



ITA\_利用手順マニュアル

Conductor

*－*第1.9版*－*

Copyright © NEC Corporation 2020. All rights reserved.

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](#_Toc90295133)

[1 ITA Conductorの概要 5](#_Toc90295134)

[2 ITA Conductorのメニュー、画面構成 6](#_Toc90295135)

[2.1 ITA Conductorメニュー 一覧 6](#_Toc90295137)

[3 ITA Conductor利用手順 7](#_Toc90295138)

[3.1 作業フロー 7](#_Toc90295139)

[4 機能・操作方法説明 9](#_Toc90295140)

[4.1 ITA Conductor 9](#_Toc90295141)

[4.1.1 Conductorインターフェース情報 9](#_Toc90295142)

[4.1.2 Conductor通知先定義 10](#_Toc90295143)

[4.1.3 Conductorクラス一覧 12](#_Toc90295144)

[4.1.4 Conductorクラス編集 13](#_Toc90295145)

[4.1.5 Conductor作業実行 29](#_Toc90295146)

[4.1.6 Conductor作業確認 32](#_Toc90295147)

[4.1.7 Conductor作業一覧 35](#_Toc90295148)

[4.1.8 Conductor定期作業実行 35](#_Toc90295149)

[5 付録 39](#_Toc90295150)

[5.1 Conductor通知先定義 39](#_Toc90295151)

[5.1.1 Conductor通知先定義設定例 39](#_Toc90295152)

[5.1.2 通知ログ出力例 42](#_Toc90295153)

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。

# 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となる以下の機能を提供します。

・ ジョブフローの作成、管理、実行

ITAの操作手順におけるConductorの位置づけは、「ファーストステップガイド」を参照してください。

# 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



## 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実行するConductor作業を管理します。 |

# ITA Conductor利用手順

## 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Movemetについては、共有の対象外となります。）

**①機器情報を登録/確認する**

**②オペレーションを登録/確認する**

**⑥Conductorを登録する**

**⑦Conductorを確認する**

**⑧Conductorを実行する**

**⑨Conductorの実行結果を確認する**

**⑩Conductorの実行履歴を確認する。**

**④Movementを確認する**

**③ITAの各DriverからMovementを**

**登録する**

**⑤Conductorのインターフェース情報を登録する**

基本コンソール

Conductor

**図 3.1-1 作業フロー**

# 機能・操作方法説明

## ITA Conductor

### 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。

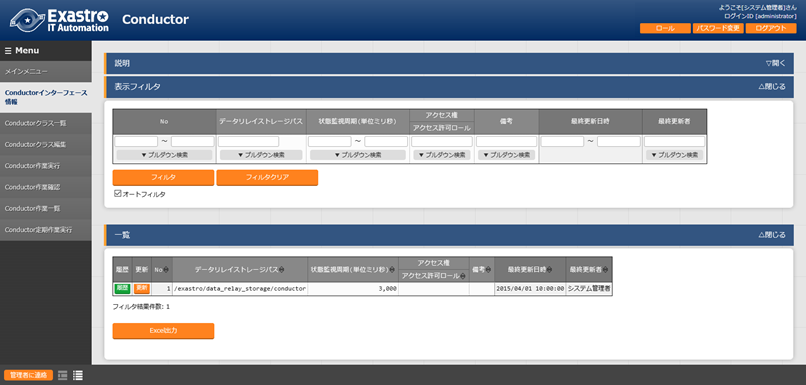


図 4.1- 1「Conductorインターフェース情報」メニュー

1. 「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 | - |

### 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。

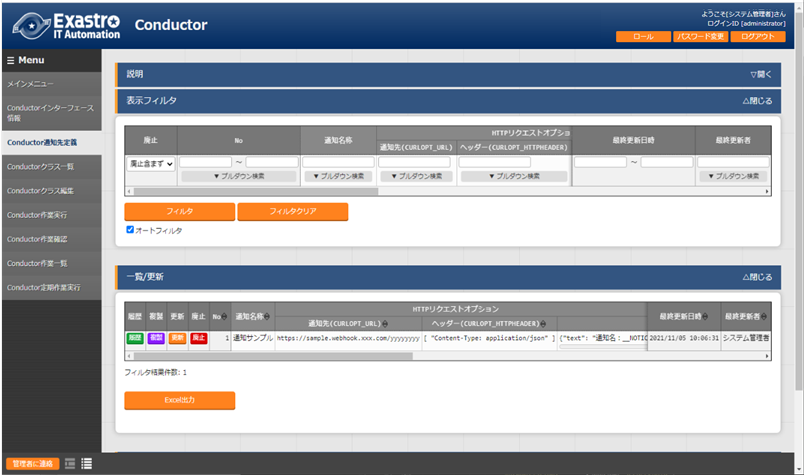


図 4.1- 2「Conductor通知先定義」メニュー

1. 「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。

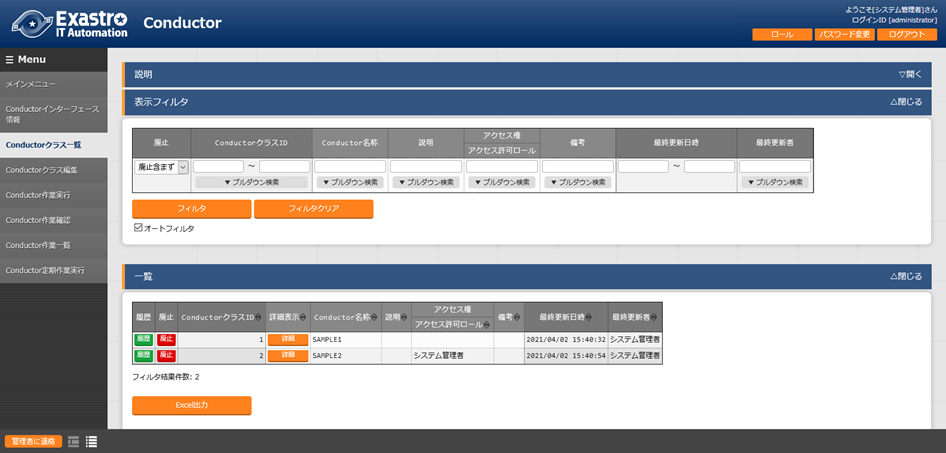
<作業確認URL(FQDN)>/default/menu/01\_browse.php?no=2100180005&conductor\_instance\_id=X

|  |
| --- |
| 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 |

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。



admin@nec.amc.jp

~

~

~

~

:set bin noeol

図 4.1- 3「Conductorクラス一覧」メニュー

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、ジョブ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。

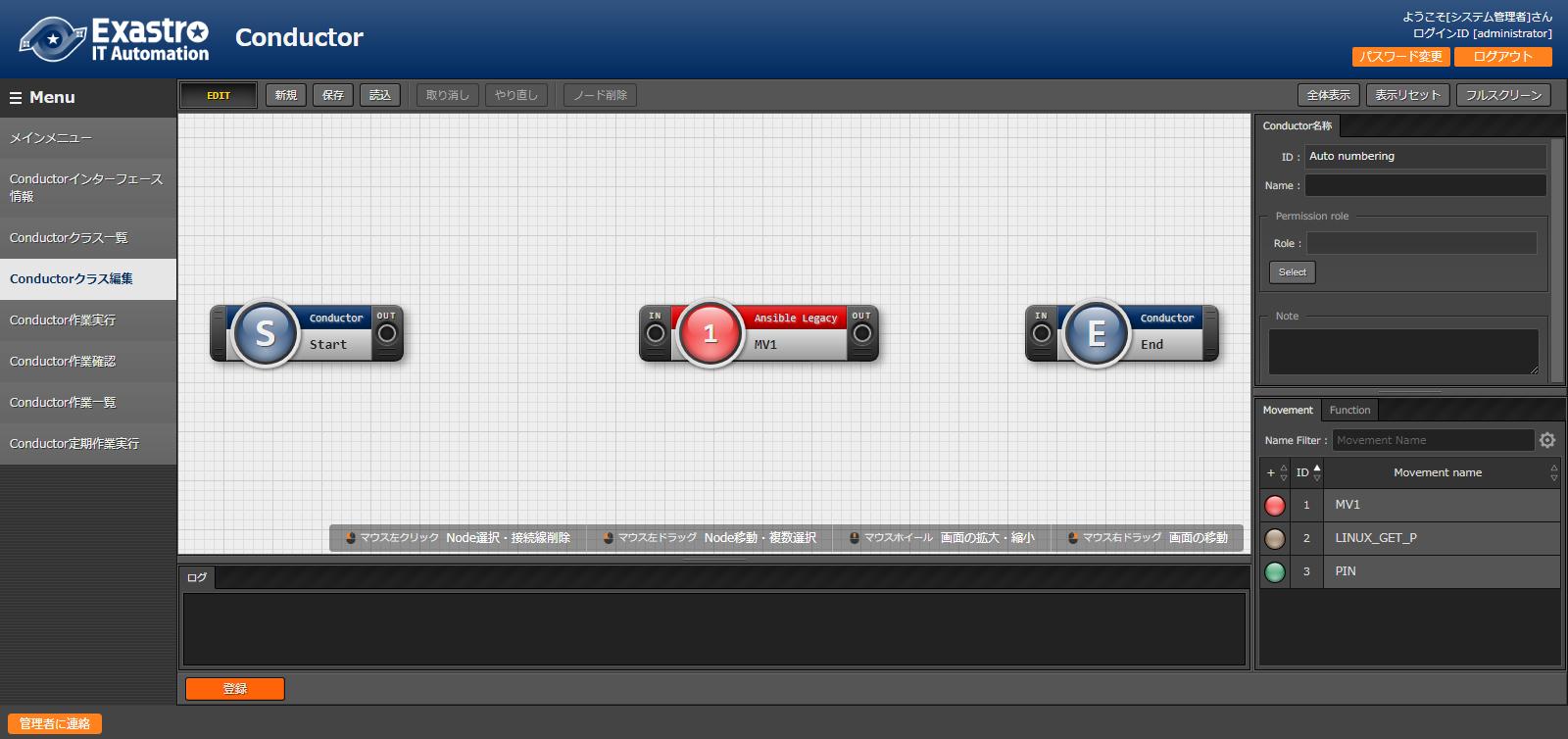
表 4.1- 4 モード一覧

|  |  |
| --- | --- |
| **モード** | **説明** |
| EDIT | * 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 | * 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。

**B）詳細情報**

****

**ドラッグ＆ドロップ**

**A）Node一覧**

**Node**

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

1. 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 | ジョブ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。 |

* Node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。

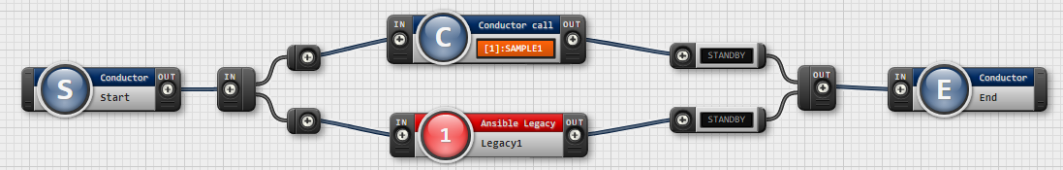


図 4.1- 5 Node制約事項（正常例：Parallel branch）

* + 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でマージする事はできません。

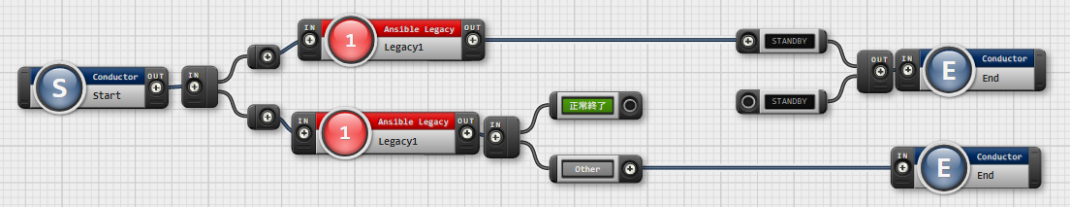


図 4.1- 7 Node制約事項（NG例：Conditional branch）

* + 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。

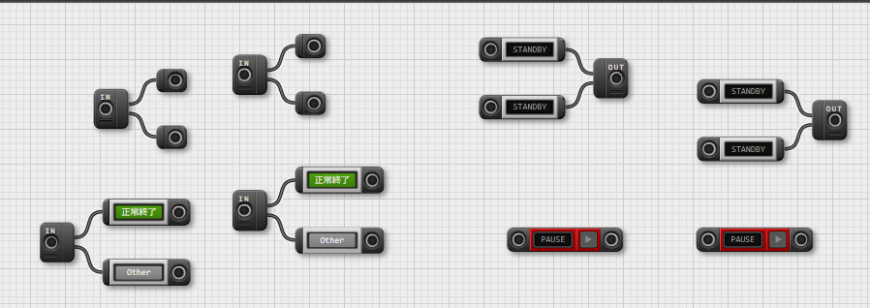


図 4.1- 8 Node制約事項（NG例：連続使用）

* + 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

1. 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。

1. 「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。  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。 | - | 選択 | ※ |
| 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。

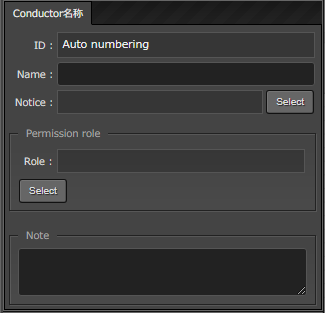


図 4.1- 10 「Conductor」名称タブ



図 4.1- 11 「Notice」のポップアップ

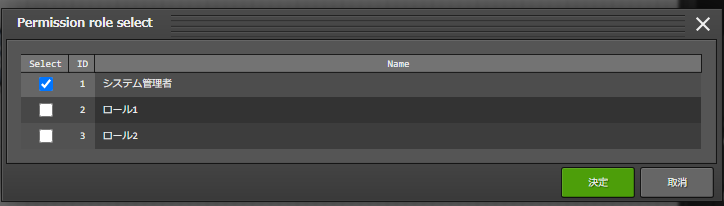


図 4.1- 12 「Permission role」のポップアップ

1. 「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 | - |

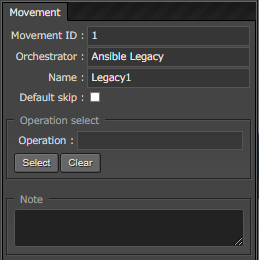


図 4.1- 13「Movement」タブ

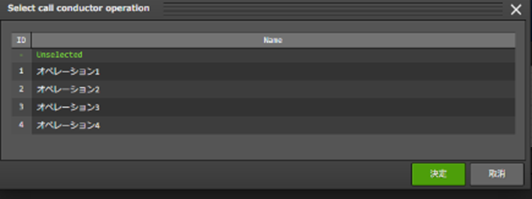


図 4.1- 14 「Operation select」のポップアップ

1. 「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 | - |

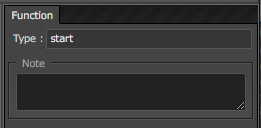


図 4.1- 15 「Functionタブ」

1. 「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を選択できます。  選択したConductorクラス名称が表示されます。 | 〇 | 選択 | - |
| 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 | - |

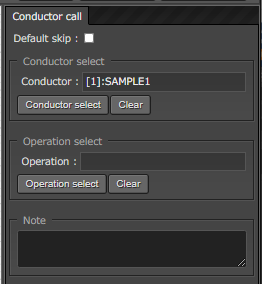


図 4.1- 16 「Conductor call」タブ

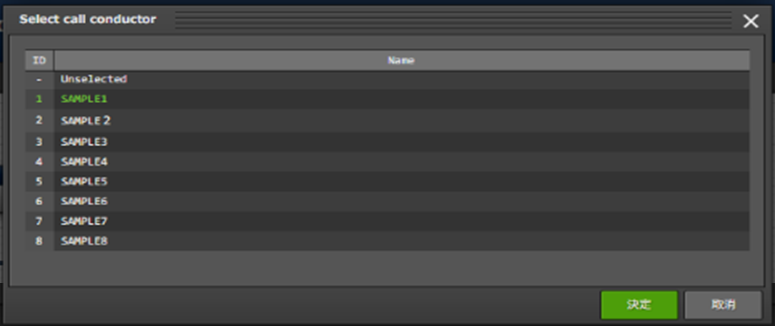


図 4.1- 17「Conductor select」のポップアップ

1. 「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。 | 〇 | 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 | - |

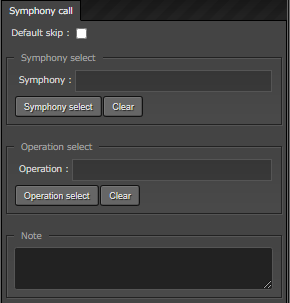


図 4.1- 18 「Symphony call」タブ

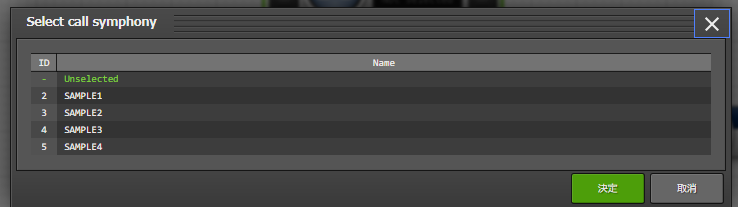


図 4.1- 19 「Symphony select」のポップアップ

1. 「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 | - |

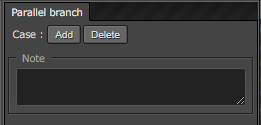


図 4.1- 20 「Parallel branch」タブ

1. 「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。   |  |  | | --- | --- | | **Case1** | 正常終了 | | **Other** | 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 | | - | 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 | - |

※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。

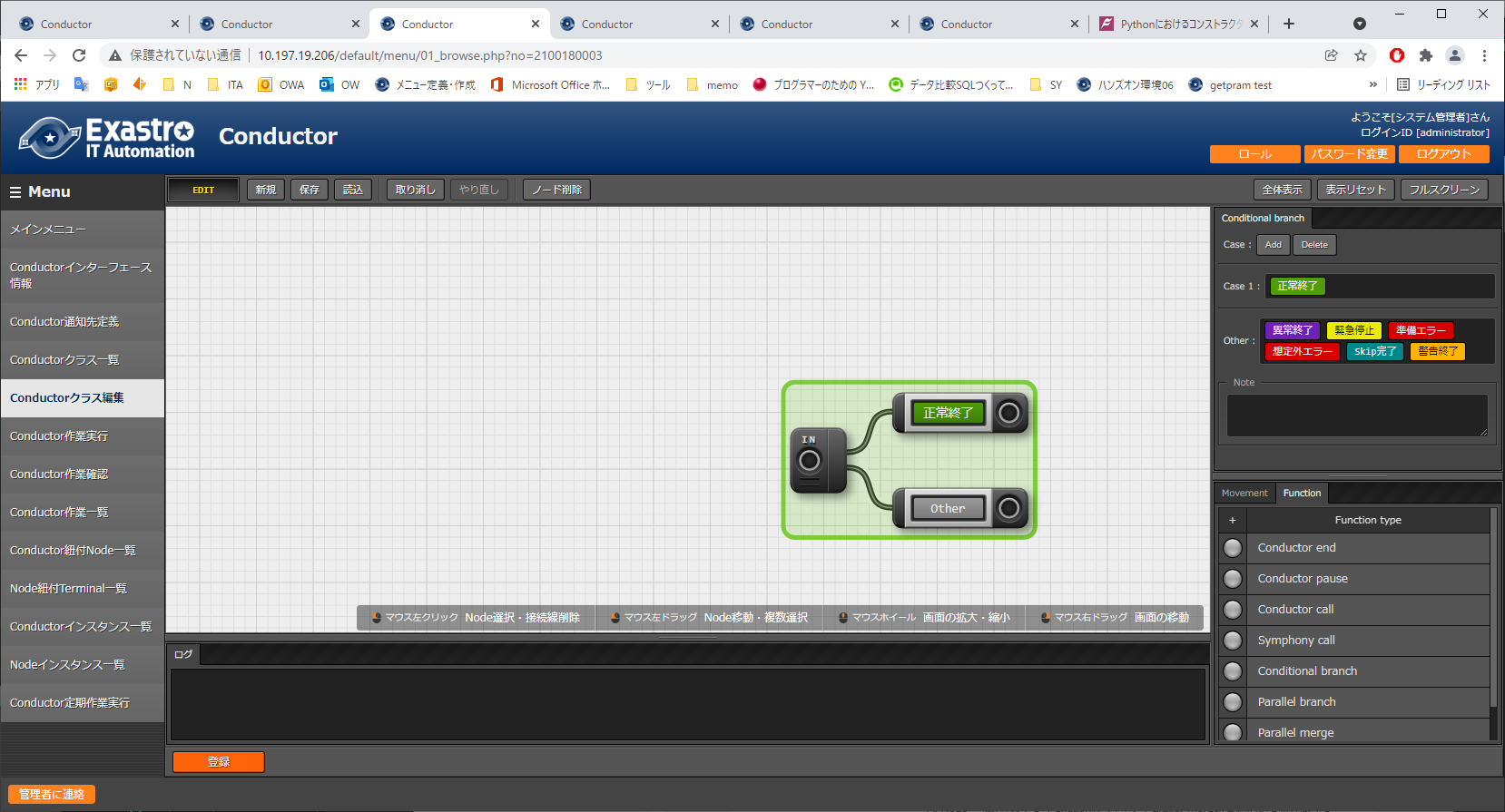


図 4.1- 21 「Conditional branch」タブ

1. 「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 | - |

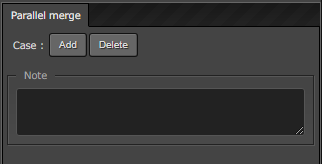


図 4.1- 22 「Merge」タブ

1. 「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。  - 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

1. 「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 | - |

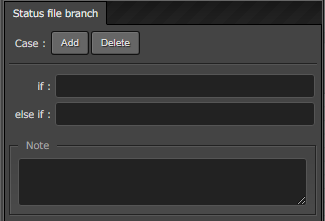


図 4.1- 24 「Status file branch」タブ

※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。

1. 「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 | - |

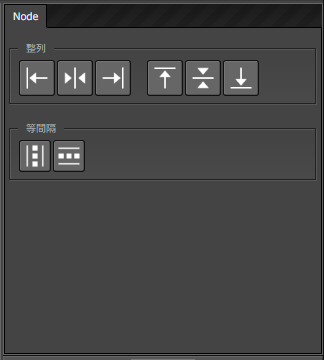


図 4.1- 25 「Node」タブ

* 「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。 | 〇 | - | - |  |
| 保存 | 描画されている状態を、ファイル形式で保存します。 | 〇 | - | - |  |
| 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。 | 〇 | - | - |  |
| 取り消し | 直前の処理を取り消します。 | 〇 | - | 〇 |  |
| やり直し | 直前の取り消しをやり直します。 | 〇 | - | 〇 |  |
| ノード削除 | 選択しているNodeを削除します。 | 〇 | - | 〇 |  |
| 登録 | 登録を実施します。 | 〇 | - | - |  |
| 編集 | EDITモードへ変更し、Conductorの編集を行います。 | - | 〇 | 〇 |  |
| 流用新規 | 登録済のConductorを流用して、新規作成が行えます。 | - | 〇 | 〇 |  |
| 更新 | 編集内容を更新します。 | - | - | 〇 |  |
| 再読込 | 変更をキャンセルし、変更前の状態へ戻します。 | - | - | 〇 |  |
| キャンセル | 変更をキャンセルし、VIEWモードへ変更します。 |  | - | 〇 |  |

1. 「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。

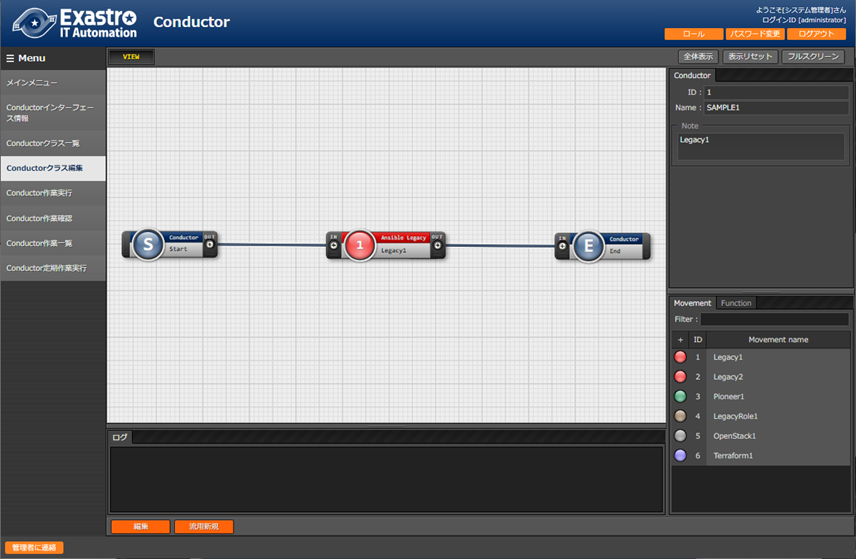


図 4.1- 26 「Conductorクラス編集」メニュー（「VIEW」モード）

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。 |

1. 「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。

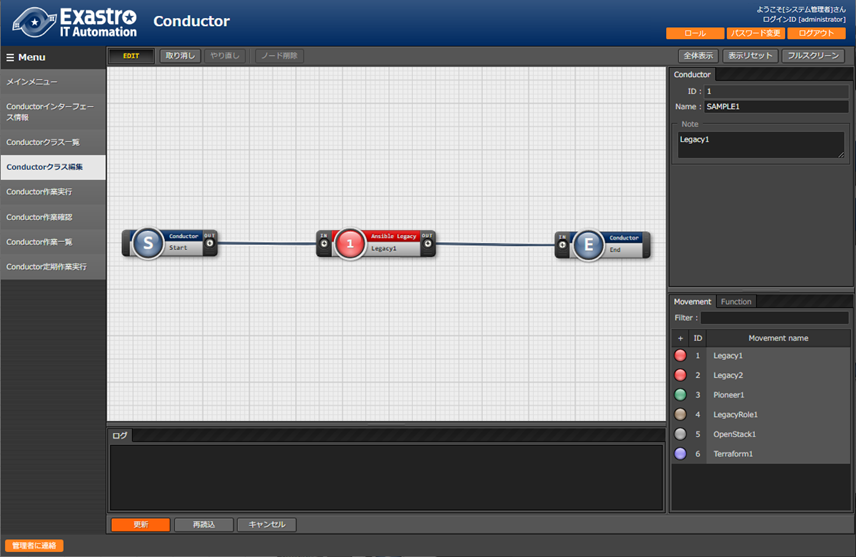


図 4.1- 27 「Conductorクラス編集」メニュー（「EDIT」モード）

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。 |
| 「再読込」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。 |

admin@nec.amc.jp

~

~

~

~

:set bin noeol

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。
* 「スケジューリング」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。

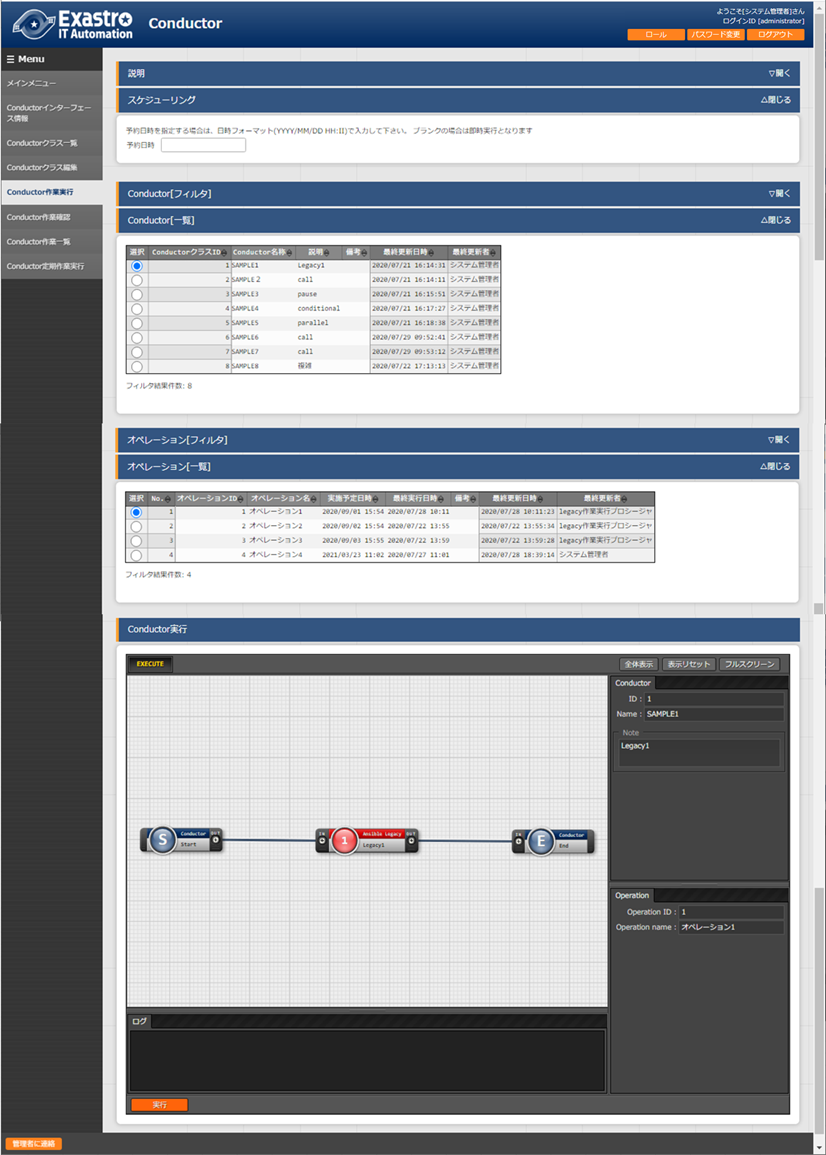


図 4.1- 28 「Conductor作業実行」メニュー

* 「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](https://ky-labo/ansible_driver/legacy/ns/mainmenu/01_browse.php)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

「Cnductor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。

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。

admin@nec.amc.jp

~

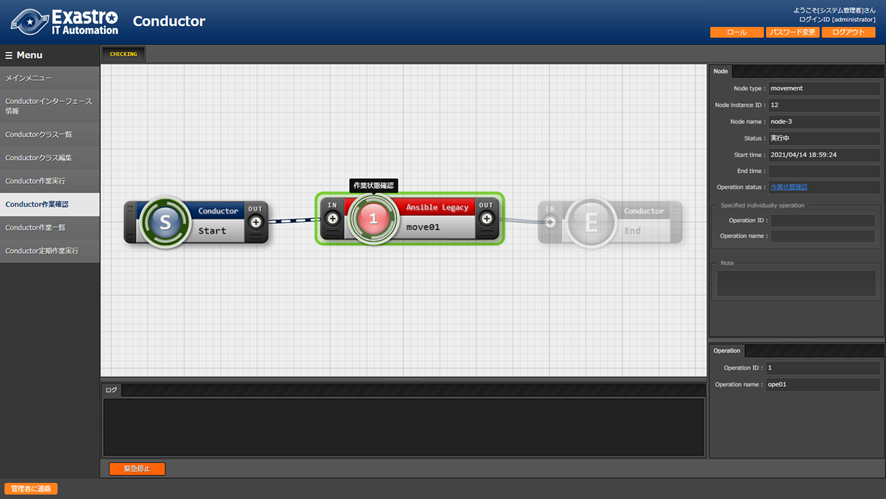
~

~

~

:set bin noeol

* 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。



**ログ**

**実行状況サークル**

**Node**

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

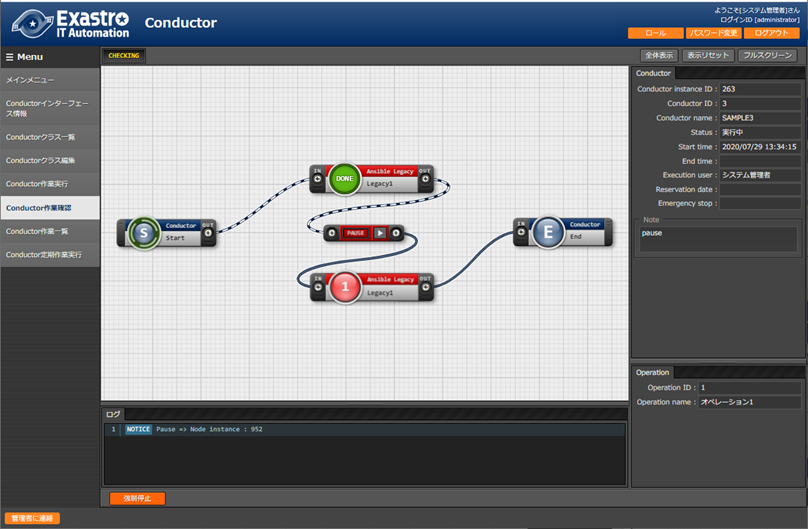
****

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

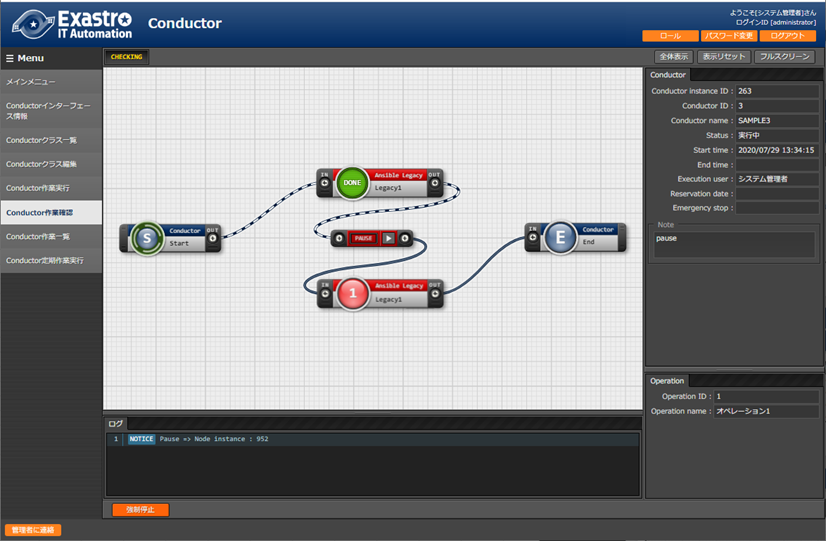
****

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。 |

### 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。

「投入データ一式(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。

「結果データ一式(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。

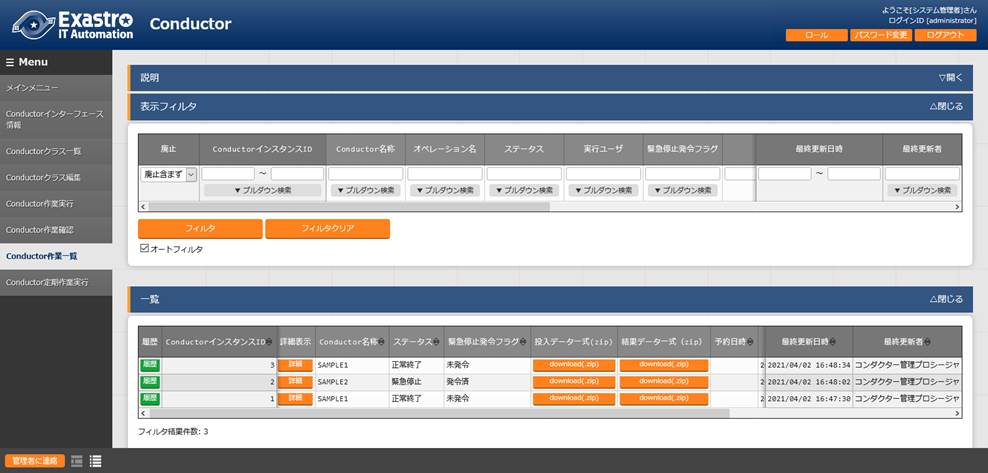


図 4.1- 33 「Conductor作業一覧」メニュー

### 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「4.1.7Conductor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。

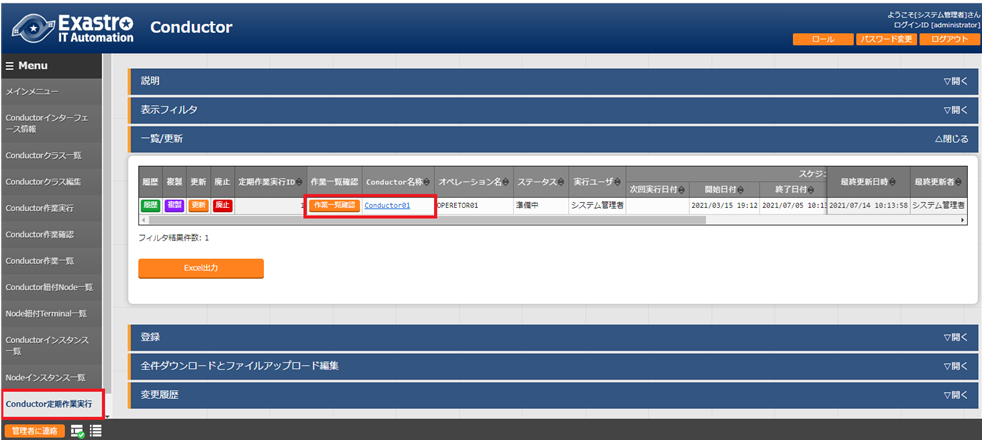


図 4.1- 34 「Conductor定期作業実行」メニュー

1. 「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。

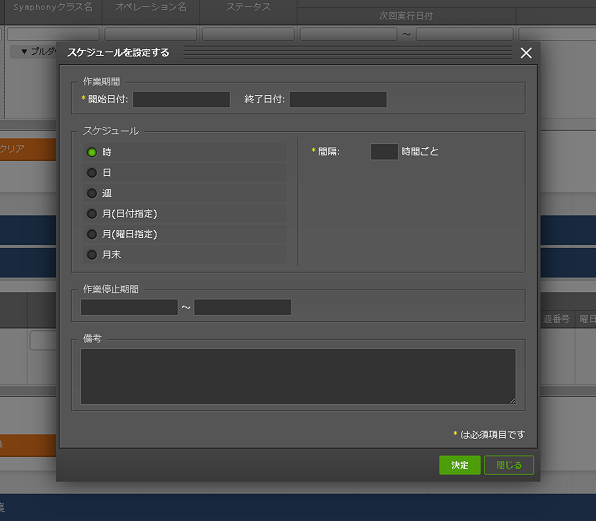


図 4.1- 35 「スケジュール設定」ウインドウ

1. 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 | 定期的に実行する周期を選択します。  「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、定期的に実行する間隔を入力します。 | ○ | 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、  定期的に実行する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、定期的に実行する曜日を選択します。 | ※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、定期的に実行する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 | 定期的に実行する時間を入力します。 | ※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。 |

1. 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。

# 付録

## Conductor通知先定義

### 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\_\_, <br> Conductor名称: \_\_CONDUCTOR\_NAME\_\_, <br> ConductorインスタンスID:\_\_CONDUCTOR\_INSTANCE\_ID\_\_, <br> オペレーションID: \_\_OPERATION\_ID\_\_, <br>オペレーション名:\_\_OPERATION\_NAME\_\_, <br>ステータスID: \_\_STATUS\_ID\_\_, <br>ステータス: \_\_STATUS\_NAME\_\_, <br>実行ユーザー: \_\_EXECUTION\_USER\_\_, <br> 予約日時: \_\_TIME\_BOOK\_\_, <br>開始日時: \_\_TIME\_START\_\_, <br>終了日時: \_\_TIME\_END\_\_, <br>緊急停止フラグ: \_\_ABORT\_FLAG\_\_, <br> 作業URL: \_\_JUMP\_URL\_\_, <br> "} |
| 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 |

### 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\_XXXXXXXX] =>  ・・・・・・・・略・・・・・・・・  )  [RESSULT] => 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/yyyyyyyy  [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\_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  )  ) |

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/yyyyyyyy  [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\_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  )  ) |