligns the selected nodes vertically with equally spacing in-between them. | - | Select | - |
|  | Aligns the selected nodes horizontally with equally spacing in-between them. | - | Select | - |

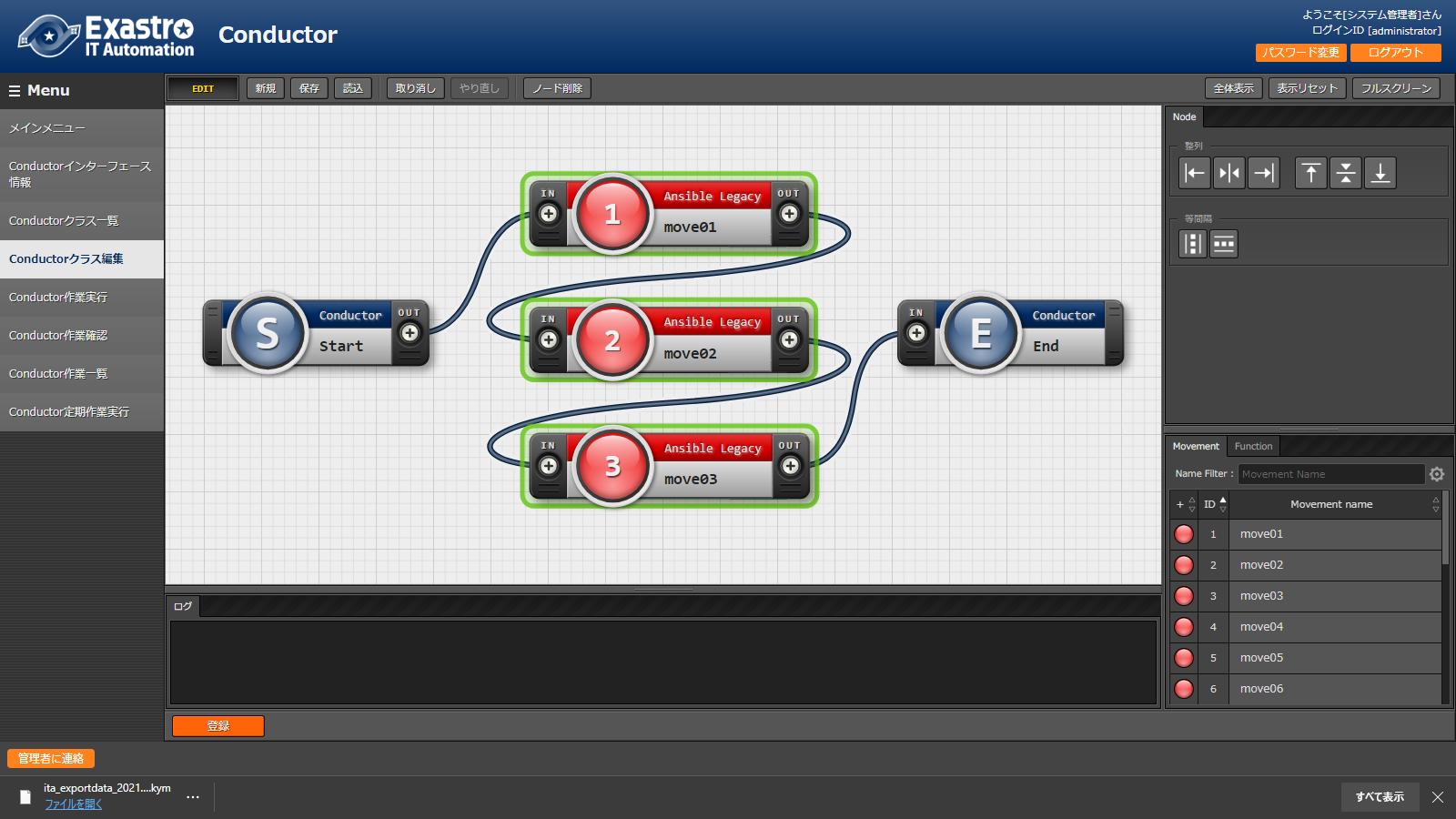


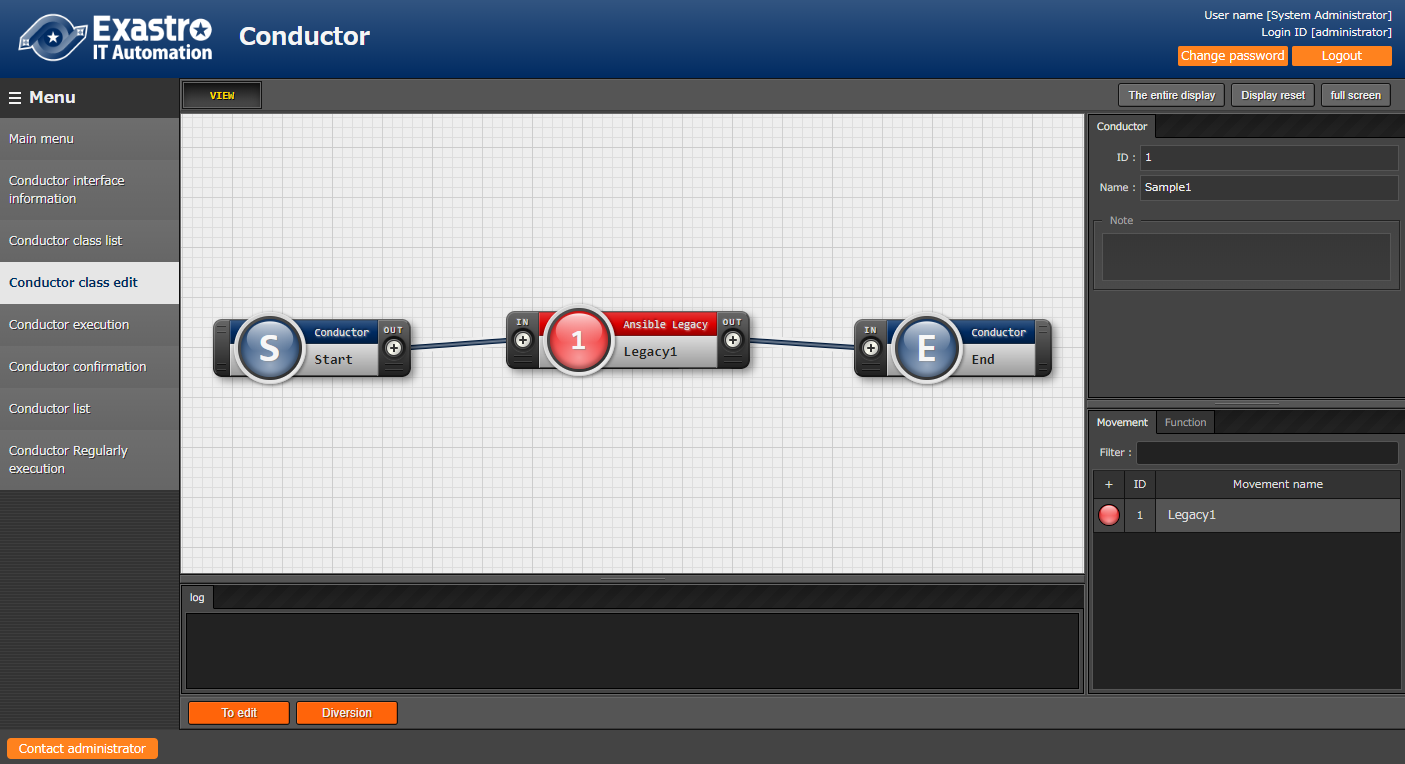
Figure 4.1- 4 “Node” tab

* Operations that can be executed in Class edit screen is as follows.

**Figure 4.1-12　List of operations that can be performed in Conductor class edit screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item** | **Description** | **Register**  **(EDIT）** | **Update**  **(VIEW)** | **Update**  **(EDIT）** | **Remarks** |
| New | Return to the default status. | 〇 | - | - |  |
| Save | Save the current edit screen as file. | 〇 | - | - |  |
| Read | Read and restore status from saved file. | 〇 | - | - |  |
| Cancel | Cancel the previous operation. | 〇 | - | 〇 |  |
| Redo | Redo the cancelled operation. | 〇 | - | 〇 |  |
| Delete node | Delete the selected node. | 〇 | - | 〇 |  |
| Registration | Perform registration | 〇 | - | - |  |
| To Edit | Switch to EDIT mode to perform edit of Constructor class. | - | 〇 | 〇 |  |
| Diversion | Diverse registered Conductor and register a new conductor. | - | 〇 | 〇 |  |
| Update | Update the edited content. | - | - | 〇 |  |
| Reload | Discard the modification and return to the status before edit. | - | - | 〇 |  |
| Cancel | Discard the modification and switch to VIEW mode |  | - | 〇 |  |

(2) View mode.  
When moving from [Conductor class list] screen to Conductor class edit screen or after registration, the following screen will be displayed



**Figure 4.1‑21 “Conductor class edit” menu (“View” mode)**

Table 4.1- 7　 “View” mode

|  |  |
| --- | --- |
| **Item** | **Description** |
| “To edit” button | Press this button to edit a registered Conductor |
| “Diversion” button | Press this button to copy a registered Conductor. |

admin@nec.amc.jp

~

~

~

~

:set bin noeol

1. The following screen will be displayed if “To edit” button is clicked.

admin@nec.amc.jp

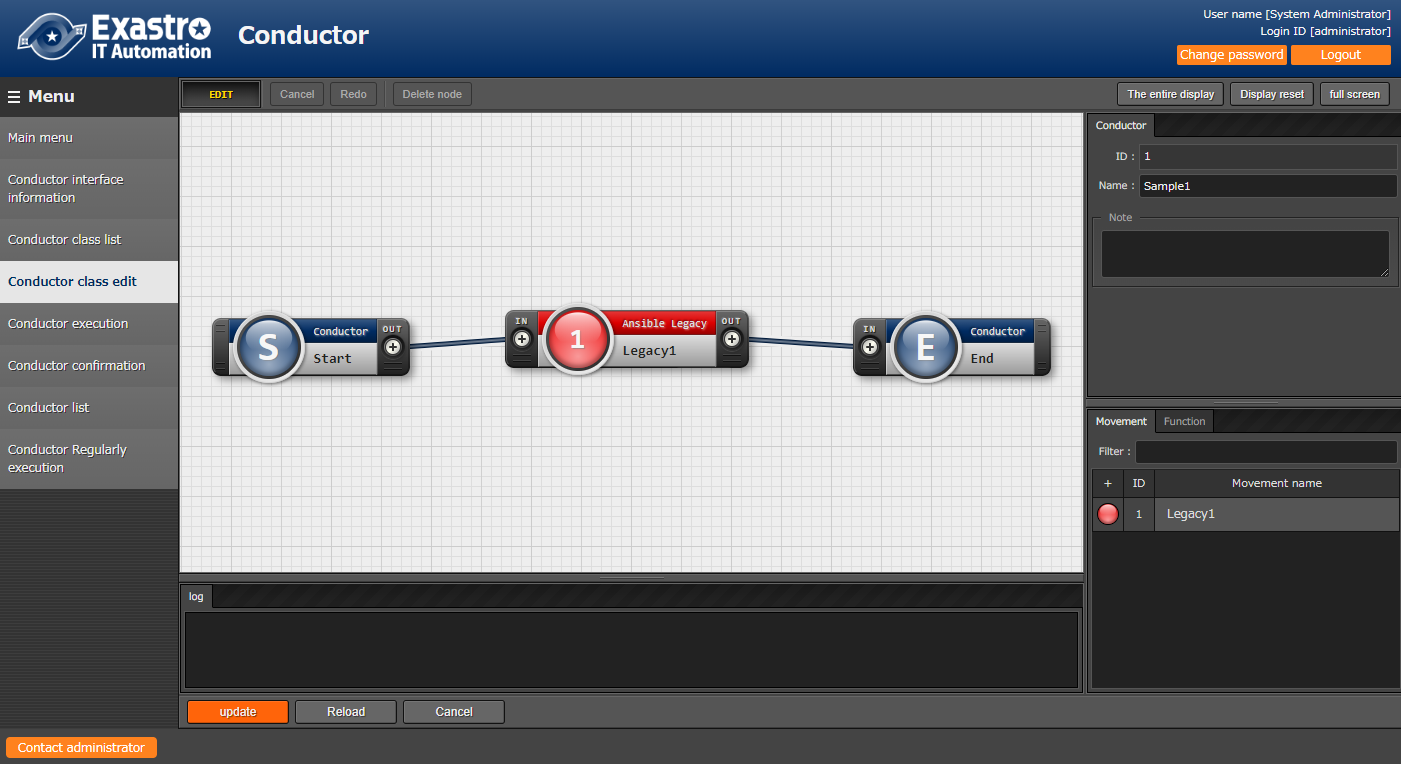
~

~

~

~

:set bin noeol



**Figure 4.1‑22 “Conductor class edit” menu (“Edit” mode)**

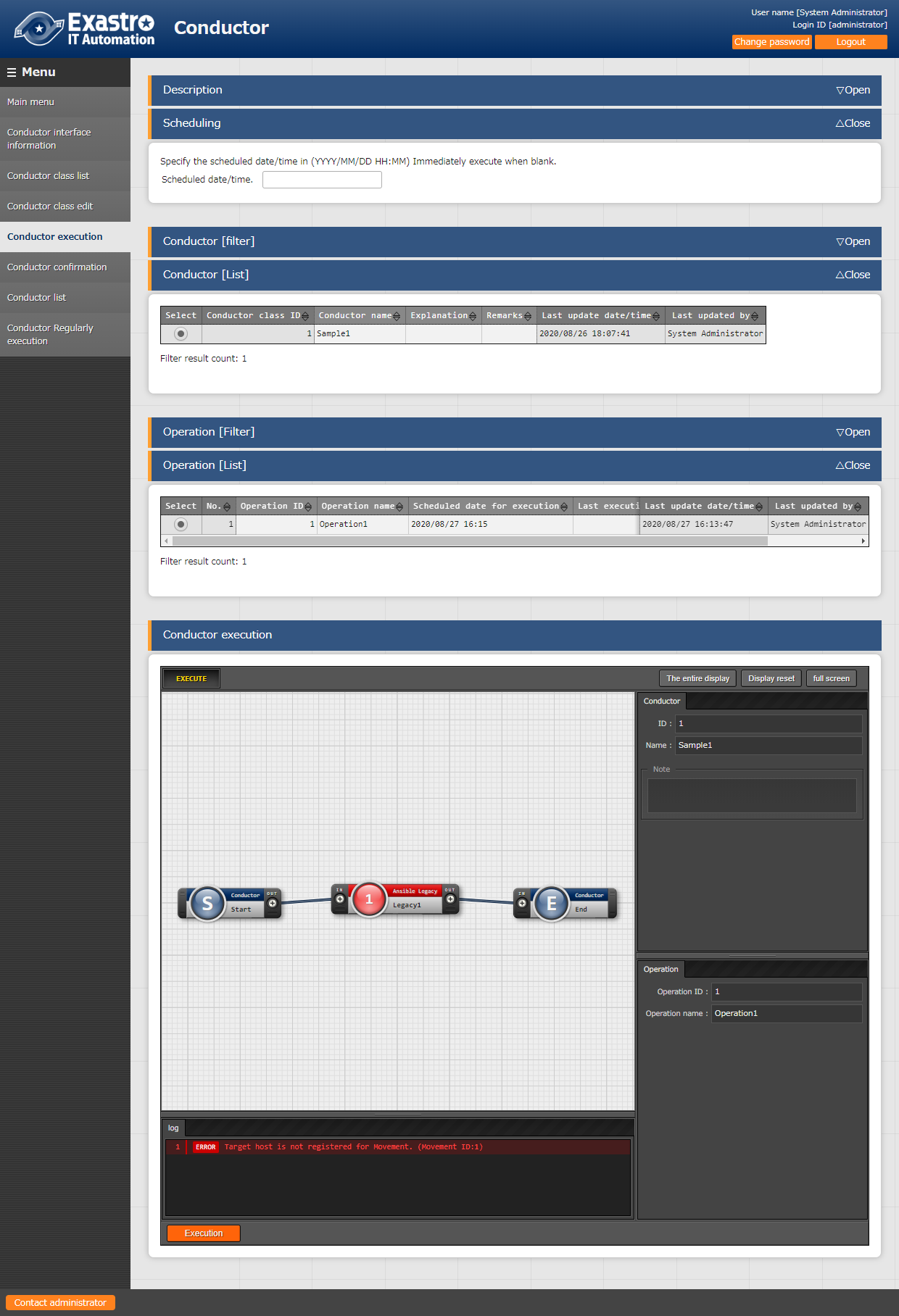
Table 4.1- 8　 “Edit” Mode

|  |  |
| --- | --- |
| **Item** | **Description** |
| The entire display | All nodes will be displayed |
| “Display reset” button | The display will reset according to the “Conductor\_start” |
| “Full screen” button | Makes the browser window go full screen.  ※Press the “end full screen mode” to exit full screen. |
| “Update” button | Saves the edited contents |
| “Reload” button | Resets the edit screen and removes any changes. |
| “Cancel” button | Cancels the process and returns the screen before the “Edit” button was pressed. |

### Conductor execution

1. Indicate Conductor execution in [Conductor execution] screen.

* “Conductor [List]” displays the Conductors registered in “4.1.3 Conductor class list”.
* “Operation [List]” displays the Operations registered in “Basic console.
  1. Please refer to “User Instruction Manual” for details.
* Select radio button in “Conductor [List]” and “Operation [List]”, then click the “Execution” button to move to “4.1.6 Conductor confirmation” then start tracing of execution.
* Enter “Scheduled date/time” then click the “Execution” button will schedule execution. The scheduled execution can be checked in “4.1.7 Conductor list”.  
  ※Date/Time before current time can’t be entered.
* The setting value of Operation and skip for Movement and Conductor Call can be changed.
  1. Setting value will not reflect to registered data. The setting value will only reflect to Conductor executions.



**Figure 4.1‑28 Submenu screen (Conductor execution)**

The list of items in Conductor execution screen is as follows.

**Table 4.1-21　Registration screen items (Conductor execution)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input**  **Required** | **Input type** | **Restrictions** |
| Scheduled date/time | Specify the scheduled date and time of Conductor execution | - | Manual input | Date and time before the current time cannot be entered |
| Conductor [List] | The Conductor registered in “4.1.7 Conductor class list” will be displayed. | 〇 | Radio buttons |  |
| Operation [List] | The operations registered in “Basic console” will be displayed | 〇 | Radio buttons |  |
| Skip | Check to skip the target operation  ※Refer to the “About skip” in below | - | Checkbox |  |
| Operation | ※Refer to the “About specifying Operation” in below | - | Manual input |  |
| Notice | Check notification setting | - | Button |  |
| Execution | Execute register Conductor | 〇 | Button |  |

* About specifying Operation.

Click the “Select” button in “Operation Select” column will display a modal of Operation list.

Users can specify Operation that is different from the Operation specified by radio button.

According to the specification, Conductor can be executed with the “Specific value” substituted with the value registered for other Operation ID in the “Substitution value list” menu of the Orchestrator which that Movement belongs to (e.g. “Substitution value list” in ITA Anisble-Legacy console).

The Operation ID specified in Conductor class edit screen is saved according to register/update.

Moreover, users can change the Operation for each step of Conductor before execution.

However, the settings in Conductor execution screen only reflects to Conductor execution. The settings will not be saved.

Users can take use of this function to diverse the Movement to operate for another server.

* About Skip

Users can change the status of Skip.

The skip setting in Conductor class edit screen is saved according to register/update.

Moreover, users can change the skip setting for each step of Conductor before execution.

The settings will not be saved.

Users can take use of this function to temporary skip operation or execute operation while executing Conductor.

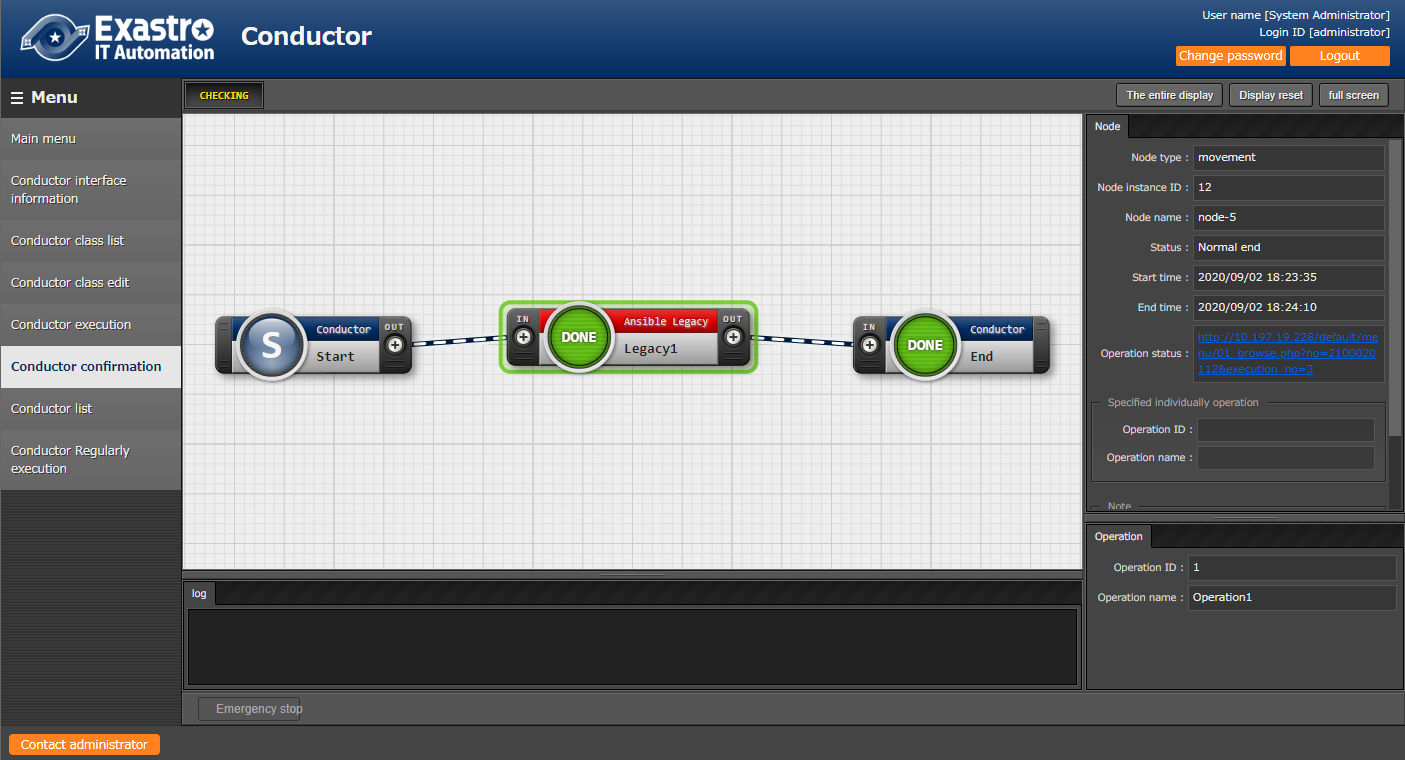
Conductor confirmation

1. In [Conductor confirmation] screen, the status of Conductor execution is displayed.

By clicking the “Details” button in “4.1.7 Conductor list”, the status of the selected Conductor will be displayed. Users can execute “Cancel reservation”, “Resume” or “Emergency stop” according to the situation.

The execution status of each Node can be displayed by selecting them.

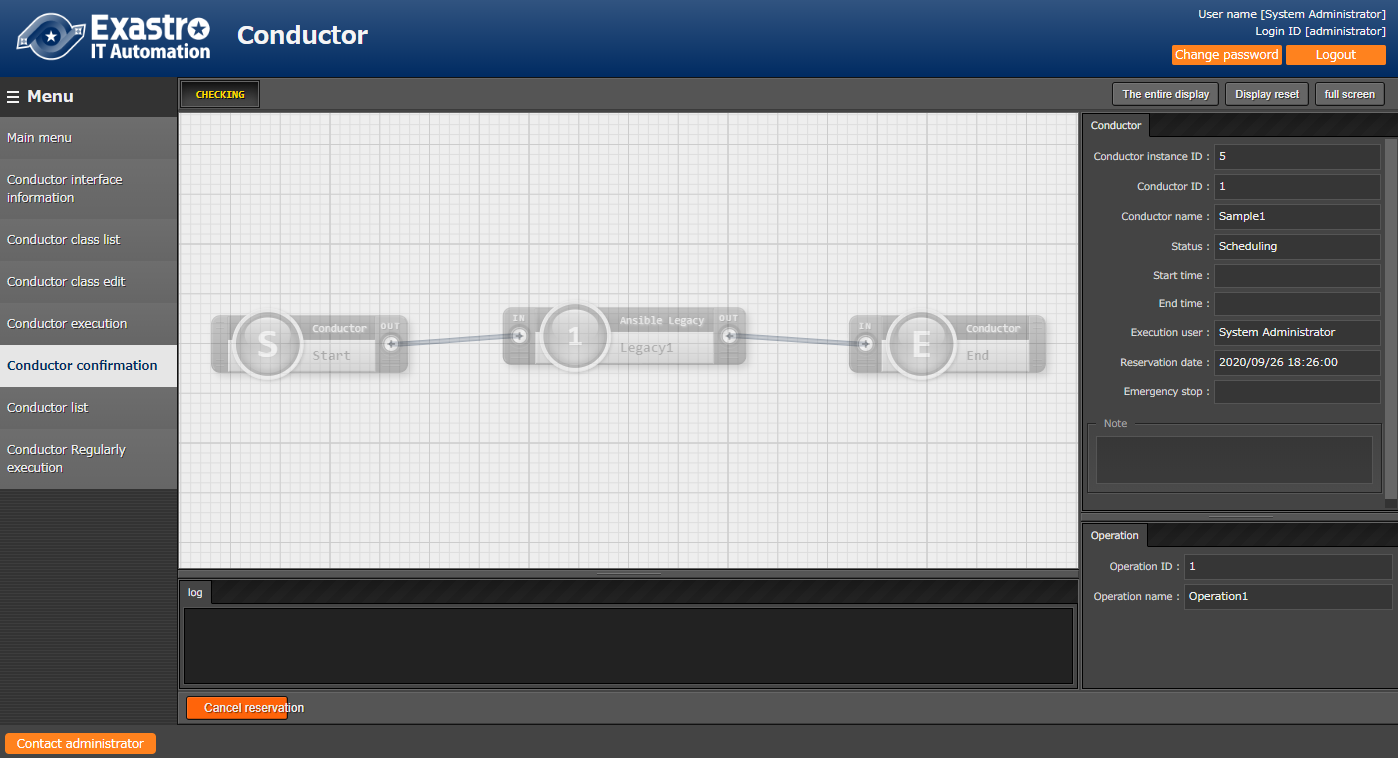
To check the details of the execution status, uses can select the URL in “Operation status” of “Movement” and “Conductor Call”.



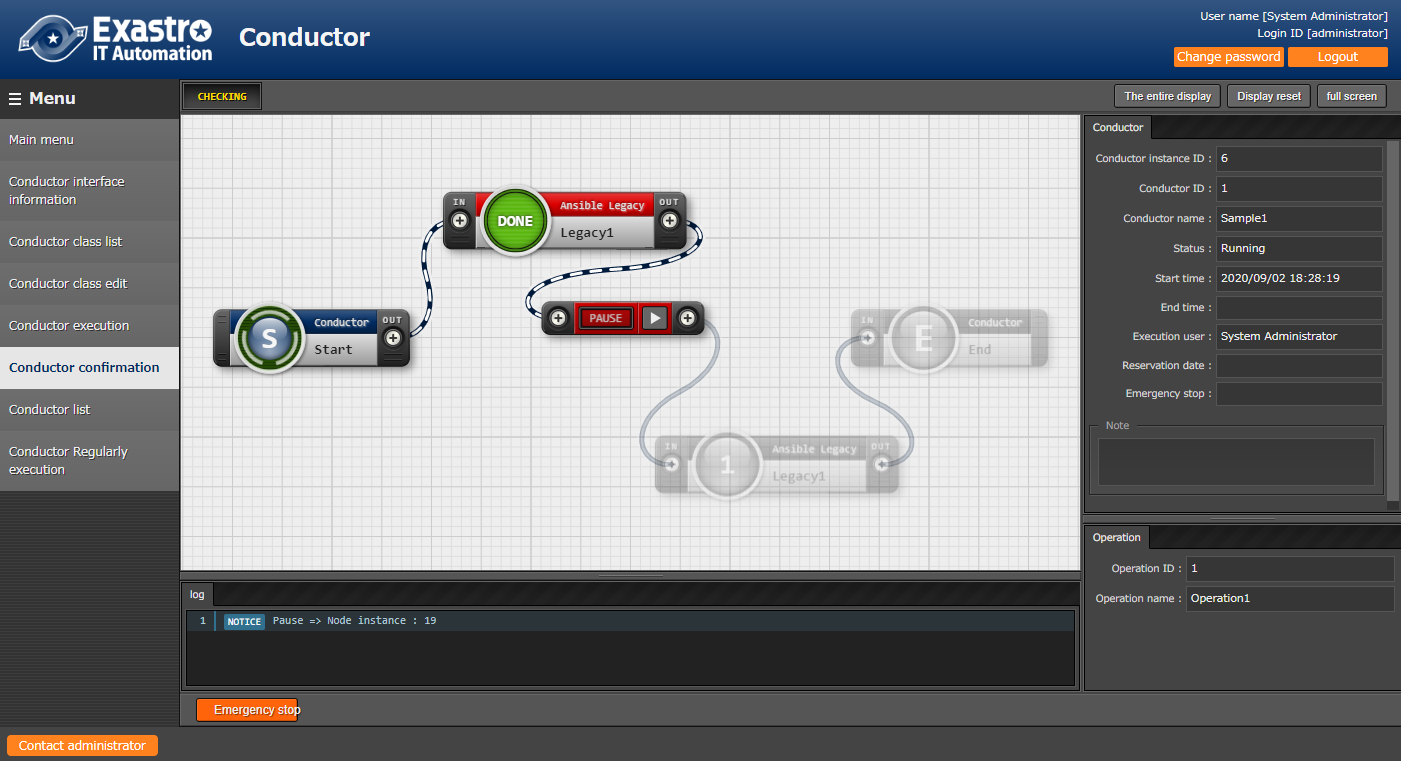
**Figure 4.1‑29 Submenu screen (Conductor confirmation)**

※If you edit the Conductor that has been executed in “Conductor execution” with "Conductor class edit", it will be in a different state from the Conductor during execution, so even if you click the "Details" button, the status may not be displayed. If you want to edit the Conductor that has already been executed and then execute again, it is recommended to create another Conductor with a new diversion by using "Conductor class edit" and use it.

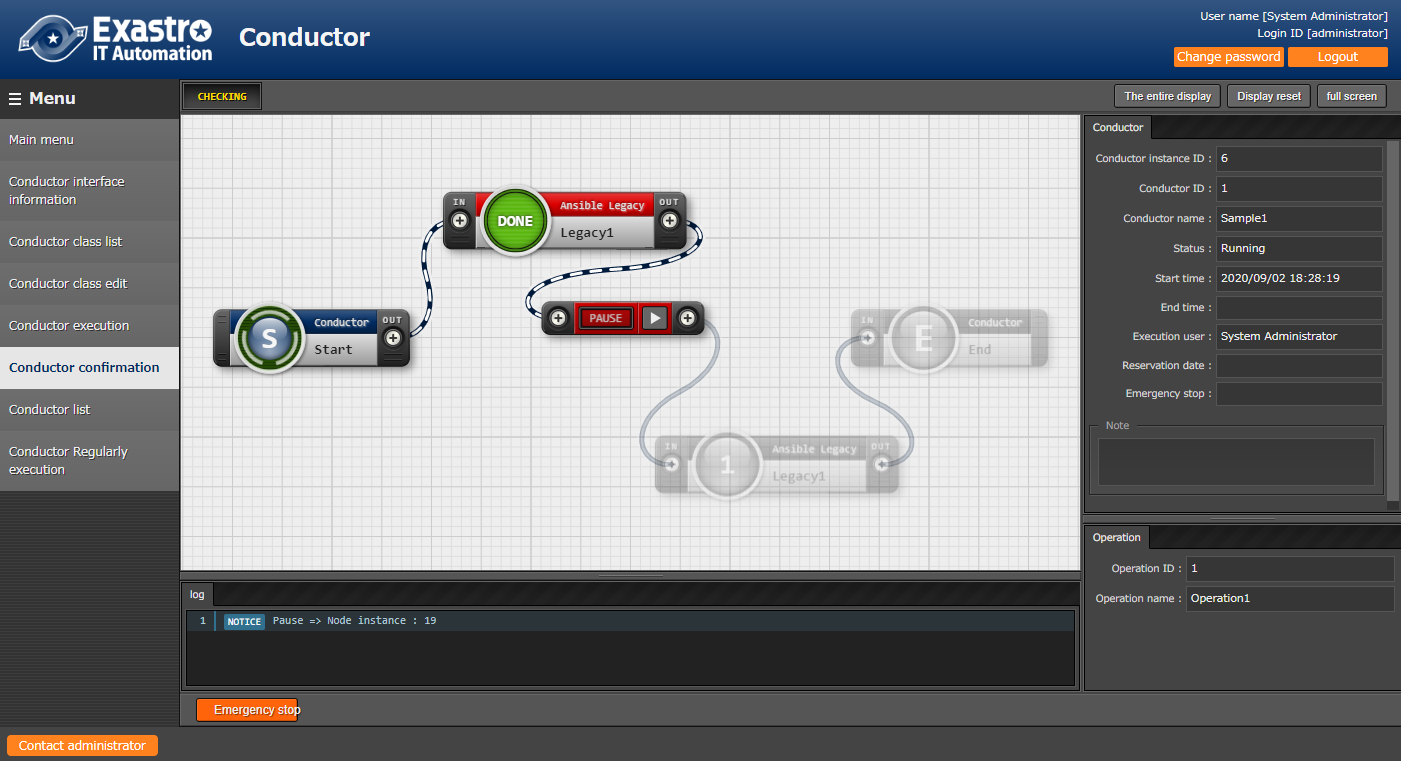
・If the selected Conductor execution is scheduled and is yet executed, a “Cancel reservation” button will be displayed.   
・If the button is clicked, the status in “4.1.7 Conductor list” will become “Unexecuted (Schedule)” and will not be executed.



**Figure 4.1‑30 Submenu screen (Conductor confirmation – Cancel reservation）**

****

**Figure 4.1‑31 Submenu screen (Conductor confirmation – Resume)**

****

**Figure 4.1‑32 Submenu screen (Conductor confirmation – Emergency stop)**

The list of items in Conductor confirmation screen is as follows.

**Table 4.1-22　Registration screen list (Conductor confirmation)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input type** | **Restriction** |
| Resume | Cancel pause and continue operation execution | - | button | - |
| Emergency stop | Stop Conductor execution | - | button | - |
| Cancel reservation | Cancel scheduled Conductor execution | - | button | Displayed only when execution is scheduled and is yet executed. |

1. The "Conductor confirmation" menu displays the execution status of all executed Conductors.

* In the upper right corner, users can see the information about any selected Nodes.

The tab name changes depending on the selected Node.

1. "Conductor name" tab.

* This tab is displayed if no Nodes are selected.
* The tab items are as follows.
* Table 4.1- 23 “Conductor name” tab

|  |  |
| --- | --- |
| **Item** | **Description** |
| Conductor instance ID | Conductor instance ID A unique ID which is automatically assigned for each Conductor Instance. |
| Conductor name | Conductor name Displays the names of running Conductor classes. |
| Status | Status Displays the status for running Conductors.  One of the following statuses will be displayed.  ・Not executed  ・Not executed (Reserved)  ・Executing  ・Executing(postponed)  ・Normal end  ・Emergency stop  ・Abnormal end  ・Unexpected error  ・Reservation deleted |
| Pause Status | Pause status  Displays the “Pause status” for any running Conductors that are paused. This item will display the “Pause status” for any conductors that are called using the “Conductor call” function. This item will be blank if the conductor is unpaused. |
| Start time | Start time Displays the date and time of when the Conductor was executed. |
| End time | End time Displays the data and time of when the Conductor ended. |
| Execution user | Execution user Displays the user who executed the Conductor. |
| Reservation date | Reservation date Displays the conductor’s reservation date and time. |
| Emergency stop | Emergency stop flag  Displays the status “Stopped” if the Conductor has been emergency stopped. Will display “Not stopped” if else. |
| Note | Remarks/Notes Displays any description and notes for the Conductor. |

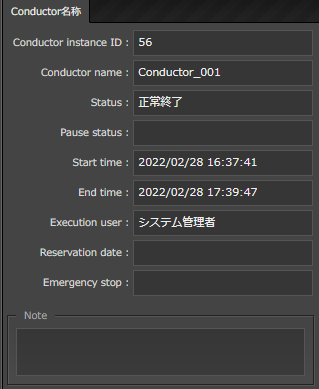
****

Figure 4.1- 5 “Conductor name” tab

1. “Node” tab

* This tab is displayed when a Node is selected.
* The tab items are as follows.
* Table 4.1- 24 “Node name” tab

|  |  |  |
| --- | --- | --- |
| **Item** | | **Description** |
| Node type | | Displays the Type of the Node |
| Node Instance ID | | A unique ID which is automatically assigned for each Node Instance. |
| Node name | | Displays the name of the Node class. |
| Status | | Displays the status of running Nodes. One of the following statuses will be displayed.  ・Not executed  ・Preparing  ・Executing  ・Executing(Postponed)  ・Executed  ・Abnormal end  ・Emergency stop  ・Paused  ・Normal end  ・Preparation error  ・Unexpected error  ・Skip complete  ・Post-skip pause  ・Skip complete  ・Warning |
| Status file | | Displays the status file value if the selected node is a Movement. |
| Start time | | Displays the date and time the node was executed. |
| End time | | Displays the data and time the node ended. |
| Operation status | | Displays a link that leads the user to the operation confirmation screen of the conductor, symphony or movement. |
| Specified individually operation | Operation ID | ID of the specified individual operation. |
| Operation Name | Name of the specified individual operation. |
| Note | | Displays any description and notes for the Node |

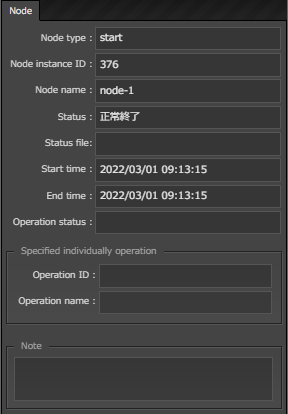
****

Figure 4.1- 34 “Node” name

### Conductor list

1. Users can manage executed Conductor operations in “Conductor list” screen.

By specifying the criteria and clicking the “Filter” button, the table of Conductor list will be displayed.

Users can click the “Details” button to move to “4.1.6 Conductor confirmation” screen.

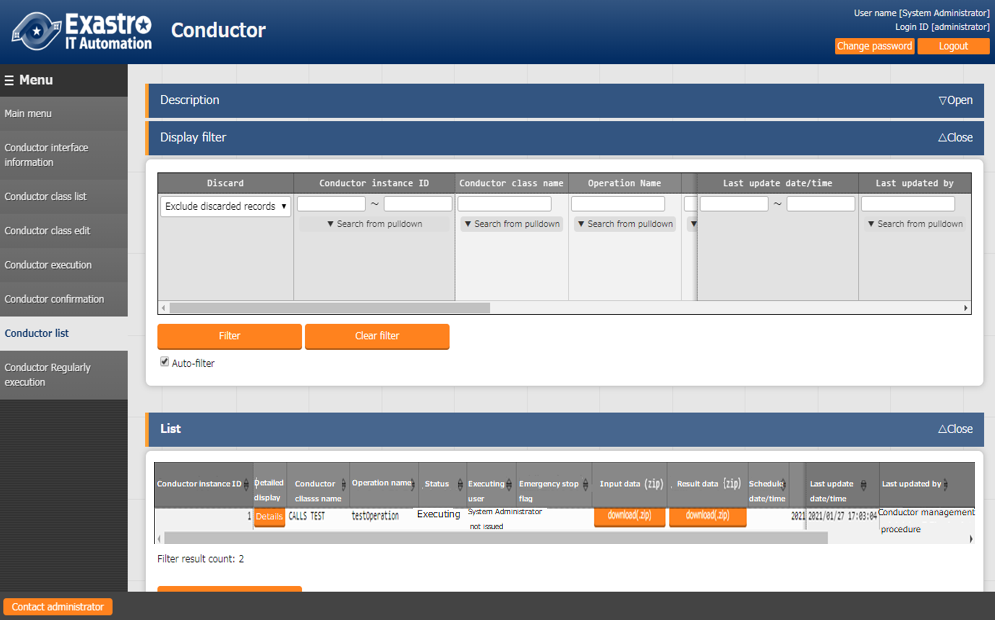
Conductor

Click "Download (.zip)" under "Input data (zip)" to download all Movements executed under Conductor and its data files.

Click "Download (.zip)" under "Result data (zip)" to download all execution logs, error logs and such of all of the Movements executed under Conductor.

Notification log can be downloaded from notification log.   
Refer to “5.1.2 Notification log output example” for samples.

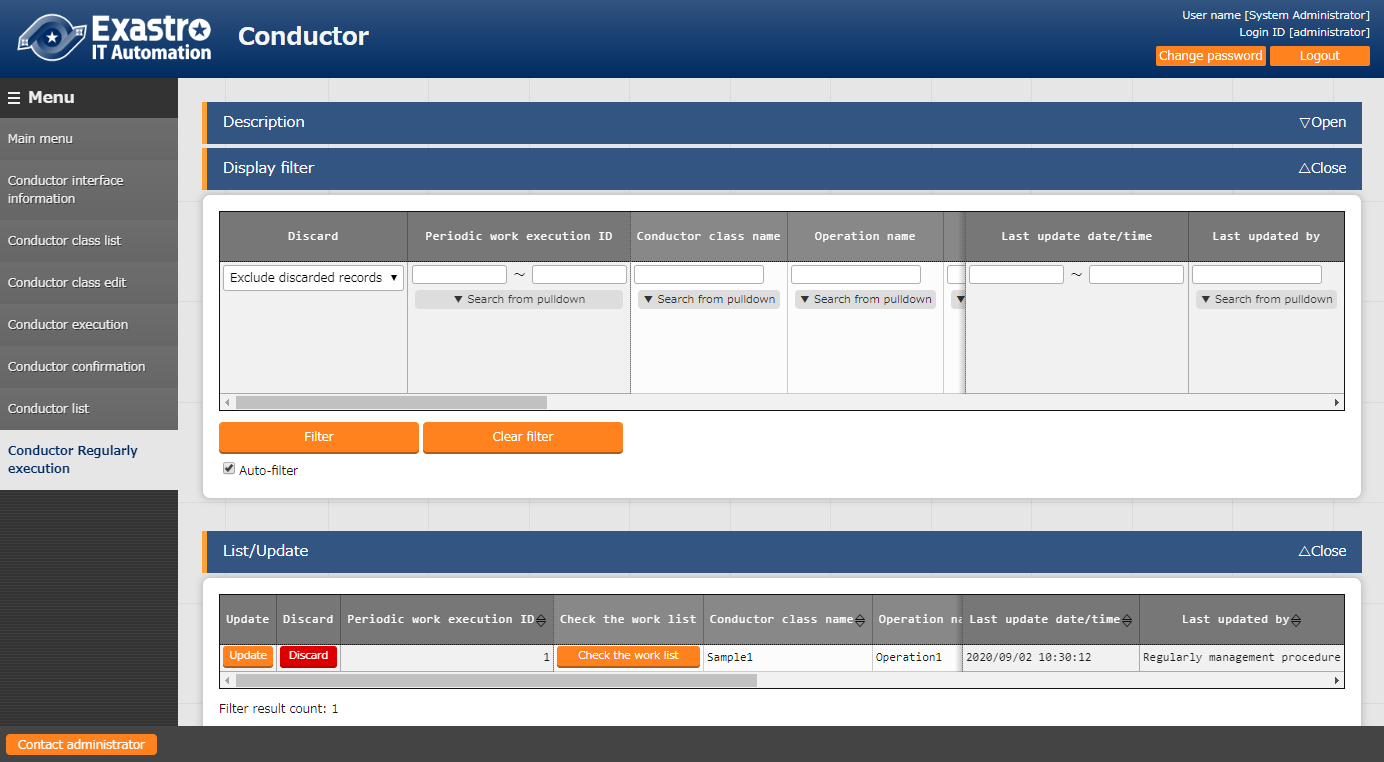
If the Conductor has a hierarchical structure, the movement at the end will also be targeted.



**Figure 4.1‑33 Submenu screen (Conductor list)**

### Conductor regularly execution

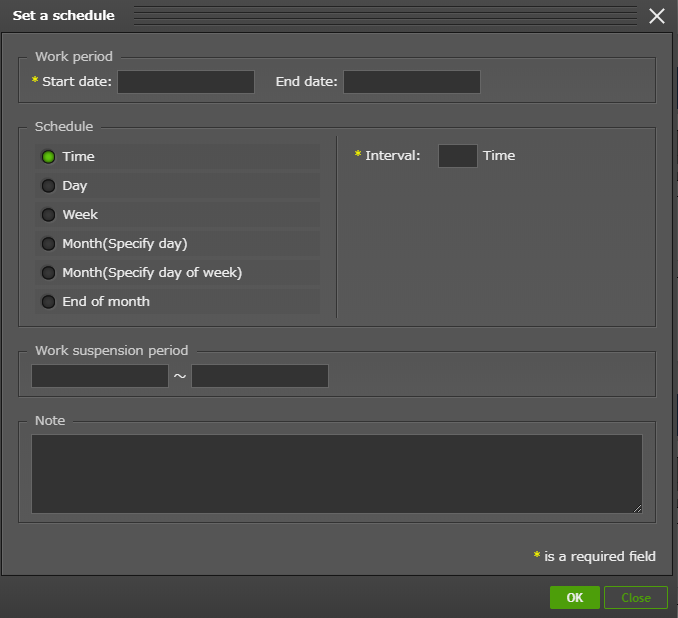
1. Users can manage regular execution of Conductor operation in [Conductor regularly execution] screen. Click the “Check the work list” in “List” will move to “4.1.7 Conductor list” screen with the target Conductor executed by regular execution. Click the Conductor name list to move to “4.1.4 Conductor class edit”



**Figure 4.1‑34 Submenu screen (Conductor regularly execution)**

1. Click “Register” - “Start Registration” button to set regular execution.

Schedule can only be set in the setting window by clicking “Schedule settings” button.



**Figure 4.1‑35 Schedule settings screen (Regularly execution)**

1. The list of items in Conductor confirmation screen is as follows.

**Table 4.1-23　Register screen item list (Regularly execution)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item** | | **Description** | **Input**  **required** | **Input type** | **Restrictions** |
| Conductor class name | | Conductor registered in “4.1.3 Conductor ” are displayed. | 〇 | List selection | - |
| Operation name | | Operation registered in “Basic Console – Input operation list” | 〇 | List selection | - |
| Status | | Refer to the following “Table 4.2-11 Status list (Regularly execution)” | - | Automatic  input | - |
| Execution User | | User executed “Register” ”renew” will be registered as operation user.  Regulatory work operation is registered to “4.1.7Conductor list” the “Operation user” is also passed on.  When operation user cannot operate selected　”Conductor name”　(Ex. User does not have access authority for the Movement)) status becomes “tie up error”. | - | Automatic  input |  |
| Schedule setting | | A button that opens a modal window to set details of schedule. | - | - | - |
| Schedule | Next execution date | Based on the registered schedule, the execution date will be updated automatically. | - | Automatic  input | - |
| Start date | Enter the start date of regular work execution.  "Next execution date" is always updated with the date after "Start date". | 〇 | Manual  input | Enter by Schedule setting only |
| End date | Enter the end date of regular work execution.  The status will become “completed” if “Next execution date” passed “End date”. | - | Manual  input | Enter by Schedule setting only |
| Period | Select the period of regular execution.  “Time”, “Day”, “Week”, “Month (Specify day)”, “Month (Specify day of week)”, ”End of month” can be selected. | 〇 | Radio button | Enter by Schedule setting only |
| Interval | Select the regular execution interval based on the selected period. | 〇 | Manual  input | Enter by Schedule setting only |
| Week number | Used when period is “Month (Specify day of week)”, select the week number to execute work. | ※1 | List selection | Enter by Schedule setting only |
| Day of week | Used when period is “Week” or “Month (Specify day of week)”, select the day of week to execute work. | ※2 | List selection | Enter by Schedule setting only |
| Day | Used when period is “Month (Specify day)”, select the date to execute work. | ※3 | Manual input | Enter by Schedule setting only |
| Time | Enter the time of regular execution. | ※4 | Manual input | Enter by Schedule setting only |
| Work suspension period | Start | Enter the start date/time of work suspension period.  During the time between start time and end time, registered Symphony will not be executed. | ※5 | Manual input | Enter by Schedule setting only |
| End | Enter the end date/time of work suspension period.  During the time between start time and end time, registered Symphony will not be executed. | ※5 | Manual input | Enter by Schedule setting only |
| Remarks | | Free description field. | - | Manual input | - |

※1 Week number is required when period is “Month (Specify day of week)”.

※2 Day of week is required when period is “Month (Specify day of week)”.

※3 Day is required when period is “Month (Specify day)”.

※4 Time is required when period is “Day”, “Week”, “Month (Specify day)”, “Month (Specify day of week)”, ”End of month”.※5 When setting work suspension period, both “Start” and “End” are required.

**Table 4.1-24　Status list (Regular execution)**

|  |  |
| --- | --- |
| **Status name** | **Description** |
| In preparation | The status immediately after registration.  The status will become “In operation” when backyard updates “Next execution date” automatically. |
| In operation | The status of normal execution.  The system registers operation to “4.1.7 Conductor list” 3 minutes before “Next execution date”, then updates “Next execution date” based on the schedule setting. |
| Completed | The status when “Next execution date” passed “End date”. Further Conductor execution registration will not be performed. |
| Mismatch error | The status when setting value of schedule is not correct. |
| Linking error | The status when registering execution failed in “4.1.7 Conductor list”.  Same as the status “In operation”, system registered execution in “4.1.7 Conductor list”, then updates “Next execution date” based on the schedule setting. If registration of execution failed again, the status will remain “Linking error”. |
| Unexpected error | The status when errors other than “Mismatch error” and “Linking error” happens. |
| Conductor discard | The status when the registered Conductor is discarded. The status will be updated to “In preparation” if the discarded Conductor is restored. |
| Operation discard | The status when the registered Operation is discarded. The status will be updated to “In preparation” if the discarded Operation is restored. |

1. The status will become “In preparation” immediately after registered in “Regular execution” menu. Backyard will update “Next execution date” based on the registered schedule setting, then the status will become “In operation”.

If the status is “In operation” or “Linking error”, the system registers operation to “4.1.7 Conductor list” 3 minutes before “Next execution date”, then updates “Next execution date” based on the schedule setting.

※ When pause is set in the Symphony which is registered in regularly execution, if users don’t “resume” in “4.1.6　Conductor confirmation” after operation is registered, the status in “4.1.7 Conductor list” will remain “Executing”.

1. The "Conductor execution" menu allows users to execute Conductors.

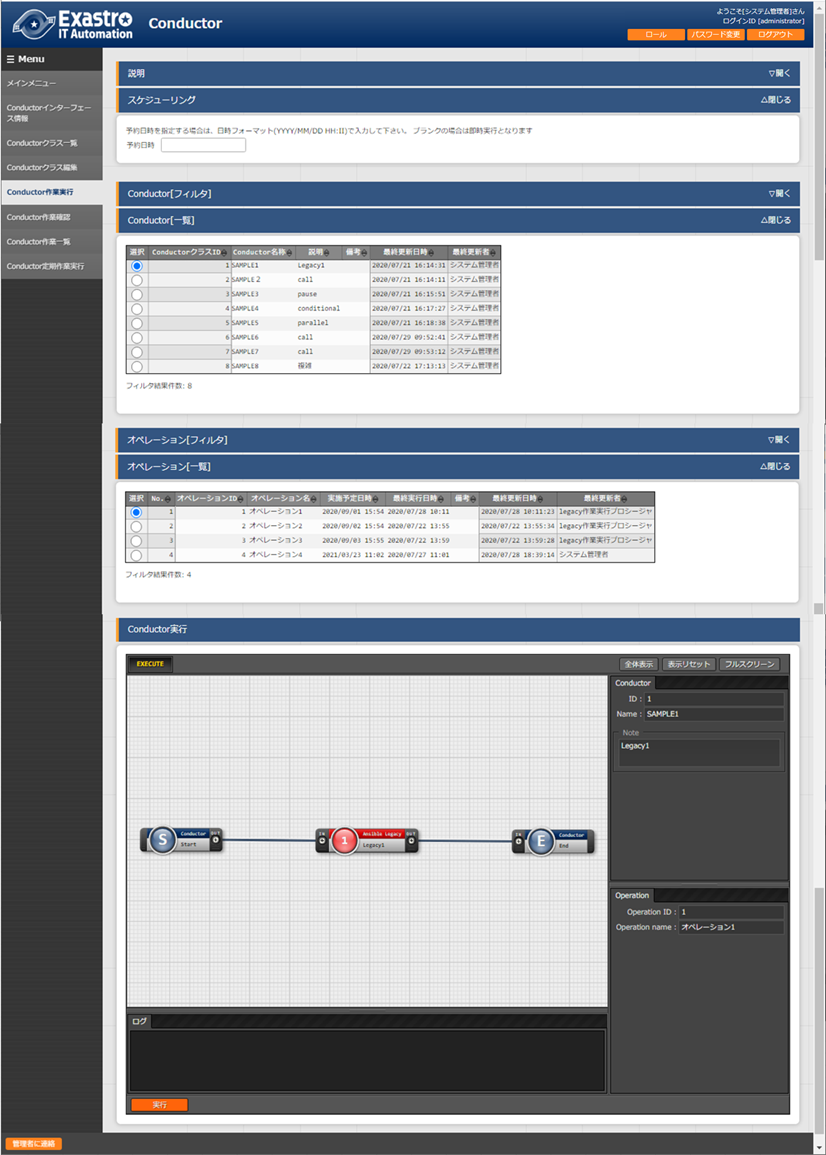
* The "Conductor list" submenu displays the Conductors registered in”4.1.3 Conductor Class list”
* The "Operation list" submenu displays the Operations registered in the "Basic console" > "Operation list" menu.
* For more information, please see the "[Exastro-ITA\_User\_Instruction\_Manual\_Basic\_Console](%22Exastro-ITA_User_Instruction_Manual_Basic_Console%22.https:/github.com/exastro-suite/it-automation-docs/blob/v1.10.0/asset/Documents/Exastro-ITA_User_Instruction_Manual_Basic_Console.pdf)".
* Select a Conductor and Operation from the "Conductor/Operation [List]" submenus using the radio buttons and press the "Execute" button in order to execute the Conductor.
* Doing so will move the user to "4.1.6 Conductor confirmation" screen where they can trace the operation.
* Users can also schedule when they want the Conductor to be executed by inputing their desired data in the "Schedule" submenu. The registered information can be seen in "4.1.7 Conductor list".
* Note that it is not possible to input a date/time that has already been passed.
* It is only possible to change the setting values for Movements, Conductor Calls, Symphony Call operations and Skips.
* ※Any changes done in the Conductor Execute menu will not be reflected to the registered data in the Conductor Edit menu
* The access permission's common roles set to the selected Conductors and Operations are carried over to the executed Conductor.
* If there are no common roles, the operation cannot be executed.
* 
* Figure 4.1- 28 "Conductor Execution" menu
* The "Conductor execution" menu's common items can be found below

Table 4.1- 21　"Conductor common" items list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Input required** | **Input method** | **Restrictions** |
| Reservation time/date | Specify the date/time of which the Conductor will be executed. | - | Manual input | Not possible to enter a date/time earlier than the current date/time |
| Conductor[List] | Displays the conductors registered in “4.1.7Conductorclass list”. | ○ | Radio button |  |
| Operation[List] | Displays the operations registered in “4.1.4 Input operation list”. | ○ | Radio button |  |
| Skip | Allows users to skip desired operations  ※See “Specifying Operations” below | - | Check box |  |
| Operation | ※See “Specifying operations” below | - | Manual input |  |
| Notice | Allows users to confirm their notification settings | - | Button |  |
| Execute | Executes the registered Conductor | ○ | Button |  |

* + Specifying Operations

Press the "Select" button in the "Operation" row to display a list of Operations.

Users can then specify an operation different to the already specified Operation's Operation ID.

This allows the user to assign and execute "specific values" registered for Operation IDs in the "Assignment Value Management" menu of the orchestrator to which the Movement belongs (e.g., in the "Substitute Value list" menu in the ITA Ansible-Legacy console).

If the user has individually specified an Operation ID, the user can press the Conductor register/Update button in the Conductor class edit screen to save the configuration.

Users can also individually specify the operation in the Conductor execution screen before actually executing the Conductor. Users can change the Operation ID even if it has been edited and saved in the Conductor class edit menu.

Note that specifying the operation Id in the Conductor execution screen will only affect the current execution, meaning that the configuration (post changing Operation ID) will not be saved.

We recommend that you use this function when you want to use the same Movement, but for different servers.

* + Skip

Users can skip certain nodes by checking the "Skip" box.

If the user has changed the conductor to skip some of the nodes in the Conductor Class edit screen, the user can press the "Register" or "Update" button to save the configuration.

Users can also change which nodes to skip in the Conductor execution screen before actually executing the Conductor.

Note that specifying which nodes the users want to skip in the Conductor execution screen will only affect the current execution, meaning that the configuration will not be saved.

We recommend that you use this function if you want to temporarily jump over some of the operations (nodes)

* Access permissions for the executing operations.
  + Access permissions for Executing Operations

If the user executes a Conductor/Symphony that contains Movements displayed in the "Conductor execution" menu or all operations called using the "Conductor Call" and "Symphony Call" functions without having permission to them, a validation error will occur.

# Appendix

## Conductor notification destination definition

### Conductor notification destination definition setting

■Teams setting example

|  |  |
| --- | --- |
| Notification name | Notification sample |
| Notification destination (CURLOPT\_URL) | Enter Teams Webhook |
| Header (CURLOPT\_HTTPHEADER) | [ "Content-Type: application/json" ] |
| Message (CURLOPT\_POSTFIELDS) | {"text": "Notification name：\_\_NOTICE\_NAME\_\_, <br> Conductor name: \_\_CONDUCTOR\_NAME\_\_, <br> Conductor instance ID:\_\_CONDUCTOR\_INSTANCE\_ID\_\_, <br> Operation ID: \_\_OPERATION\_ID\_\_, <br>Operation name:\_\_OPERATION\_NAME\_\_, <br>Status ID: \_\_STATUS\_ID\_\_, <br>Status: \_\_STATUS\_NAME\_\_, <br>Execution user: \_\_EXECUTION\_USER\_\_, <br> Book time: \_\_TIME\_BOOK\_\_, <br>Start time: \_\_TIME\_START\_\_, <br>End time: \_\_TIME\_END\_\_, <br>Emergency shutdown flag: \_\_ABORT\_FLAG\_\_, <br> Operation URL: \_\_JUMP\_URL\_\_, <br> "} |
| PROXY / URL (CURLOPT\_PROXY) | http://proxy.co.jp |
| PROXY / PORT (CURLOPT\_PROXYPORT) | 8080 |
| Work confirmation URL(FQDN) | http://exastro-it-automation.local |
| Other |  |
| Start date |  |
| End date |  |

■Teams notification display example



■Slack setting exmaple

|  |  |
| --- | --- |
| Notification name | Notification sample |
| Notification destination(CURLOPT\_URL) | Enter Webhook URL for Slack |
| Hearder (CURLOPT\_HTTPHEADER) | [ "Content-Type: application/json" ] |
| Message (CURLOPT\_POSTFIELDS) | {  "username": "ITAConductor operation notification",  "text": "notification name：\_\_NOTICE\_NAME\_\_, \n Conductor name \_\_CONDUCTOR\_NAME\_\_, \n Conductor instance ID:\_\_CONDUCTOR\_INSTANCE\_ID\_\_, \n Operation ID: \_\_OPERATION\_ID\_\_, \n Operation name:\_OPERATION\_NAME\_\_, \n Status ID: \_\_STATUS\_ID\_\_, \n Status: \_\_STATUS\_NAME\_\_, \n Execution user: \_\_EXECUTION\_USER\_\_, \n Book time: \_\_TIME\_BOOK\_\_, \n Start time: \_\_TIME\_START\_\_, \n End time: \_\_TIME\_END\_\_, \n Emergency shutdown flag: \_\_ABORT\_FLAG\_\_, \n Operation URL: \_\_JUMP\_URL\_\_ "  } |
| PROXY / URL (CURLOPT\_PROXY) | http://proxy.co.jp |
| PROXY / PORT (CURLOPT\_PROXYPORT) | 8080 |
| Work confirmation URL(FQDN) | http://exastro-it-automation.local |
| Other |  |
| Start date |  |
| End date |  |

■Slack notification display example



■Setting sample (Proxy setting、Notification setting、other)

|  |  |
| --- | --- |
| Notification name | Notification sample |
| (CURLOPT\_URL) | https://sample.webhook.xxx.com/yyyyyyyy |
| Header (CURLOPT\_HTTPHEADER) | [ "Content-Type: application/json" ] |
| Message (CURLOPT\_POSTFIELDS) | {"text": "Notification contents"} |
| PROXY / URL (CURLOPT\_PROXY) | http://proxy.co.jp |
| PROXY / PORT (CURLOPT\_PROXYPORT) | 8080 |
| Work confirmation URL(FQDN) | http://exastro-it-automation.local |
| Other | {"CURLOPT\_TIMEOUT":"10"} |
| Start date | 2020/01/01 00:00:00 |
| End date | 2020/01/01 00:00:00 |
| Remarks | Free description field |

### Notification log output example

Notification log configuration

|  |
| --- |
| YYYY-MM-dd HH:ii:ss Notification execution results(<ID:Notification name>,<ID:Status name>)  Array  (  [RETURN\_MSG] =>　 ：Notification execution return value  [OPTION] => Array 　　　　　　　　　　　 ：Notification execution optionン  (  [CURLOPT\_XXXXXXXX] =>  ・・・・・・・・・・・・・・・・  )  [RESSULT] => Array ：Notification execution results  (  [url] => ：Notification URL  [http\_code] => 　：HTTP Status code  ・・・・・・・・・・・・・・・・  )  ) |

Ex) Notification log (Normal)

|  |
| --- |
| 2021-11-05 15:10:22 Notification execution results(2:Notification sample,5:Normal end)  Array  (  [RETURN\_MSG] => 1  [OPTION] => Array  (  [CURLOPT\_CUSTOMREQUEST] => POST  [CURLOPT\_HEADER] =>  [CURLOPT\_SSL\_VERIFYPEER] =>  [CURLOPT\_SSL\_VERIFYHOST] => 0  [CURLOPT\_TIMEOUT] => 5  [CURLOPT\_CONNECTTIMEOUT] => 2  [CURLOPT\_RETURNTRANSFER] => 1  [CURLOPT\_HTTPPROXYTUNNEL] => 1  [CURLOPT\_URL] => https://sample.webhook.xxx.com/yyyyyyyy  [CURLOPT\_HTTPHEADER] => Array  (  [0] => Content-Type: application/json  )  [CURLOPT\_POSTFIELDS] => {"text": "Notification name：Notification sample2, <br> Conductor name: NULL, <br> Conductor instance ID:3, <br> Operation ID: 1, <br>Operation name:OP\_NULL, <br>Status ID: 5, <br>Status: Normal end, <br>Execution user: System administrator, <br>Reservation date: , <br>Start date/time: 2021/11/05 15:10:08, <br>End date/time: 2021/11/05 15:10:18, <br>Emergency stop flag: Not set, <br> Operation URL: http://exastro-it-automation.local/default/menu/01\_browse.php?no=2100180005&conductor\_instance\_id=3, <br> "}  [CURLOPT\_PROXY] => https://sample.proxy.xxx.com  [CURLOPT\_PROXYPORT] => 8080  )  [RESSULT] => Array  (  [url] => https://sample.webhook.xxx.com/yyyyyyyy  [content\_type] => text/plain; charset=utf-8  [http\_code] => 200  [header\_size] => 834  [request\_size] => 1005  [filetime] => -1  [ssl\_verify\_result] => 0  [redirect\_count] => 0  [total\_time] => 1.519411  [namelookup\_time] => 0.083714  [connect\_time] => 0.107712  [pretransfer\_time] => 0.44203  [size\_upload] => 560  [size\_download] => 1  [speed\_download] => 0  [speed\_upload] => 368  [download\_content\_length] => 1  [upload\_content\_length] => 560  [starttransfer\_time] => 1.519364  [redirect\_time] => 0  [redirect\_url] =>  [primary\_ip] => XXX.XXX.XXX.XXX  [certinfo] => Array  (  )  [primary\_port] => 8080  [local\_ip] => XXX.XXX.XXX.XXX  [local\_port] => 39874  )  ) |

Ex) Notification log (Error)

|  |
| --- |
| 2021-11-05 15:10:20 Notification execution results(1:Notification sample. 5:Normal end)  Array  (  [RETURN\_MSG] =>  [OPTION] => Array  (  [CURLOPT\_CUSTOMREQUEST] => POST  [CURLOPT\_HEADER] =>  [CURLOPT\_SSL\_VERIFYPEER] =>  [CURLOPT\_SSL\_VERIFYHOST] => 0  [CURLOPT\_TIMEOUT] => 5  [CURLOPT\_CONNECTTIMEOUT] => 2  [CURLOPT\_RETURNTRANSFER] => 1  [CURLOPT\_HTTPPROXYTUNNEL] => 1  [CURLOPT\_URL] => https://sample.webhook.xxx.com/yyyyyyyy  [CURLOPT\_HTTPHEADER] => Array  (  [0] => Content-Type: application/json  )  [CURLOPT\_POSTFIELDS] => {"text": "Notification name：Notification sample, <br> Conductor name: NULL, <br> Conductor instance ID:3, <br> Operation ID: 1, <br>operation name :OP\_NULL, <br>Status ID: 5, <br>Status : Normal end, <br> Execution user: System administrator, <br>Reservation date/time: , <br>Start date/time: 2021/11/05 15:10:08, <br>End date/time: 2021/11/05 15:10:18, <br>Emergency stop flag: Not set, <br> Operation URL: http://exastro-it-automation.local/default/menu/01\_browse.php?no=2100180005&conductor\_instance\_id=3, <br> "}  [CURLOPT\_PROXY] =>  [CURLOPT\_PROXYPORT] =>  )  [RESSULT] => Array  (  [url] => https://sample.webhook.xxx.com/yyyyyyyy  [content\_type] =>  [http\_code] => 0  [header\_size] => 0  [request\_size] => 0  [filetime] => -1  [ssl\_verify\_result] => 0  [redirect\_count] => 0  [total\_time] => 2.011686  [namelookup\_time] => 0.532318  [connect\_time] => 0  [pretransfer\_time] => 0  [size\_upload] => 0  [size\_download] => 0  [speed\_download] => 0  [speed\_upload] => 0  [download\_content\_length] => -1  [upload\_content\_length] => -1  [starttransfer\_time] => 0  [redirect\_time] => 0  [redirect\_url] =>  [primary\_ip] => XXX.XXX.XXX.XXX  [certinfo] => Array  (  )  [primary\_port] => 443  [local\_ip] =>  [local\_port] => 0  )  ) |