



ITA_利用手順マニュアル

Conductor

—第1.9版—

免責事項

本書の内容はすべて日本電気株式会社が所有する著作権に保護されています。

本書の内容の一部または全部を無断で転載および複写することは禁止されています。

本書の内容は将来予告なしに変更することがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任を負いません。

日本電気株式会社は、本書の内容に関し、その正確性、有用性、確実性その他いかなる保証もいたしません。

商標

- ・ LinuxはLinus Torvalds氏の米国およびその他の国における登録商標または商標です。
- ・ Red Hatは、Red Hat, Inc.の米国およびその他の国における登録商標または商標です。
- ・ Apache、Apache Tomcat、Tomcatは、Apache Software Foundationの登録商標または商標です。
- ・ Ansibleは、Red Hat, Inc.の登録商標または商標です。
- ・ AnsibleTowerは、Red Hat, Inc.の登録商標または商標です。
- ・ Terraformは、HashiCorpの登録商標または商標です。

その他、本書に記載のシステム名、会社名、製品名は、各社の登録商標もしくは商標です。

なお、® マーク、TM マークは本書に明記しておりません。

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

目次

はじめに	4
1 ITA Conductor の概要	5
2 ITA Conductor のメニュー、画面構成	6
2.1 ITA Conductor メニュー 一覧	6
3 ITA Conductor 利用手順	7
3.1 作業フロー	7
4 機能・操作方法説明	9
4.1 ITA Conductor	9
4.1.1 Conductor インターフェース情報	9
4.1.2 Conductor 通知先定義	10
4.1.3 Conductor クラス一覧	12
4.1.4 Conductor クラス編集	13
4.1.5 Conductor 作業実行	29
4.1.6 Conductor 作業確認	32
4.1.7 Conductor 作業一覧	35
4.1.8 Conductor 定期作業実行	35
5 付録	39
5.1 Conductor 通知先定義	39
5.1.1 Conductor 通知先定義設定例	39
5.1.2 通知ログ出力例	42

はじめに

本書は、ITA の Conductor の機能および操作方法について説明します。

1 ITA Conductor の概要

本章では Conductor メニューの機能、操作方法について説明します。

Conductor では、ITA を利用して作業を行う上で共通に必要な以下の機能を提供します。

2 ITA Conductor のメニュー、画面構成

本章では、ITA Conductor のメニュー、画面構成について説明します

2.1 ITA Conductor メニュー 一覧

ITA 共通/Conductor のメニューを以下に示します。

表 2.1-1 ITA Conductor 画面一覧

No	メニュー グループ	メニュー	説明
1	Conductor	Conductor インターフェース情報	Conductor 作業実行時の Movement で共有するディレクトリパスなどをメンテナンス(閲覧/登録/更新/廃止)できます
2		Conductor クラス一覧	Conductor クラスをメンテナンス(閲覧/廃止)できます 「詳細」をクリックすると Conductor クラス編集メニューに遷移します
3		Conductor クラス編集	Conductor クラスを編集できます
4		Conductor 作業実行	Conductor 作業を実行できます
5		Conductor 作業確認	Conductor 作業の実行結果を確認できます
6		Conductor 作業一覧	Conductor 一覧(実行履歴)を閲覧できます 「詳細」をクリックすると Conductor 作業確認に遷移します
7		Conductor 定期作業実行	スケジュールに従って定期的に行う Conductor 作業を管理します。

3 ITA Conductor 利用手順

3.1 作業フロー

ITA Conductor における標準的な作業フローは以下のとおりです。
各作業の詳細は次項に記載しています。

- 「機器情報」、「オペレーション」の登録方法については、「利用手順マニュアル 基本コンソール」を参照してください。
- Movement の登録方法については、各 Driver の利用手順マニュアルを参照してください。
- Conductor 実行時の各 Movement で共有するディレクトリパスが利用可能です。
Movement 間で情報伝達する必要がある場合、共有するディレクトリパスを利用する事で情報の受け渡しが可能となります。
共有対象については、「Ansible driver」と「Terraform driver」を対象としています。
「Ansible driver」に関する詳細は、別紙、「利用手順マニュアル Ansible-driver」を参照してください。
「Terraform driver」に関する詳細は、別紙、「利用手順マニュアル Terraform-driver」を参照してください。
- 「Conductor call」、「Symphony call」で実行した作業フローについては、それぞれ個別に共有ディレクトリパスを持ちます。（作業フローを跨いでいる Movement については、共有の対象外となります。）

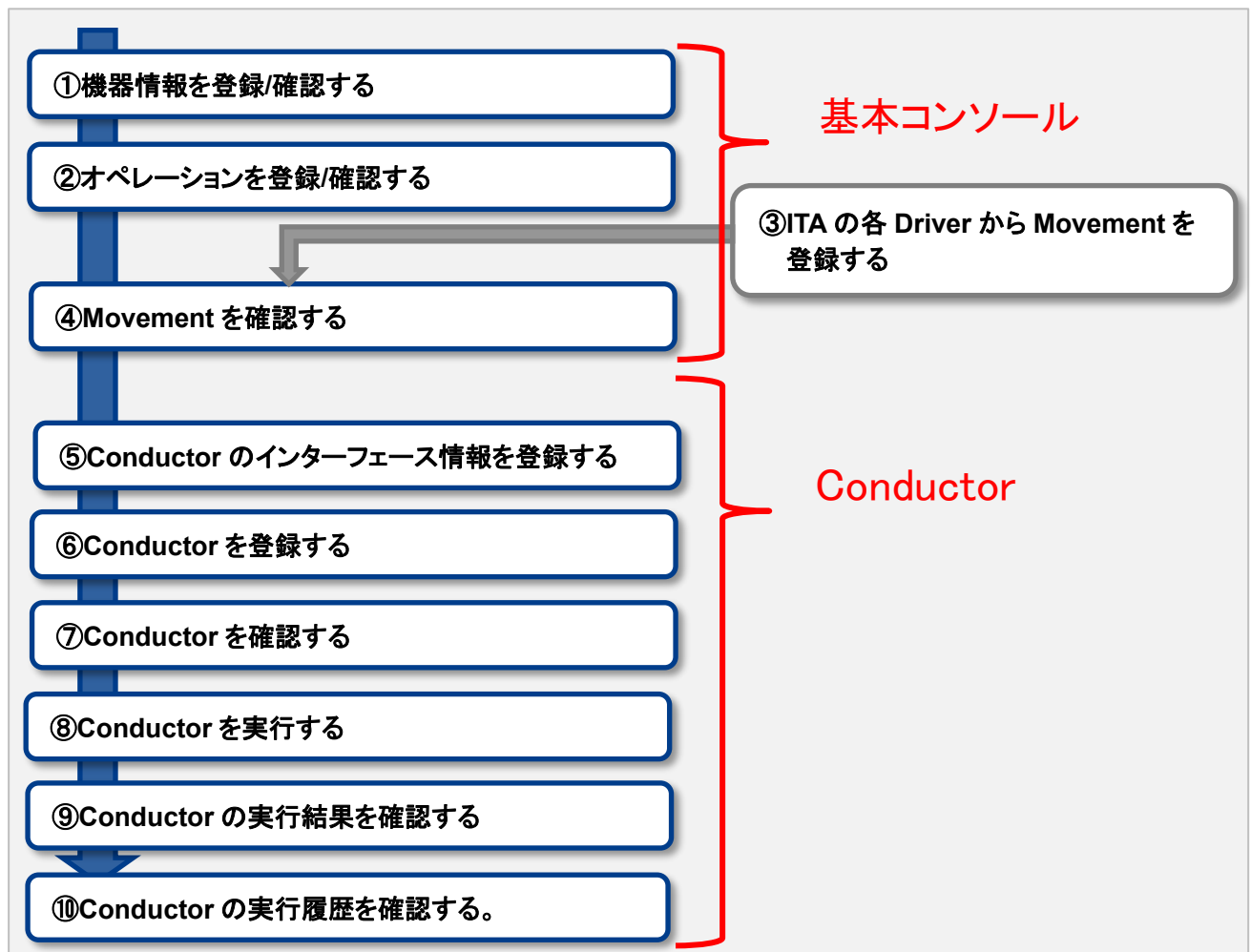


図 3.1-1 作業フロー

4 機能・操作方法説明

4.1 ITA Conductor

4.1.1 Conductor インターフェース情報

- (1) 「Conductor インターフェース情報」メニューでは、Conductor から実行される各 Movement で共有するディレクトリのパスと「Conductor 作業確認」メニューのリフレッシュ間隔を設定します。

図 4.1- 1「Conductor インターフェース情報」メニュー

- (2) 「Conductor インターフェース情報」メニュー > 「一覧」サブメニューの詳細は以下のとおりです。

表 4.1- 1「一覧」サブメニュー

項目	説明	入力必須	入力形式	制約事項
データリレイストレージパス	Conductor 実行時、各 Movement で共有するディレクトリを、ITA サーバから見たディレクトリパスで入力します。 Ansible、Ansible-Tower サーバから見たパスは、Ansible-Driver の利用手順マニュアルのインターフェース情報を参照して下さい。 Terraform との連携は共有ディレクトリをせずに REST API で情報を取得するため、Terraform-Driver ではこのディレクトリパスが直接使用されます。	○	手動入力	最大長 128 バイト
状態監視周期(単位ミリ秒)	「4.1.5 Conductor 作業実行」の表示をリフレッシュする間隔を入力します。通常は 3000 ミリ秒程度が推奨値です。	○	手動入力	最小値 1000 ミリ秒
備考	自由記述欄です	-	手動入力	-

4.1.2 Conductor 通知先定義

「Conductor 通知先定義」メニューでは、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。通知は Webhook を利用して行います。

登録した Conductor 通知先定義は、「Conductor クラス編集」の「Notice」にて、Conductor 実行中に、通知を行うステータスに対して設定を行います。

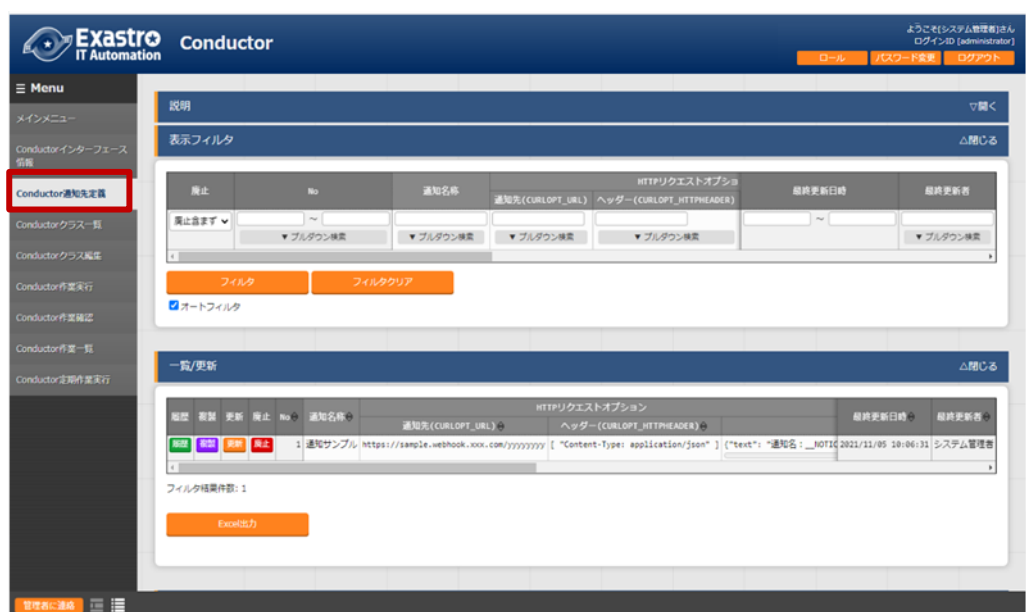


図 4.1- 2「Conductor 通知先定義」メニュー

(3) 「Conductor 通知先定義」メニュー > 「一覧」サブメニューの詳細は以下のとおりです。

表 4.1- 2「一覧」サブメニュー

項目	説明	入力 必須	入力形式	制約事項
通知名称	通知名称を入力してください。	○	手動入力	
通知先(CURLOPT_URL)	通知先の URL を入力してください。	○	手動入力	
ヘッダー (CURLOPT_HTTPHEADER)	HTTP ヘッダフィールドを JSON 形式で入力してください。	○	手動入力	
メッセージ (CURLOPT_POSTFIELDS)	メッセージ内容を通知先のサービスの仕様に沿って入力してください。	○	手動入力	※
PROXY / URL (CURLOPT_PROXY)	PROXY の設定が必要な場合、URL を入力してください。		手動入力	
PROXY / PORT (CURLOPT_PROXYPORT)	PROXY の設定が必要な場合、PORT を入力してください。		手動入力	
作業確認 URL(FQDN)	作業確認用 URL の予約変数で使用する FQDN を入力してください。		手動入力	
その他	JSON 形式で入力してください。 使用できるオプションについては、curl_setopt() のオプションに対応するものは使用可能です。 PHP の cURL 関数について参照してください。		手動入力	
開始日時	通知を抑止したい場合、入力してください。		手動入力	
終了日時	通知を抑止したい場合、入力してください。		手動入力	
備考	自由記述欄です			

※ メッセージ内容で利用可能な ITA 独自変数は、以下の表参照してください。
Teams、Slack の入力例について、「5.1.1Conductor 通知先定義設定例」を参照ください。

表 4.1- 3 Conductor 通知先定義 ITA 独自変数

ITA 独自変数	変数指定内容	備考
__CONDUCTOR_INSTANCE_ID__	Conductor インスタンス ID	
__CONDUCTOR_NAME__	Conductor インスタンス名	
__OPERATION_ID__	オペレーション ID	
__OPERATION_NAME__	オペレーション名	
__STATUS_ID__	ステータス ID	
__STATUS_NAME__	ステータス名	
__EXECUTION_USER__	実行ユーザー	
__TIME_BOOK__	予約日時	
__TIME_START__	開始日時	
__TIME_END__	終了日時	
__JUMP_URL__	作業確認 URL	※

※作業確認 URL の出力内容は、「作業確認 URL(FQDN)」の入力内容が使用され以下の形で出力されます。

<作業確認 URL(FQDN)>/default/menu/01_browse.php?no=2100180005&conductor_instance_id=X

例:作業確認 URL(FQDN)の値を「http://exastro-it-automation.local」としていた場合

http://exastro-it-automation.local/default/menu/01_browse.php?no=2100180005&conductor_instance_id=X

4.1.3 Conductor クラス一覧

- (1) 「Conductor クラス一覧」メニューでは、登録済みの Conductor クラスを参照/廃止できます。
「一覧」サブメニューの「詳細」ボタンをクリックすると、その Conductor クラスの編集画面 [4.1.4 Conductor クラス編集]へ遷移できます。

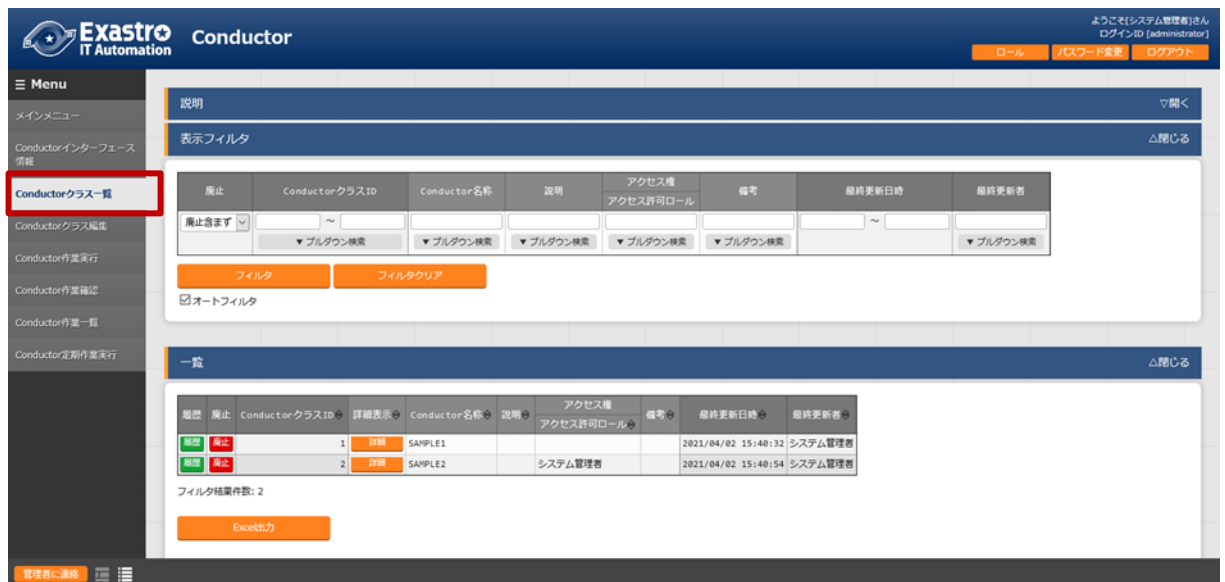


図 4.1- 3「Conductor クラス一覧」メニュー

4.1.4 Conductor クラス編集

(1) 「Conductor クラス編集」メニューについて

- ・ Conductor 名称と、ワークフローを構成する各パーツ(以下、Node)を登録します。
- ・ 画面上のモードについて、以下 2 種のモードがあります。

表 4.1- 4 モード一覧

モード	説明
EDIT	<ul style="list-style-type: none"> ・ Conductor クラスを編集できるモード ・ 「Conductor クラス編集」メニューのデフォルトのモード ・ EDIT モードから「登録」ボタン/「更新」ボタン押下後に、VIEW モードへ変更
VIEW	<ul style="list-style-type: none"> ・ Conductor クラスを閲覧のみできるモード ・ 「Conductor クラス一覧」メニューの「詳細」ボタンから画面遷移後のモード ・ VIEW モードから「編集」ボタン押下後に、EDIT モードへ変更

各モードで可能な操作については「表 4.1- 18「Conductor クラス編集」メニュー実行操作一覧」を参照。

- ・ 選択した Node により、「B) 詳細情報」に表示される内容が変わります。



図 4.1- 4「Conductor クラス編集」メニュー(モード: EDIT)

A) Node 一覧

- ・ 画面右側下部の領域には、利用可能な Node が表示されます。
- ・ 以下のタブから構成されます。

➤ Movemet タブ

- ✧ 登録済みの Movement の ID と名称の一覧




➤ Function タブ

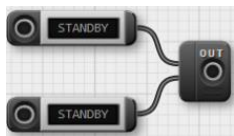
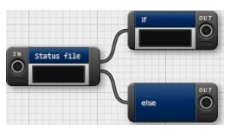

- ✧ Conductor end
- ✧ Conductor pause
- ✧ Conductor call
- ✧ Symphony call

- ◇ Conditional branch
- ◇ Parallel branch
- ◇ Parallel merge
- ◇ Status File branch

- ・ 各 Node について、詳細は以下の通りです。

表 4.1- 5 Node 一覧

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。 ・正常終了 ・異常終了 ・緊急停止 ・準備エラー ・想定外エラー ・SKIP 完了 ・警告終了
	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。

	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。

de における制約事項は以下の通りです。

- 登録/更新を行うには、全ての Node の IN/OUT が接続されている必要があります。

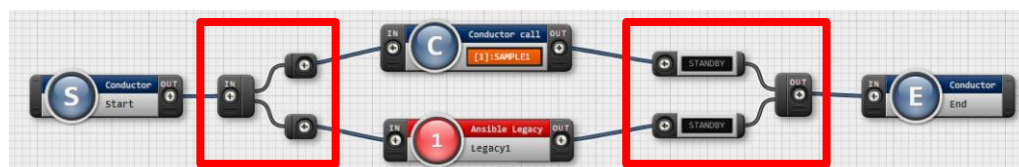


図 4.1- 5 Node 制約事項(正常例:Parallel branch)

- Parallel merge を使用する場合、Parallel branch を使用している必要があります。

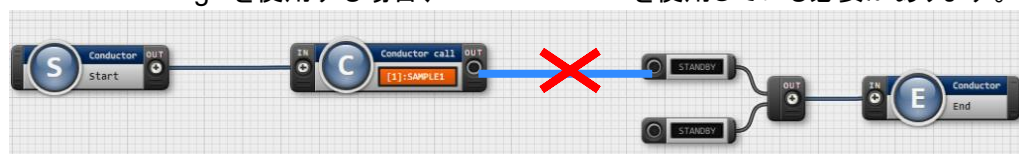


図 4.1- 6 Node 制約事項(NG 例:Parallel branch)

- Conditional branch で分岐されたフローについて Parallel merge でマージすることはできません。

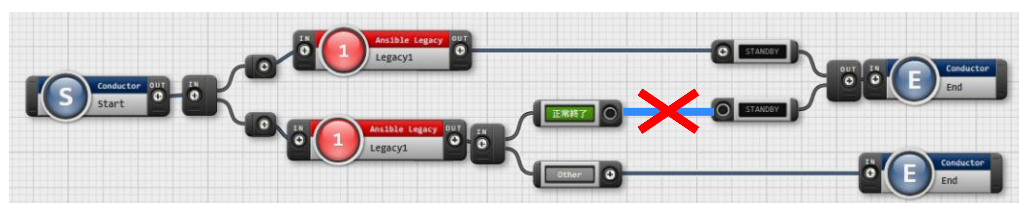


図 4.1- 7 Node 制約事項(NG 例:Conditional branch)

- Parallel branch、Conditional branch、Parallel merge、Conductor pause について、連続

して同じ種類の Node を接続する事はできません。

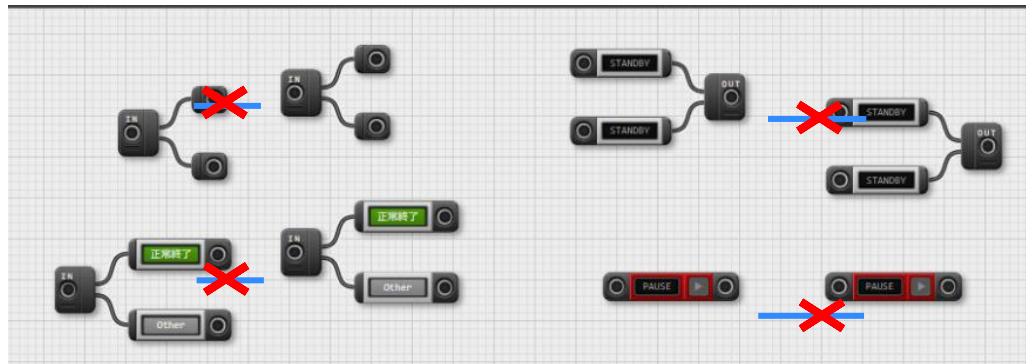


図 4.1- 8 Node 制約事項(NG 例:連続使用)

- Conductor call について、更新中の Conductor を Conductor call で指定し、更新することはできません。

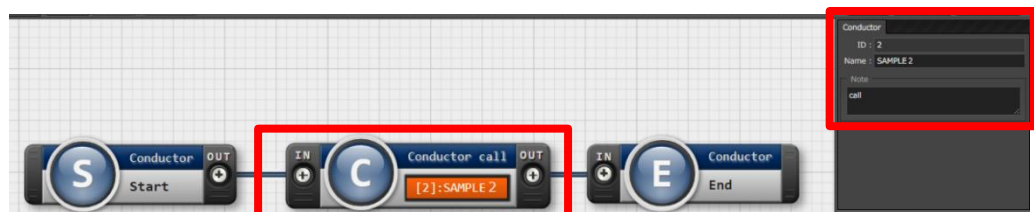


図 4.1- 9 Node 制約事項(NG 例:Conductor call)

- ・ 各 Node を Node 一覧からドラッグ & ドロップで追加することが可能です。
- ・ Node 選択時「B) 詳細情報」に表示される「Note」欄には、処理説明やコメントをメモすることが可能です。
- ・ 「Note」欄の記述は処理実行に影響はありません。Web 上でのみ参照できるメモ欄です。
- ・ Node 設定後、「登録」ボタンを押下して Conductor クラスを登録します

B) 詳細情報

- ・ 画面右側上部の領域には、選択している Node の詳細情報が表示されます。
- ・ 選択している Node によってタブの名称が変わります。

i. 「Conductor 名称」タブ

- ・ Node 未選択の場合表示されます。
- ・ タブ内の項目は以下の通りです。

表 4.1- 6 「Conductor 名称」タブ

項目	説明	入力 必須	入力形式	制約事項
ID	Conductor に対応した一意の ID が自動採番されます。	－	自動入力	－
Name	任意の Conductor クラス名称を入力します。	○	手動入力	－
Notice	実行する通知を選択します。	－	選択	※

	各ステータスに対して、実行する通知を複数選択できます。			
Role	この Conductor へアクセス可能なロールを選択します。 ロールが 1 つも選択されていない場合は、すべてのロールがアクセス可能となります。 「Permission role select」ボタンを押下して表示される一覧から任意の値を選択できます。	-	選択	-
Note	Conductor クラスに対する説明やコメントを入力します。	-	手動入力	-

※選択できる通知は、「4.1.2Conductor 通知先定義」で登録したものが対象となります。

図 4.1- 10 「Conductor」名称タブ

図 4.1- 11 「Notice」のポップアップ

図 4.1- 12 「Permission role」のポップアップ

- ii. 「Movement」タブ
- ・ 「A)Node 一覧」における「Movement」タブ内の Node を選択した場合表示されます。
 - ・ タブ内の項目は以下の通りです。

表 4.1- 7「Movement」タブ

項目	説明	入力 必須	入力形式	制約事項
Movement ID	選択した Movement の ID が表示されます。	-	自動入力	-
Orchestrator	選択した Movement のオーケストレータ名が表示されます。	-	自動入力	-
Name	選択した Movement の名称が表示されます。	-	自動入力	-
Default skip	対象作業をスキップする場合にチェックします。「Conductor 作業実行」メニューにて、変更可能なパラメータです。	-	手動入力	
Operation	「Select」ボタンを押下して表示される一覧から任意の値を選択できます。 選択したオペレーション名が表示されます。	-	選択	-
Note	Node に対する説明やコメントを入力できます。	-	手動入力	-

図 4.1- 13「Movement」タブ

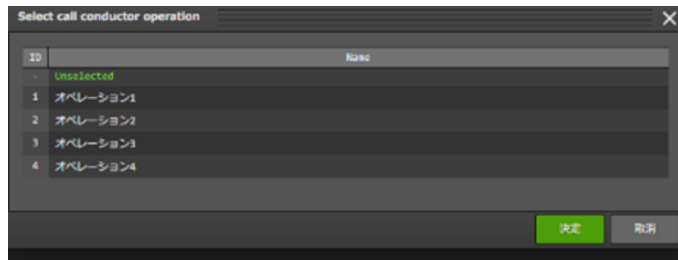


図 4.1- 14 「Operation select」のポップアップ

- iii. 「Function」タブ
- ・ 「A) Node 一覧」における「Function」タブ内の「Conductor start」「Conductor end」「Conductor pause」を選択した場合表示されます。
 - ・ タブ内の項目は以下の通りです。

表 4.1- 8 「Function」タブ

項目	説明	入力 必須	入力形式	制約事項
Type	選択した Node のタイプが表示されます。	－	自動入力	－
Note	Node に対する説明やコメントを入力できます。	－	手動入力	－

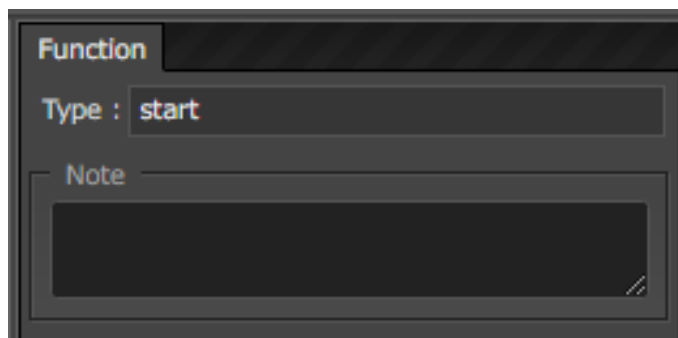


図 4.1- 15 「Function タブ」

- iv. 「Conductor call」タブ
- ・ 「A) Node 一覧」における「Function」タブ内の「Conductor call」を選択した場合表示されます。
 - ・ タブ内の項目は以下の通りです。

表 4.1- 9 「Conductor call」タブ

項目	説明	入力 必須	入力形式	制約事項
Default skip	対象作業をスキップする場合にチェックします。 Conductor 作業実行画面にて、変更可能なパラメータです。	－	手動入力	－
Conductor	「Conductor Select」ボタンを押下して表示される一覧から Conductor を選択でき	○	選択	－

	ます。 選択した Conductor クラス名称が表示されます。			
Operation	「Conductor Select」ボタンを押下して表示される一覧から、任意のオペレーションを選択できます。 指定したオペレーション名が表示されます。	-	選択	-
Note	Node に対する説明やコメントを入力できます。	-	手動入力	-

図 4.1- 16 「Conductor call」タブ

図 4.1- 17「Conductor select」のポップアップ

- v. 「Symphony call」タブ
 - ・ 「A)Node 一覧」における「Function」タブ内の「Symphony call」を選択した場合表示されます。

- ・ タブ内の項目は以下の通りです。

表 4.1- 10 「Symphony call」タブ

項目	説明	入力 必須	入力形式	制約事項
Default skip	対象作業をスキップする場合にチェックします。 Conductor 作業実行画面にて、変更可能なパラメータです。	－	手動入力	－
Symphony	「Symphony Select」ボタンを押下して表示される一覧から Symphony を選択できます。 選択した Symphony クラス名称が表示されます。	○	選択	－
Operation	「Operation Select」ボタンを押下して表示される一覧からオペレーションを選択できます。 選択したオペレーション名が表示されます。	－	選択	－
Note	Node に対する説明やコメントを入力できます。	－	手動入力	－

図 4.1- 18 「Symphony call」タブ

図 4.1- 19 「Symphony select」のポップアップ

- vi. 「Parallel branch」タブ
- ・ 「A)Node 一覧」における「Function」タブ内の「Parallel branch」を選択した場合表示されます。
 - ・ タブ内の項目は以下の通りです。

表 4.1- 11 「Parallel branch」タブ

項目	説明	入力 必須	入力形式	制約事項
case	分岐数を設定します。「Add」ボタン/ 「Delete」ボタンを押下して、分岐を増減 します。 デフォルトの分岐数は2です。2以下の値 は設定できません。	－	選択	－
Note	Node に対する説明やコメントを入力でき ます。	－	手動入力	－

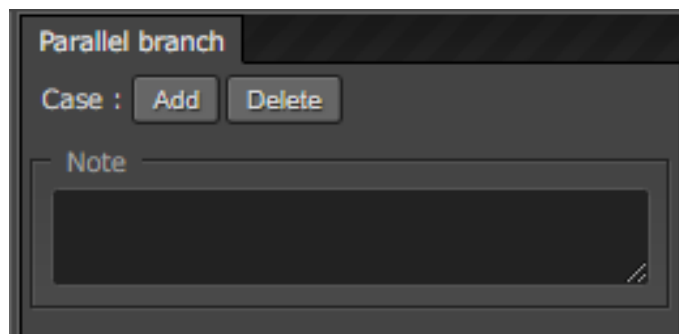


図 4.1- 20 「Parallel branch」タブ

- vii. 「Conditional branch」タブ
- ・ 「A)Node 一覧」における「Function」タブ内の「Conditional branch」を選択した場合表示されます。
 - ・ タブ内の項目は以下の通りです。

表 4.1- 12 「Conditional branch」タブ

項目	説明	入力 必須	入力形式	制約事項				
case (1-6)	Movement、Conductor call、Symphony call の 実行結果による条件分岐を設定します。 ドラッグアンドドロップで設定を変更できます。 デフォルトは以下の通りです。 <table><tr><td>Case1</td><td>正常終了</td></tr><tr><td>Other</td><td>異常終了、緊急停止、準備工</td></tr></table>	Case1	正常終了	Other	異常終了、緊急停止、準備工	-	選択	※
Case1	正常終了							
Other	異常終了、緊急停止、準備工							

		ラー、想定外エラー、skip 完了、警告終了			
Note	Node に対する説明やコメントを入力できます。		-	手動入力	-

※警告終了は、Conductor にのみ対応したステータスとなります。「Movement」、「Symphony call ノード」と「Conditional branch」を接続した場合、警告終了の Case への分岐へ進行することはありません。

図 4.1- 21 「Conditional branch」タブ

viii. 「Parallel Merge」タブ

- ・ 「A) Node 一覧」における「Function」タブ内の「Parallel merge」を選択した場合表示されます。
- ・ タブ内の項目は以下の通りです。

表 4.1- 13 「Merge」タブ

項目	説明	入力必須	入力形式	制約事項
case	分岐数を設定します。「Add」ボタン/「Delete」ボタンを押下して、分岐を増減します。 デフォルトの分岐数は2です。2以下の値は設定できません。	-	選択	
Note	Node に対する説明やコメントを入力できます。	-	手動入力	-

図 4.1- 22 「Merge」タブ

ix. 「End」タブ

- ・ 「A)Node 一覧」における「Function」タブ内の「End」を選択した場合表示されます。
- ・ タブ内の項目は以下の通りです。

表 4.1- 14 「End」タブ

項目	説明	入力 必須	入力形式	制約事項
End status	End まで処理された際に、選択されたステータスが、Conductor のステータスへ反映されます。 <ul style="list-style-type: none"> - 正常終了 (デフォルト値) - 警告終了 - 異常終了 複数の End ノードまで処理された場合、反映されるステータスの優先度は以下です。 優先度: 正常終了 < 警告終了 < 異常終了	-	選択	
Note	Node に対する説明やコメントを入力できます。	-	手動入力	-

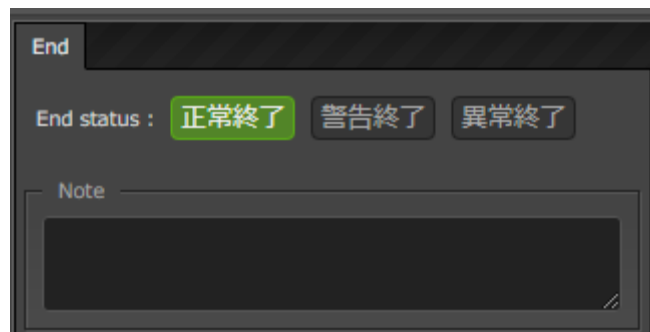


図 4.1- 23 「End」タブ

x. 「Status file branch」タブ

- ・ 「A)Node 一覧」における「Function」タブ内の「Status file branch」を選択した場合表示されます。
- ・ タブ内の項目は以下の通りです。

表 4.1- 15 「Status file branch」タブ

項目	説明	入力 必須	入力形式	制約事項
If / elseif	Movement のステータスファイルによる条件分岐を設定します。 「Add」ボタン/「Delete」ボタンを押下して、分岐を増減します。 デフォルトの分岐は「if」と「else」です。	-	手動入力	※
Note	Node に対する説明やコメントを入力できます。	-	手動入力	-

図 4.1- 24 「Status file branch」タブ

※参照するステータスファイルについて

- 参照するステータスファイルは、各 Movment の作業結果ディレクトリ配下の「MOVEMENT_STATUS_FILE」を参照します。
- ステータスファイルが存在しない場合、「else」側の処理を行います。
- ステータスファイル内の内容が、複数行(改行コードを含む)場合、改行コード以降は、除外した値を評価対象とします。

例) 改行含むステータスファイルの内容

```
1
23
4
```

ステータスファイルの内容として、「1」として、評価を行います。

表 4.1- 16 ステータスファイル ITA 独自変数

ITA 独自変数	変数指定内容	制約事項
__movement_status_filepath__	作業結果ディレクトリ配下の「MOVEMENT_STATUS_FILE」のパス	※







※「Ansible-Legacy」、「Ansible-Pioneer」、「Ansible-LegacyRole」で対応しています。

xi. 「Node」タブ

- ・ 「A)Node 一覧」における「Movement」タブおよび「Function」タブ内の Node を複数選択した場合表示されます。
- ・ 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。
- ・ タブ内の項目は以下の通りです。

表 4.1- 17 「Node」タブ

項目	説明	入力必須	入力形式	制約事項
	複数選択した Node を左揃えに整列します。	-	選択	-
	複数選択した Node を左右中央揃えに整列します。	-	選択	-

	複数選択した Node を右揃えに整列します。	-	選択	-
	複数選択した Node を上揃えに整列します。	-	選択	-
	複数選択した Node を上下中央揃えに整列します。	-	選択	-
	複数選択した Node を下揃えに整列します。	-	選択	-
	複数選択した Node を左右等間隔にします。	-	選択	-
	複数選択した Node を上下等間隔にします。	-	選択	-

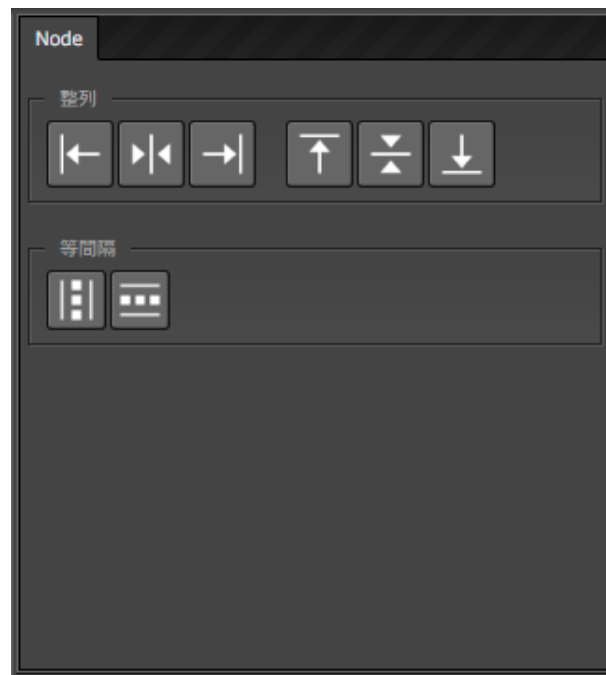


図 4.1- 25 「Node」タブ

- ・ 「Conductor クラス編集」メニューで実行可能な操作は以下の通りです。

表 4.1- 18「Conductor クラス編集」メニュー実行操作一覧

項目	説明	新規	更新		備考
		EDIT	VIEW	EDIT	
新規	初期状態へ戻します。	○	-	-	

保存	描画されている状態を、ファイル形式で保存します。	○	-	-	
読み込み	保存したファイルを読み込み、状態を復元します。	○	-	-	
取り消し	直前の処理を取り消します。	○	-	○	
やり直し	直前の取り消しをやり直します。	○	-	○	
ノード削除	選択している Node を削除します。	○	-	○	
登録	登録を実施します。	○	-	-	
編集	EDIT モードへ変更し、Conductor の編集を行います。	-	○	○	
流用新規	登録済の Conductor を流用して、新規作成が行えます。	-	○	○	
更新	編集内容を更新します。	-	-	○	
再読込	変更をキャンセルし、変更前の状態へ戻します。	-	-	○	
キャンセル	変更をキャンセルし、VIEW モードへ変更します。		-	○	

(2) 「VIEW」モードについて

「Conductor クラス一覧」メニューから遷移した場合や、登録が完了した場合は、以下の画面が表示されます。

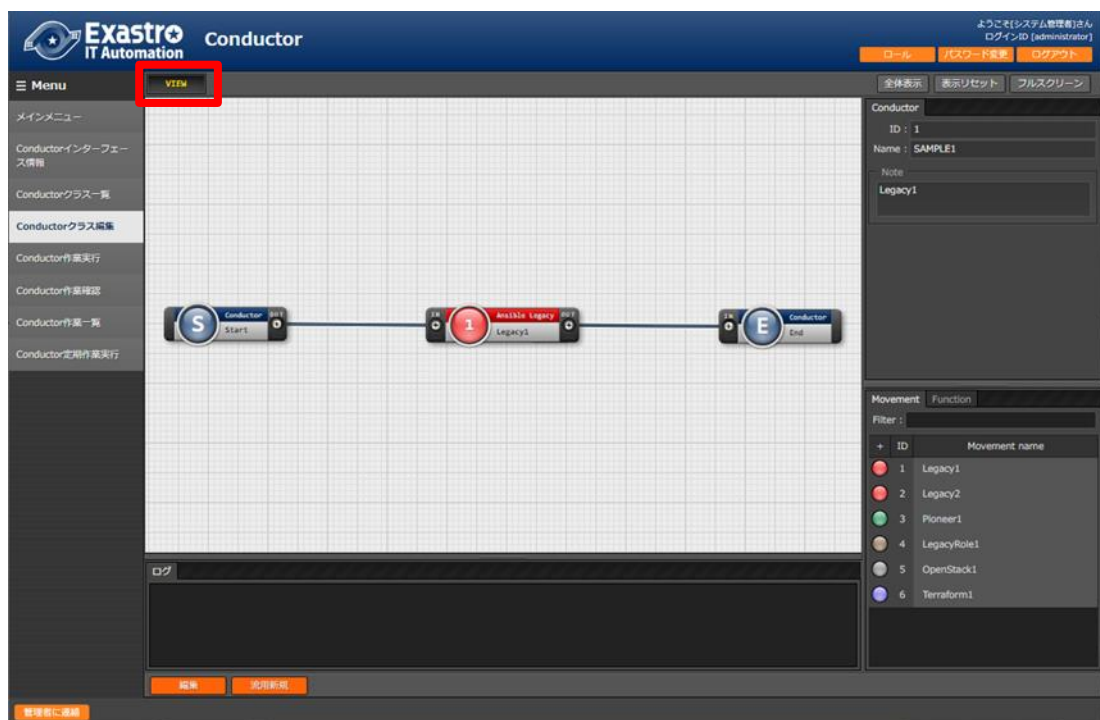


図 4.1- 26 「Conductor クラス編集」メニュー（「VIEW」モード）

表 4.1- 19 「VIEW」モード

項目	説明
「編集」ボタン	登録済みの Conductor を編集できます。
「流用新規」ボタン	登録済みの Conductor をコピーして新規作成が行えます。

(3) 「EDIT」モード

「編集」ボタンを押下した場合は、以下の画面が表示されます。

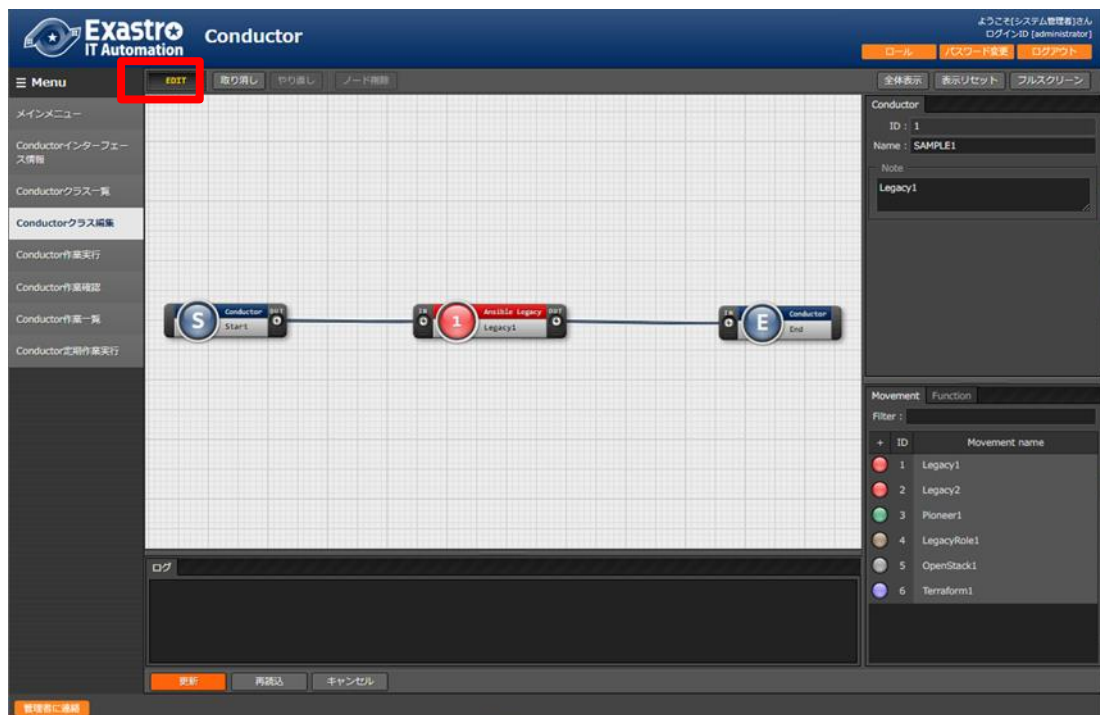


図 4.1- 27 「Conductor クラス編集」メニュー(「EDIT」モード)

表 4.1- 20 「EDIT」モード

項目	説明
「全体表示」ボタン	Node すべてが表示される縮尺で表示されます。
「表示リセット」ボタン	「Conductor start」を基準に表示がリセットされます。
「フルスクリーン」ボタン	ブラウザの表示がフルスクリーンになります。 ※フルスクリーン時は「フルスクリーン解除」ボタンに変わります。
「更新」ボタン	編集内容が保存されます。
「再読み込み」ボタン	編集内容が破棄されて登録内容の状態に戻ります。
「キャンセル」ボタン	「編集」ボタン押下前の状態に戻ります。

4.1.5 Conductor 作業実行

(1) 「Conductor 作業実行」メニューでは、Conductor 実行の指示を行います。

- ・ 「Conductor[一覧]」サブメニューには、「[4.1.3 Conductor クラス一覧](#)」で登録した Conductor が表示されます。
- ・ 「オペレーション[一覧]」サブメニューには、「基本コンソール」メニューグループ > 「オペレーション一覧」メニューで登録したオペレーションが表示されます。
※ 別紙、「利用手順マニュアル 基本コンソール」を参照。
- ・ 「Conductor[一覧]」サブメニュー及び「オペレーション[一覧]」サブメニューのラジオボタンからそれぞれ選択し「実行」ボタンを押下すると「[4.1.6 Conductor 作業確認](#)」に遷移し、作業のトレースが始まります。
- ・ 「スcheduling」サブメニューにて予約日時を入力して「実行」ボタンを押下すると、作業予約が作られます。登録情報は「[4.1.7 Conductor 作業一覧](#)」で確認できます。
※ 現在時刻より過去の日時は入力できません
- ・ Movement、Conductor Call、Symphony Call の Operation, skip のみ、設定値を変更可能です。
※ Conductor 編集で登録したデータへ変更は反映されません。作業実行にのみ反映されます。
- ・ 実行した Conductor に設定されるアクセス権について、実行時に選択した Conductor、オペレーションに設定されたアクセス権の共通するロールを継承します。共通するロールが存在しない場合、作業実行できません。

- ・ 「Conductor 実行」メニューの共通項目は以下のとおりです。

表 4.1- 21 「Conductor 実行」共通項目一覧

項目	説明	入力 必須	入力 形式	制約事項
予約日時	Conductor の実行予定日時を指定します。	-	手動入力	現在時刻より過去の日時は入力不可
Conductor[一覧]	「4.1.7Conductor クラス一覧」で登録した Conductor が表示されます。	○	ラジオ ボタン	
オペレーション[一覧]	「4.1.4 投入オペレーション一覧」で登録したオペレーションが表示されます。	○	ラジオ ボタン	
Skip	対象作業をスキップする場合にチェックします。 ※以下「オペレーションの指定について」参照	-	チェック ボックス	
Operation	※以下「オペレーションの指定について」参照	-	手動入力	
Notice	通知の設定を確認できます。	-	ボタン	
実行	登録した Conductor を実行します。	○	ボタン	

- オペレーションの指定について
「Operation 欄の select」をクリックすると、Operation のリストが表示されます。
画面のラジオボタンで指定したオペレーションのオペレーション ID とは別のオペレーションを指定することができます。
これにより、その Movement の属するオーケストレータの「代入値管理」メニュー（例：ITAAnsible-Legacy コンソールの「代入値管理」メニュー）で、ほかのオペレーション ID のものとして登録した「具体値」を代入して実行することができます。
Conductor クラス編集画面で個別指定したオペレーション ID は Conductor 登録/更新ボタンにより設定が保存されます。
また、Conductor 実行画面でも実行前に個別指定ができ、既に Conductor クラス編集メニューで個別指定登録をして保存されているオペレーション ID についても更に変更を行い Conductor 実行することができます。
ただし、Conductor 実行画面で個別指定したオペレーション ID は実行時のみの反映となり、設定は保存されません。
同じ Movement を流用し、別なサーバを操作したい時などにご活用ください。
- Skip について
Skip のチェックを変更することができます。
Conductor クラス編集画面で Default Skip の設定は登録/更新ボタンにより設定が保存されます。
また、Conductor 実行画面でも実行前に個別指定ができ、既に Conductor クラス編集メニューで保存されている Skip について変更を行い Conductor 実行することができます。
ただし、Conductor 実行画面では実行時のみの反映となり、設定は保存されません。
一時的に、処理を飛ばして、又は実施して、作業実行したい時などにご活用ください。
- 実行する作業のアクセス権限について
「Conductor 実行」メニューに表示された「Movement」および「Conductor Call」「Symphony Call」内で呼び出されるすべての作業および個別指定されたオペレーションについて、アクセス権限が無い対象が一つでも含まれている場合は「実行」ボタンを押下した際にバリデーションエラーになります。

4.1.6 Conductor 作業確認

- (1) 「Conductor 作業確認」メニューでは、Conductor の実行状態を表示します。
- ・ 「4.1.7Conductor 作業一覧」の「詳細」ボタンを押下すると、選択した Conductor 作業の処理状況をモニター表示します。状況に応じて「予約取消」、「停止解除」や「緊急停止」の投入が可能です。
 - ・ Node を選択すると、画面右側のタブ名が「Node」となり、実行状況を表示します。
 - ・ 実行中以降のステータスになっている実行状況サークルを押下すると、各ドライバの「作業状態確認」メニューに遷移し作業実行状況の詳細を確認することが可能です。
 - ・ 「Movement」、「Conductor Call」、「Symphony Call」の Node については、「Node」タブ内の「Operation status」からも「作業状態確認」メニューに遷移できます。
 - ・ 「Movement」、「Conductor Call」、「Symphony Call」の Node でのエラーが発生した場合、ログにメッセージが表示されます。

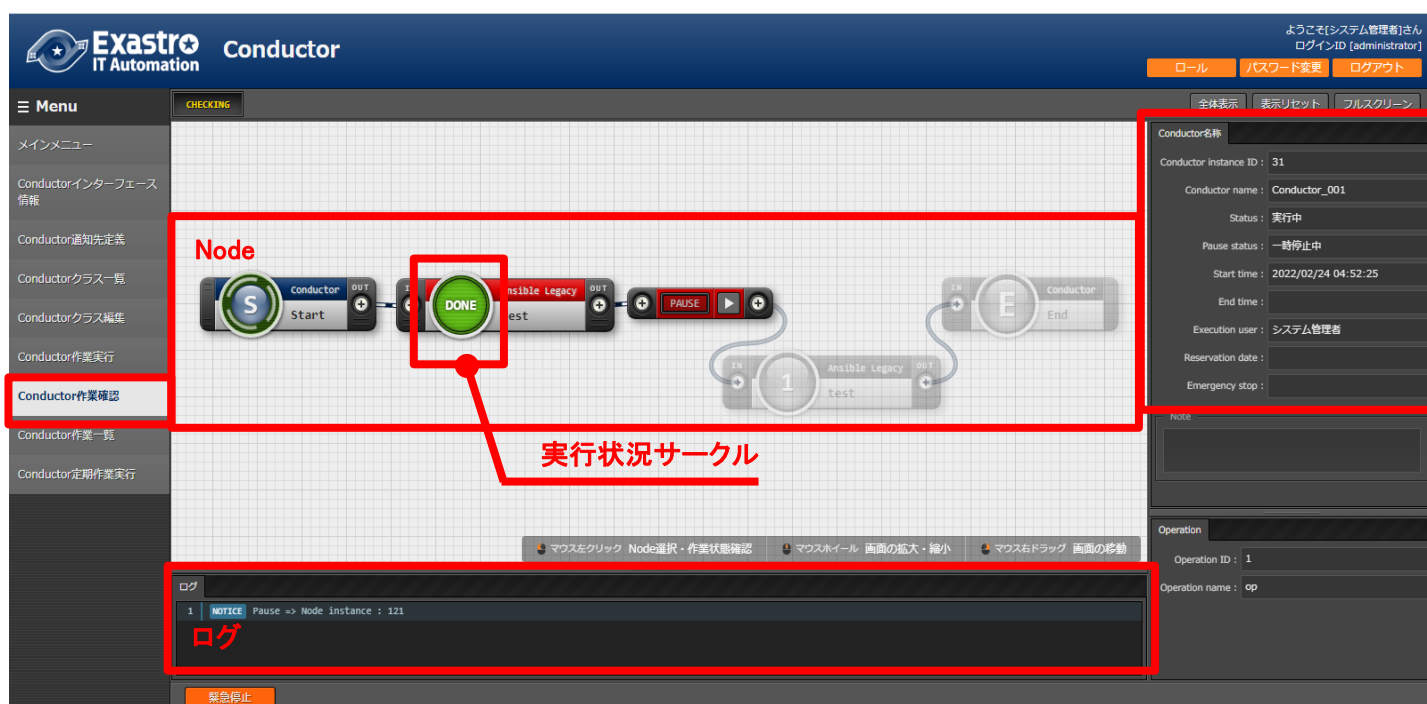


図 4.1- 29 「Conductor 作業確認」メニュー

※ 「Conductor 作業実行」メニューで実行した、作業実行済みの Conductor を「Conductor クラス編集」メニューで編集すると、作業実行時の Conductor と異なる状態となるため「詳細」ボタンを押下しても処理状況が表示されない場合があります。
作業実行済みの Conductor を編集して再度実行する場合は、「Conductor クラス編集」メニューの「流用新規」ボタンにて、別の Conductor を作成してご利用いただくことを推奨します。

- ・ 選択した Conductor 作業に予約日時が設定されていて、かつ未実行の場合は、「予約取消」ボタンが表示されます。
- ・ 「予約取消」ボタンを押下すると、「4.1.7Conductor 作業一覧」で確認できるステータスが「予約取消」となり、実行されなくなります。

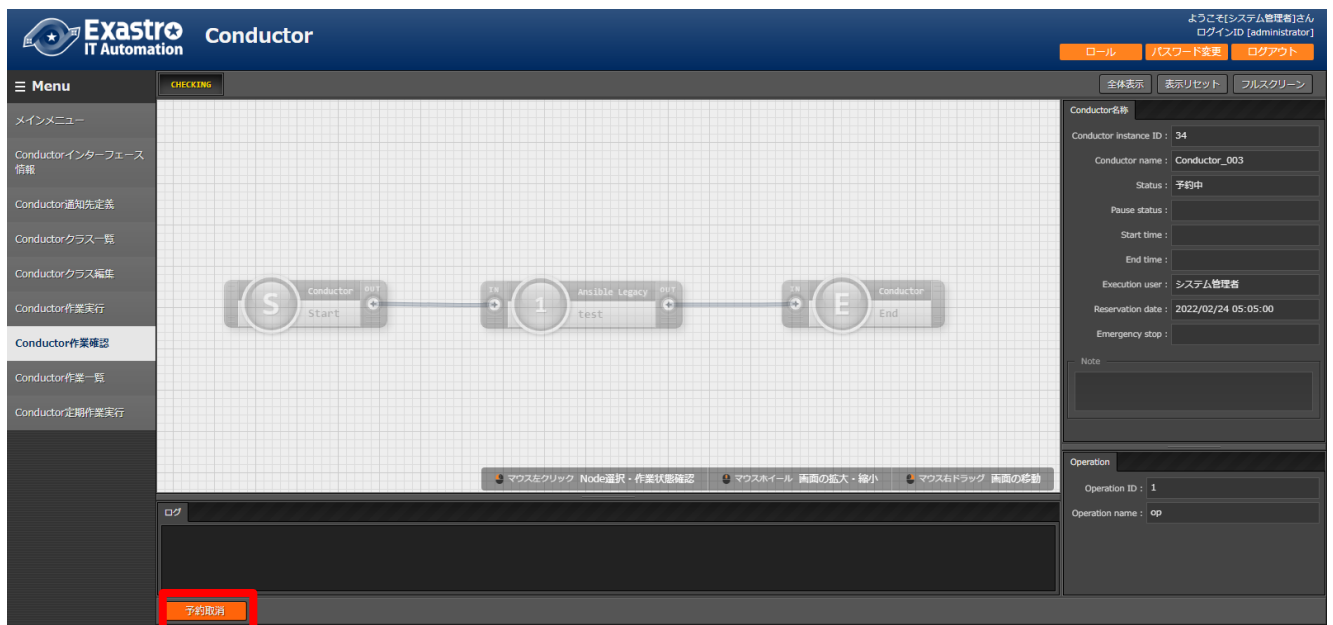


図 4.1- 30 「Conductor 作業確認」メニューの「予約取消」ボタン

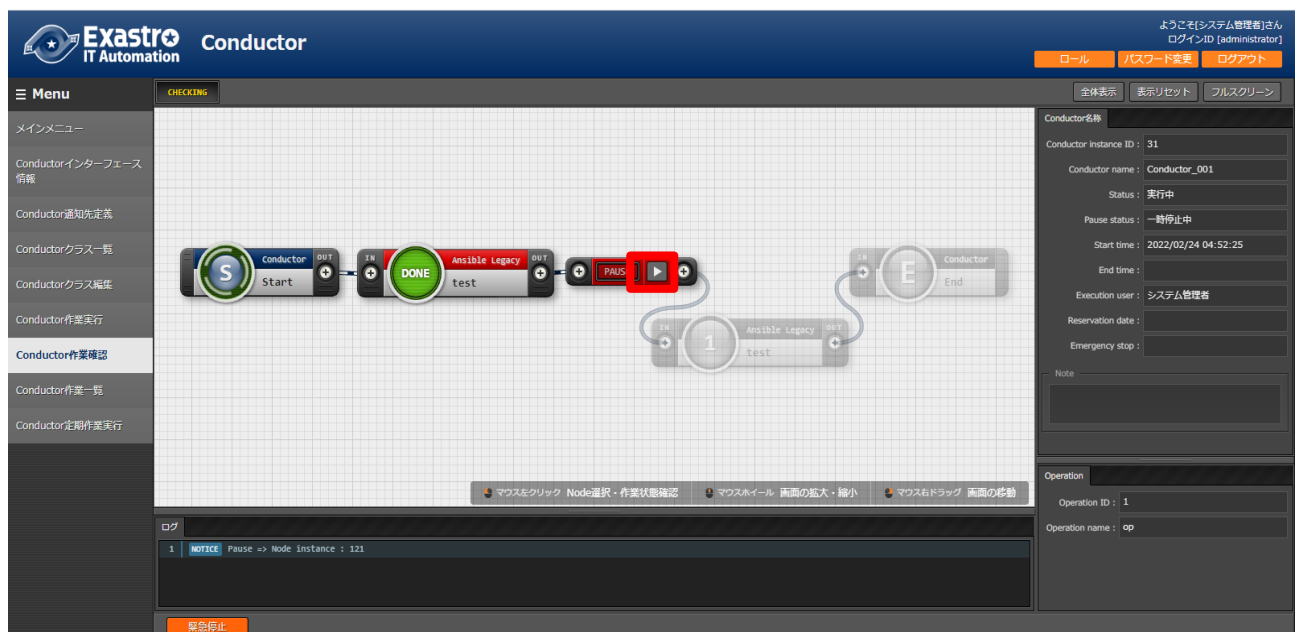


図 4.1- 31 「Conductor 作業確認」メニューの「停止解除」(Node:Conductor pause)

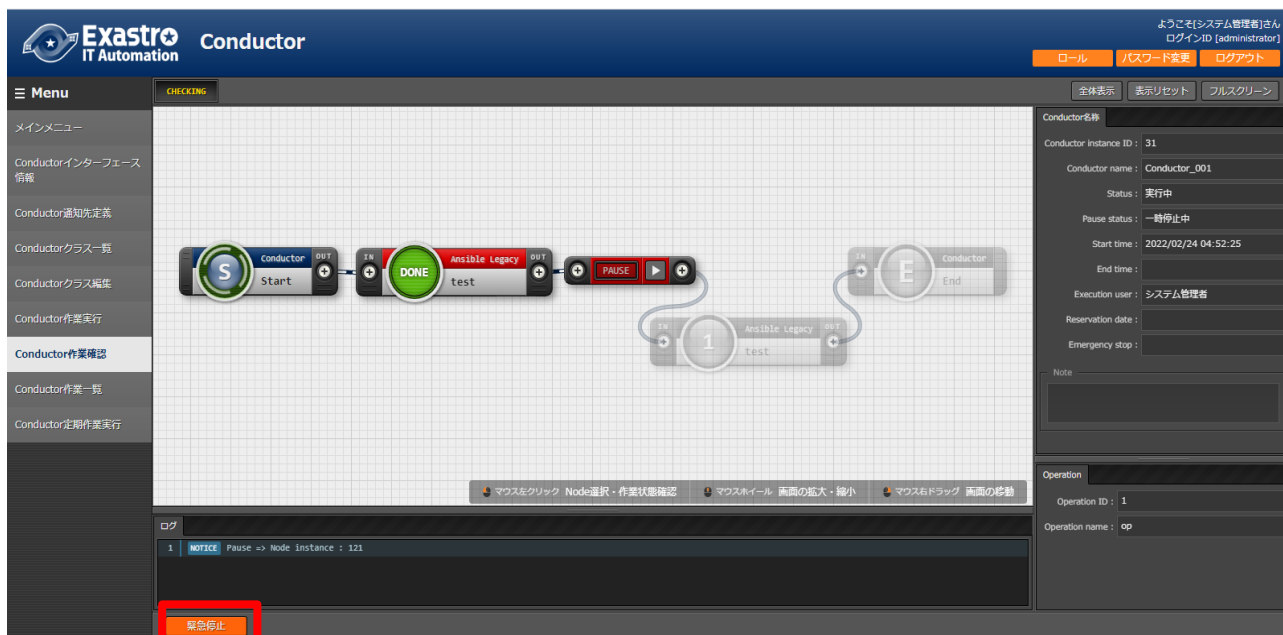


図 4.1- 32 「Conductor 作業確認」メニューの「緊急停止」ボタン

- ・ 「Conductor 作業確認」メニューの共通項目は以下のとおりです。

表 4.1- 22 「Conductor 作業確認」メニュー項目一覧

項目	説明	入力 必須	入力 形式	制約事項
停止解除	一時停止を解除します。	－	ボタン	－
緊急停止	Conductor の実行を中止します。	－	ボタン	－
予約取消	Conductor の実行予約を取り消します。	－	ボタン	予約日時が設定されていて、かつ未実行の場合に表示される。

4.1.7 Conductor 作業一覧

- (1) [Conductor 作業一覧]画面では、実行済みの Conductor の作業を管理します。
条件を指定し「フィルタ」ボタンをクリックすると、作業一覧テーブルを表示します。

作業表示欄の「詳細」ボタンをクリックすると、「4.1.6 Conductor 作業確認」画面に遷移します。

「投入データ式(zip)」の「download(.zip)」をクリックすると、実行された全ての Conductor 配下の Movement(*)の実行ファイルなどをまとめてダウンロードすることができます。

「結果データ式(zip)」の「download(.zip)」をクリックすると、実行された全ての Conductor 配下の Movement(*)の実行ログ、エラーログなどをまとめてダウンロードすることができます。

通知ログ欄から、通知の実行ログをダウンロードすることができます。

ログの出力例は、「5.1.2 通知ログ出力例」を参照してください。

(*)Conductor が階層構造になっている場合は、末端の Movement も対象になります。

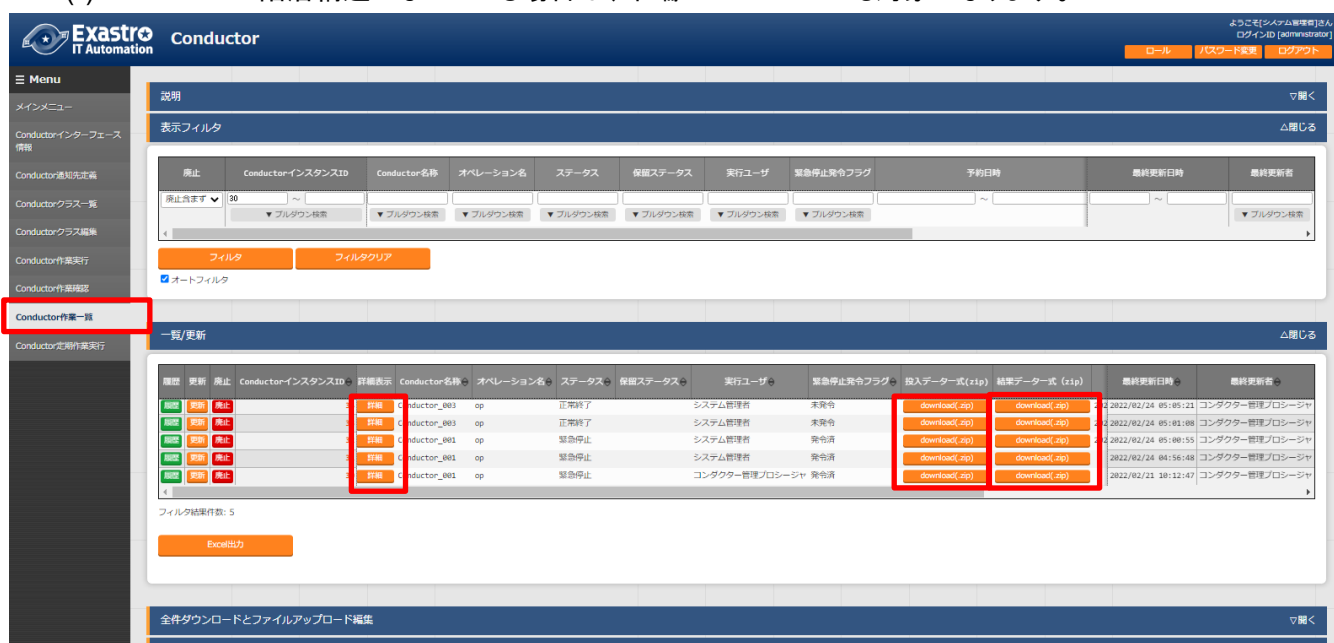


図 4.1- 33 「Conductor 作業一覧」メニュー

4.1.8 Conductor 定期作業実行

- (1) [定期作業実行]画面では、スケジュールに従って定期的に実行する Conductor 作業を管理します。
一覧の「作業一覧確認」ボタンをクリックすると、その定期作業で実行した対象をフィルタした状態で「4.1.7 Conductor 作業一覧」へ遷移できます。

Conductor 名称欄のリンクをクリックすると、対象の「4.1.4 Conductor クラス編集」へ遷移できます。

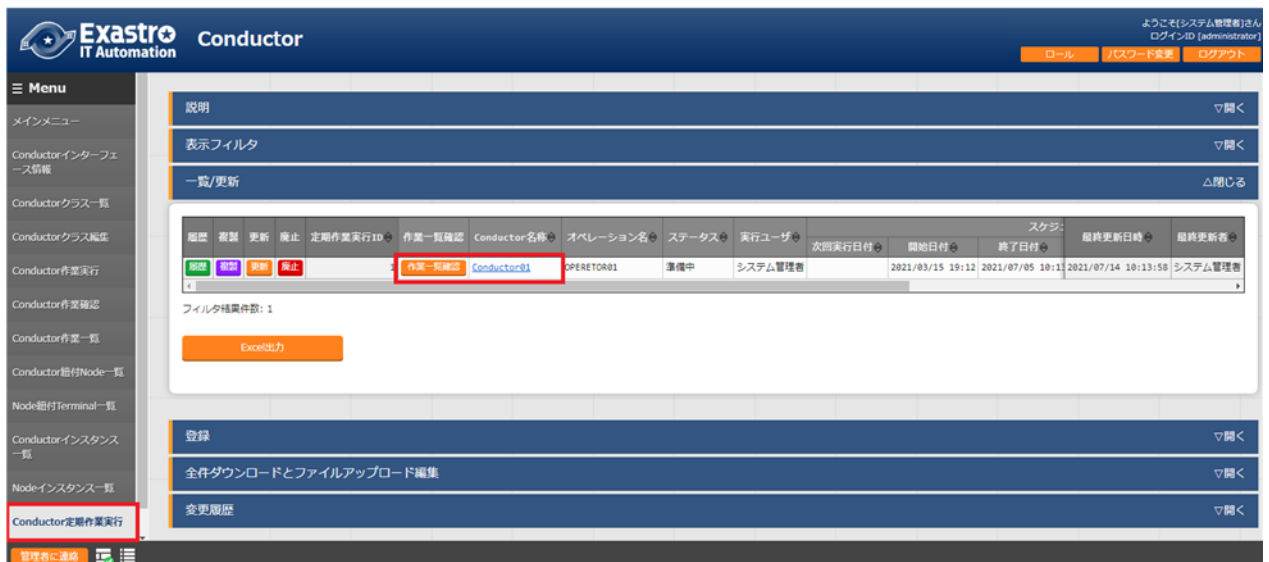


図 4.1- 34 「Conductor 定期作業実行」メニュー

- (2) 「登録」-「登録開始」ボタンより、定期作業実行の登録を行います。
スケジュールの詳細は「スケジュール設定」ボタンをクリックすることで表示されるウインドウからのみ設定できます。

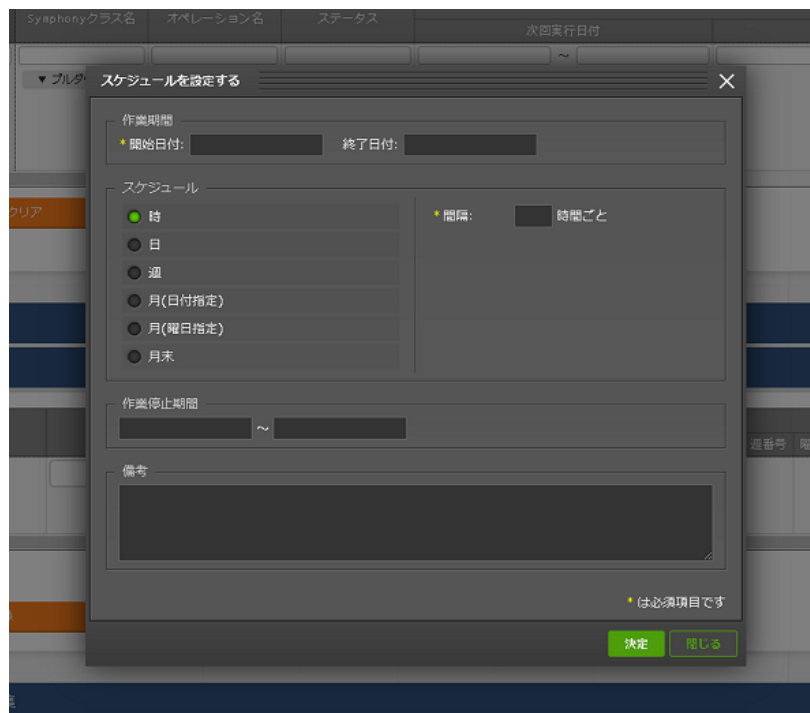


図 4.1- 35 「スケジュール設定」ウインドウ

(3) 登録画面の項目一覧は以下のとおりです

表 4.1- 23 登録画面項目一覧(定期作業実行)

項目		説明	入力 必須	入力形式	制約事項
Conductor クラス名		「4.1.3Conductor クラス一覧」で登録した Conductor がリストに表示されます。	○	リスト選択	-
オペレーション名		「基本コンソール-投入オペレーション一覧」で登録したオペレーションがリストに表示されます。	○	リスト選択	-
ステータス		下記「表 4.2 11 ステータス一覧(定期作業実行)」参照	-	自動入力	-
実行ユーザ		「登録」「更新」を実行したユーザが Conductor を実行するユーザとして登録されます。 定期作業実行が「4.1.7Conductor 作業一覧」へ作業登録をする際、「実行ユーザ」が引き継がれて登録されます。 「実行ユーザ」が指定した「Conductor クラス名」を実行できない場合 (Conductor で実行される Movement のアクセス権限を持っていない等)、ステータス「紐付けエラー」となります。	-	自動入力	
スケジュール設定		詳細なスケジュールを設定するウインドウをオープンするボタンです。	-	-	-
スケジュール	次回 実行日付	登録完了後にスケジュール設定に基づいて、次に作業を実行する日付が自動的に更新されます。	-	自動入力	-
	開始日付	定期作業実行を開始する日付を入力します。 「次回実行日付」は必ず「開始日付」以降の日付で更新されます。	○	手動入力	スケジュール設定からのみ入力可能
	終了日付	定期作業実行を終了する日付を入力します。 「次回実行日付」が「終了日付」を超えた場合にステータスが「完了」となります。	-	手動入力	スケジュール設定からのみ入力可能
	周期	定期的に行う周期を選択します。 「時」「日」「週」「月(日付指定)」「月(曜日指定)」「月末」があります。	○	ラジオ ボタン	スケジュール設定からのみ入力可能
	間隔	設定した周期に基づいた、定期的に行う間隔を入力します。	○	手動入力	スケジュール設定からのみ入力可能
	週番号	周期で「月(曜日指定)」を選択した場合に利用する、定期的に行う週番号を選択します。	※1	リスト選択	スケジュール設定からのみ入力可能
	曜日	周期で「曜日」「月(曜日指定)」を選択した場合に利用する、定期的に行う曜日を選択します。	※2	リスト選択	スケジュール設定からのみ入力可能
	日	周期で「月(日付指定)」を選択した場合に利用する、定期的に行う日にちを入力します。	※3	手動入力	スケジュール設定からのみ入力可能
	時間	定期的に行う時間を入力します。	※4	手動入力	スケジュール設定からのみ入力可能
作業停止 期間	開始	作業停止期間の開始日付を入力します。 開始日付以降から終了日付以前までの間、Conductor 作業登録が実行されないようにします。	※5	手動入力	スケジュール設定からのみ入力可能
	終了	作業停止期間の終了日付を入力します。 開始日付以降から終了日付以前までの間、Conductor 作業登録が実行されないようにします。	※5	手動入力	スケジュール設定からのみ入力可能
備考		自由記述欄です	-	手動入力	-

※1 週番号は周期に「月(曜日指定)」を選択した場合は必須です。

※2 曜日は周期に「月(曜日指定)」を選択した場合は必須です。

※3 日は周期に「月(日付指定)」を選択した場合は必須です。

※4 時間は周期に「日」「週」「月(日付指定)」「月(曜日指定)」「月末」を選択した場合は必須です。

※5 作業停止期間を設定する場合は「開始」と「終了」両方が入力されていることが必須です。

表 4.1- 24 ステータス一覧(定期作業実行)

ステータス名	説明
準備中	登録した直後のステータスです。 backyard が「次回実行日付」を自動更新するタイミングで「稼働中」になります。
稼働中	正常稼働中のステータスです。 「次回実行日付」の3分前に「4.1.7Conductor 作業一覧」への作業登録を実行し、再びスケジュール設定に基づいて「次回実行日付」が更新されます。
完了	「次回実行日付」が「終了日付」を超えた場合になるステータスです。以降は Conductor 作業登録を行いません。
不整合エラー	スケジュールの設定値に不正がある場合になるステータスです。
紐付けエラー	「4.1.7Conductor 作業一覧」への作業登録が失敗した場合になるステータスです。 ステータス「稼働中」同様、「4.1.7Conductor 作業一覧」への作業登録を実行し、再びスケジュール設定に基づいて「次回実行日付」を更新するという動作をします。その際再び作業登録が失敗した場合、ステータス「紐付けエラー」が継続されます。
想定外エラー	ステータス「不整合エラー」「紐付けエラー」以外の不具合が発生した場合になるステータスです。
Conductor 廃止	登録した Conductor が廃止された場合になるステータスです。廃止された Conductor を復活させた場合はステータス「準備中」へと更新されます。
operation 廃止	登録した operation が廃止された場合になるステータスです。廃止された operation を復活させた場合はステータス「準備中」へと更新されます。

- (4) 定期作業実行に登録した直後はステータスが「準備中」となり、その後 backyard がスケジュール設定に基づいて「次回実行日付」を更新し、そのタイミングでステータスは「稼働中」となります。
ステータスが「稼働中」か「紐付けエラー」の作業は「次回実行日付」より「Symphony / Conductor インターバル時間設定」に登録されている時間前に「4.1.7Conductor 作業一覧」への作業登録を実行し、再びスケジュール設定に基づいて「次回実行日付」が更新されます。

「Symphony / Conductor インターバル時間設定」

管理コンソール>システム設定メニューより、「次回実行日付」の何分前に作業登録を実行するか設定可能です。(Exastro-ITA_利用手順マニュアル_管理コンソール参照)

※一時停止を設定している Conductor を定期作業実行に登録した場合、作業登録後に「4.1.6 Conductor 作業確認」より「保留解除」の投入を行わないと「4.1.7Conductor 作業一覧」にステータス「実行中」で残り続けます。

5 付録

5.1 Conductor 通知先定義

5.1.1 Conductor 通知先定義設定例

■ Teams 設定例

通知名称	通知サンプル
通知先(CURLOPT_URL)	通知先の Teams の Webhook URL を入力してください。
ヘッダー (CURLOPT_HTTPHEADER)	["Content-Type: application/json"]
メッセージ (CURLOPT_POSTFIELDS)	{"text": "通知名: _NOTICE_NAME_, Conductor 名称: _CONDUCTOR_NAME_, Conductor インスタンス ID: _CONDUCTOR_INSTANCE_ID_, オペレーション ID: _OPERATION_ID_, オペレーション名: _OPERATION_NAME_, ステータス ID: _STATUS_ID_, ステータス: _STATUS_NAME_, 実行ユーザー: _EXECUTION_USER_, 予約日時: _TIME_BOOK_, 開始日時: _TIME_START_, 終了日時: _TIME_END_, 緊急停止フラグ: _ABORT_FLAG_, 作業 URL: _JUMP_URL_, "}
PROXY / URL (CURLOPT_PROXY)	http://proxy.co.jp
PROXY / PORT (CURLOPT_PROXYPORT)	8080
作業確認 URL(FQDN)	http://exastro-it-automation.local
その他	
開始日時	
終了日時	

■ Teams 通知表示例



■ Slack 設定例

通知名称	通知サンプル
通知先(CURLOPT_URL)	通知先の Slack の Webhook URL を入力してください。
ヘッダー (CURLOPT_HTTPHEADER)	["Content-Type: application/json"]
メッセージ (CURLOPT_POSTFIELDS)	{ "username": "ITAConductor 実行通知", "text": "通知名: _NOTICE_NAME_, ¥n Conductor 名称: _CONDUCTOR_NAME_, ¥n Conductor インスタンス ID: _CONDUCTOR_INSTANCE_ID_, ¥n オペレーション ID: _OPERATION_ID_, ¥n オペレーション名: _OPERATION_NAME_, ¥n ステータス ID: _STATUS_ID_, ¥n ステータス: _STATUS_NAME_, ¥n 実行ユーザー: _EXECUTION_USER_, ¥n 予約日時: _TIME_BOOK_, ¥n 開始日時: _TIME_START_, ¥n 終了日時: _TIME_END_, ¥n 緊急停止フラグ: _ABORT_FLAG_, ¥n 作業 URL: _JUMP_URL_ " }
PROXY / URL (CURLOPT_PROXY)	http://proxy.co.jp
PROXY / PORT (CURLOPT_PROXYPORT)	8080
作業確認 URL(FQDN)	http://exastro-it-automation.local
その他	
開始日時	
終了日時	

■ Slack 通知表示例



■設定サンプル(Proxy 設定、通知抑止設定、その他設定あり)

通知名称	通知サンプル
通知先(CURLOPT_URL)	https://sample.webhook.xxx.com/yyyyyyyy
ヘッダー (CURLOPT_HTTPHEADER)	["Content-Type: application/json"]
メッセージ (CURLOPT_POSTFIELDS)	{"text": "通知内容"}
PROXY / URL (CURLOPT_PROXY)	http://proxy.co.jp
PROXY / PORT (CURLOPT_PROXYPORT)	8080
作業確認 URL(FQDN)	http://exastro-it-automation.local
その他	{"CURLOPT_TIMEOUT":"10"}
開始日時	2020/01/01 00:00:00
終了日時	2020/01/01 00:00:00
備考	自由記述欄です

5.1.2 通知ログ出力例

通知ログの構造

```
YYYY-MM-dd HH:ii:ss 通知実行結果(<ID:通知名称>,<ID:ステータス名称>)
Array
(
    [RETURN_MSG] =>                                :通知実行時の返り値
    [OPTION] => Array                                :通知実行時のオプション
        (
            [CURLOPT_XXXXXXX] =>
                .....略.....
        )
    [RESULT] => Array                                :通知実行結果
        (
            [url] =>                                  :通知先 URL
            [http_code] =>                             :HTTP ステータスコード
                .....略.....
        )
)
```

例) 通知実行ログ(正常)

```
2021-11-05 15:10:22 通知実行結果(2:通知サンプル,5:正常終了)
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/yyyyyyyyy
            [CURLOPT_HTTPHEADER] => Array
                (
                    [0] => Content-Type: application/json
                )

            [CURLOPT_POSTFIELDS] => {"text": "通知名:通知サンプル 2, <br> Conductor 名称:
NULL, <br> Conductor インスタンス ID:3, <br> オペレーション ID: 1, <br>オペレーション名:OP_NULL,
<br>ステータス ID: 5, <br>ステータス: 正常終了, <br>実行ユーザー: システム管理者, <br> 予約日
時:, <br>開始日時: 2021/11/05 15:10:08, <br>終了日時: 2021/11/05 15:10:18, <br>緊急停止フラグ:
未発令, <br> 作業 URL:
http://exastro-it-automation.local/default/menu/01_browse.php?no=2100180005&conductor_instance_i
d=3, <br> "}
```

```

        [CURLOPT_PROXY] => https://sample.proxy.xxx.com
        [CURLOPT_PROXYPORT] => 8080
    )

[RESULT] => 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
)
)

```

例) 通知実行ログ(異常)

2021-11-05 15:10:20 通知実行結果(1:通知サンプル,5:正常終了)

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/yyyyyyyyy
    [CURLOPT_HTTPHEADER] => Array
        (
            [0] => Content-Type: application/json
        )

    [CURLOPT_POSTFIELDS] => {"text": "通知名:通知サンプル, <br> Conductor 名称:
NULL, <br> Conductor インスタンス ID:3, <br> オペレーション ID: 1, <br>オペレーション名:OP_NULL,
<br>ステータス ID: 5, <br>ステータス: 正常終了, <br>実行ユーザー: システム管理者, <br> 予約日
時:, <br>開始日時: 2021/11/05 15:10:08, <br>終了日時: 2021/11/05 15:10:18, <br>緊急停止フラグ:
未発令, <br> 作業 URL:
http://exastro-it-automation.local/default/menu/01_browse.php?no=2100180005&conductor_instance_i
d=3, <br> "}
    [CURLOPT_PROXY] =>
    [CURLOPT_PROXYPORT] =>
)

[RESSULT] => Array
(
    [url] => https://sample.webhook.xxx.com/yyyyyyyyy
    [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
)
)
```