

ITA\_利用手順マニュアル

Conductor

*－*第1.5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。

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](#_Toc48124396)

[1 ITA Conductorの概要 5](#_Toc48124397)

[2 ITA Conductorのメニュー、画面構成 6](#_Toc48124398)

[2.1 ITA Conductorメニュー 一覧 6](#_Toc48124400)

[3 ITA Conductor利用手順 7](#_Toc48124401)

[3.1 作業フロー 7](#_Toc48124402)

[4 機能・操作方法説明 8](#_Toc48124403)

[4.1 ITA Conductor 8](#_Toc48124404)

[4.1.1 Conductorインターフェース情報 8](#_Toc48124405)

[4.1.2 Conductorクラス一覧 9](#_Toc48124406)

[4.1.3 Conductorクラス編集 10](#_Toc48124407)

[4.1.4 Conductor作業実行 19](#_Toc48124408)

[4.1.5 Conductor作業確認 22](#_Toc48124409)

[4.1.6 Conductor作業一覧 25](#_Toc48124410)

[4.1.7 Conductor定期作業実行 25](#_Toc48124411)

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。

・ ワークフローの作成、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・画面** | **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。

**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

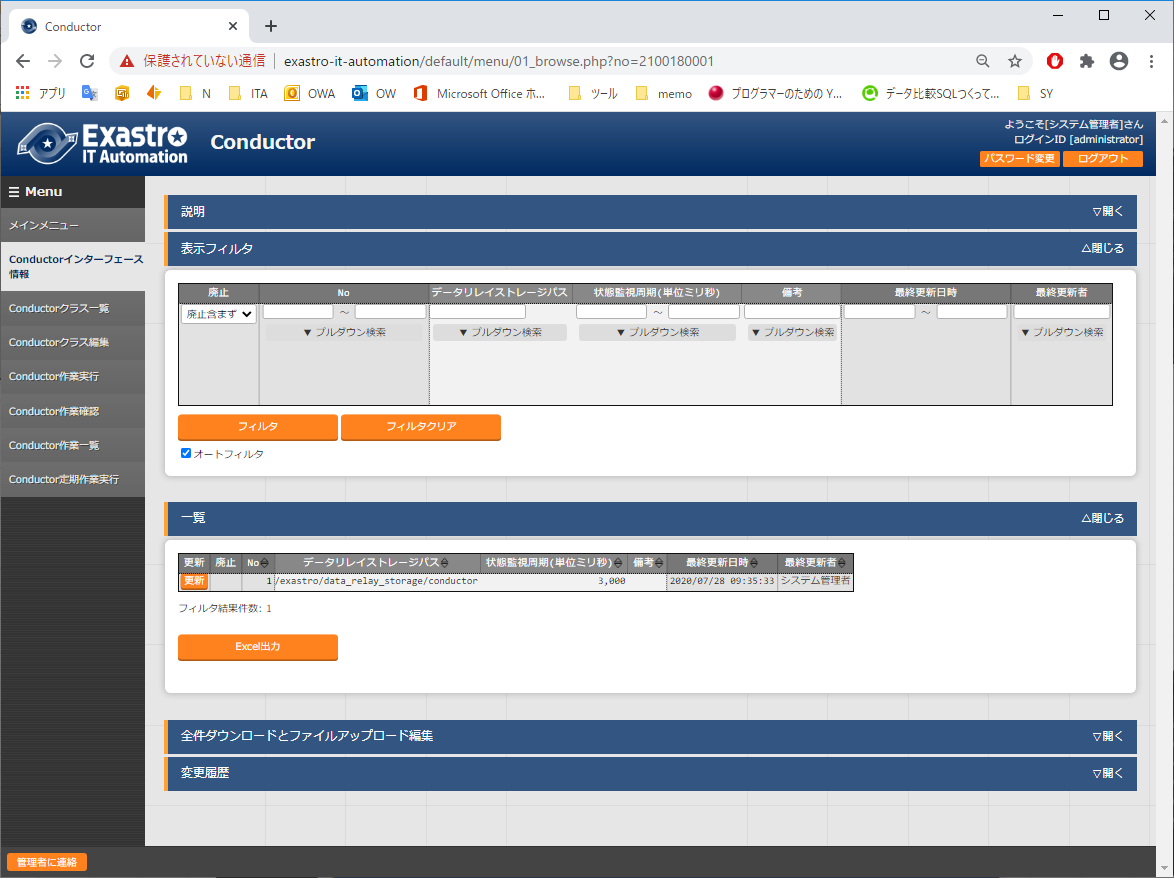
**図 3.1-1 作業フロー**

# 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]画面では、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]画面のリフレッシュ間隔を設定します。



**図 4.1‑1サブメニュー画面（Conductorインターフェース情報）**

1. 画面の項目一覧は以下のとおりです。

**表 4.1-1　登録画面項目一覧（Conductorインターフェース情報）**

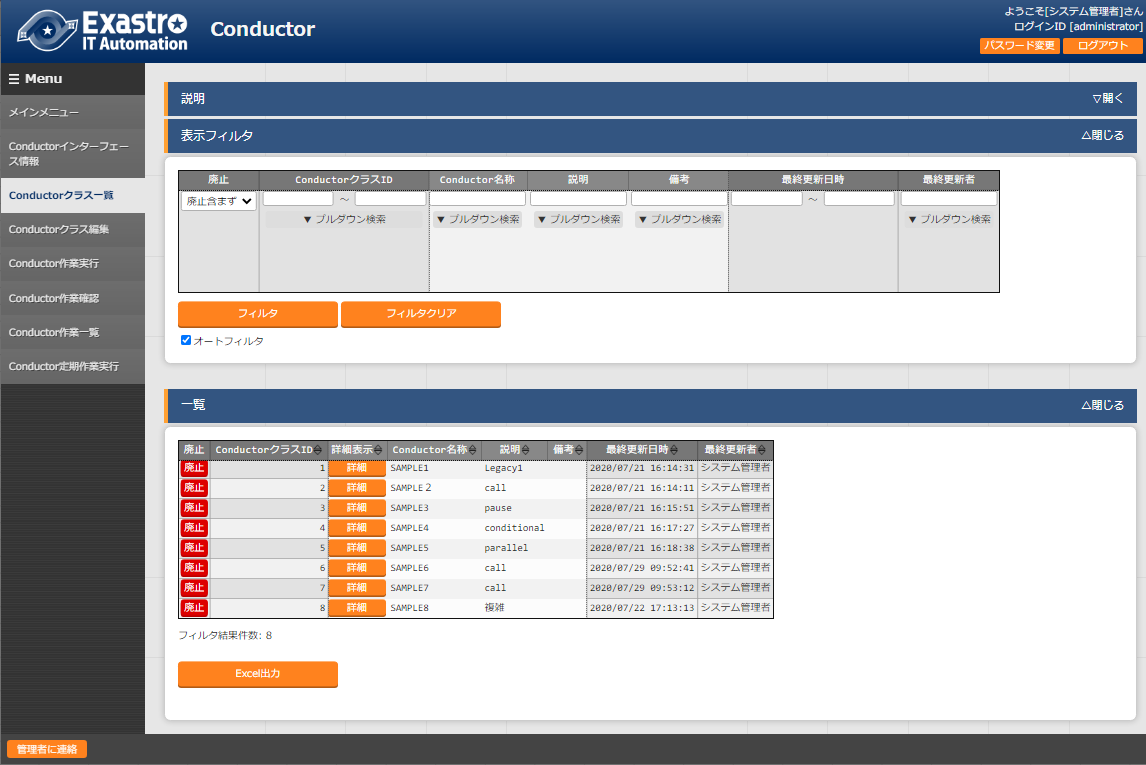
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | **説明** | **入力**  **必須** | **入力形式** | **制約事項** |
| データリレイストレージパス | Conductor実行時、各Movementで共有するディレクトリを、ITAサーバから見たディレクトリパスで入力します。  各Driverから見たパスは、各Driverの利用手順マニュアルのインターフェース情報を参照して下さい。  ディレクトリを共有することが出来るDriverは以下です。  ・Ansible  ・Ansible-Tower | ○ | 手動入力 | 最大長128バイト |
| 状態監視周期（単位ミリ秒） | 「4.1.4 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クラス一覧

1. [Conductorクラス一覧]画面では、登録済みのConductorクラスを参照/廃止できます。

一覧の「詳細」ボタンをクリックすると、そのConductorクラスの編集画面

[4.1.3 Conductorクラス編集]へ遷移できます。



admin@nec.amc.jp

~

~

~

~

:set bin noeol

**図 4.1‑2サブメニュー画面（Conductorクラス一覧）**

Conductorクラス編集

Conductorクラス編集画面、Conductorを構成するパーツ（以下、Node）、について、以下になります。

1. 各Nodeについて

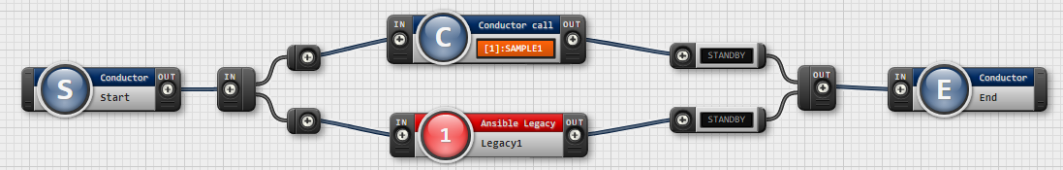
**表 4.1-2　Node一覧**

|  |  |  |
| --- | --- | --- |
| **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。 |
|  | Conditional branch | 接続された、「Movement」、「Conductor 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 |
|  | Parallel branch | 並列して、「Movement」、「Conductor 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。 |
|  | Movement各種 | Movementを実行します。 |

制約事項

・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。

・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。

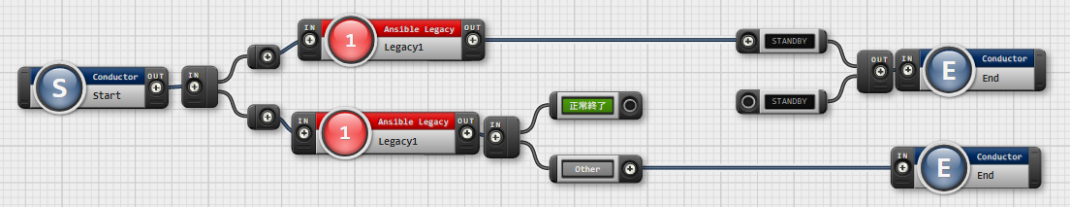


**図 4.1‑3 Node制約事項（正常例：Parallel branch）**



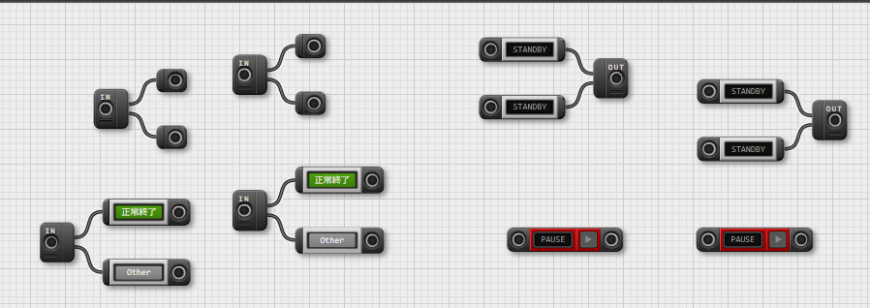
**図 4.1‑4 Node制約事項（NG例：Parallel branch）**

・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。



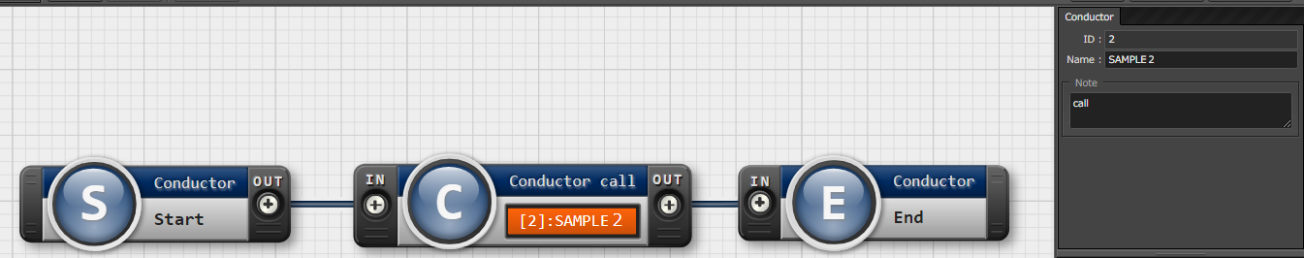
**図 4.1‑5 Node制約事項（NG例：Conditional branch）**

・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 4.1‑6 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 4.1‑7 Node制約事項（NG例：Conductor call）**

1. [Conductor クラス編集]画面について

* Conductorクラス名称とワークフローのNodeを登録します。
* 画面上のモードについて、以下のモードがあります。

各モードで可能な操作について、「**表4.1-11　 Conductorクラス編集画面実行操作一覧**」を参照。

* **表 4.1-3** **Conductor クラス編集画面モード一覧**

|  |  |
| --- | --- |
| **モード** | **説明** |
| EDIT | ・Conductor クラスを編集できるモード  ・Conductor クラス編集画面のデフォルトのモード  ・EDITモードから登録/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 クラス一覧の詳細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 |

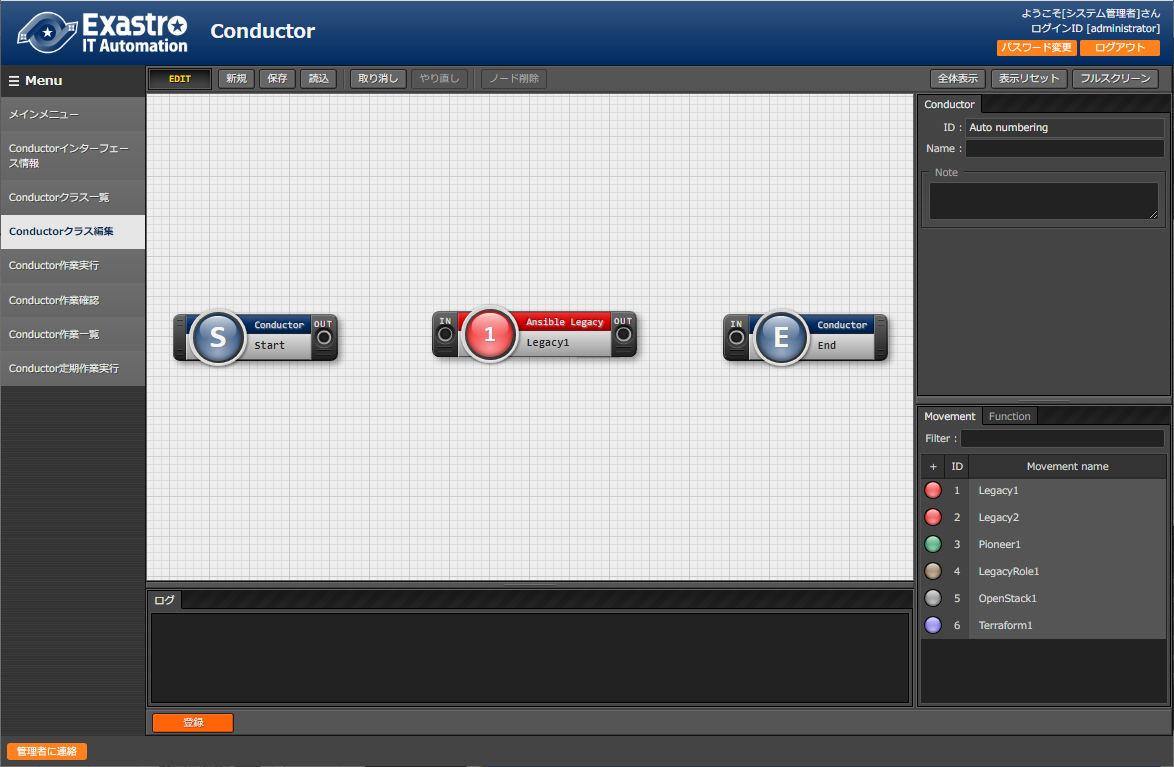
* 画面右側上部の領域には、選択しているNodeの詳細情報が表示されます。
* 画面右側下部の領域には、利用可能なNodeが表示されます。
  1. Movemetタブ
     + 登録済みのMovementのIDと名称の一覧
  2. Functionタブ
     + Conductor end
     + Conductor pause
     + Conductor call
     + Conditional branch
     + Parallel branch
     + Parallel merge

・　画面右下側のNode領域から、ドラッグ＆ドロップで各Nodeを設定できます。

* 各NodeのNote記入欄には、処理説明やコメントをメモしておけます。

この情報は、処理実行に影響はなく、Web上でのみ参照できるメモ欄です。

* Node設定後、「登録」ボタンをクリックして、Conductorクラスを登録します



**Node一覧**

**詳細情報**

編集領域

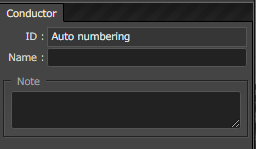
**ドラッグ＆ドロップ**

**図 4.1‑8サブメニュー画面（Conductorクラス編集：EDIT）**

* Conductorクラス編集画面の詳細、入力項目一覧は以下のとおりです。
  1. Conductor詳細、入力項目

**表4.1-4　 Conductorクラス編集画面項目一覧：（Conductor）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 | - |
| 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 | - |

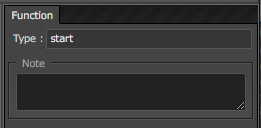


**図 4.1‑9詳細、入力項目（Conductor）**

* 1. Node（共通）詳細、入力項目

**表4.1-5　 Conductorクラス編集画面項目一覧：（Node共通）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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します | - | 手動入力 | - |

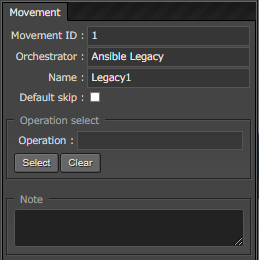


**図 4.1‑10詳細、入力項目（Node共通）**

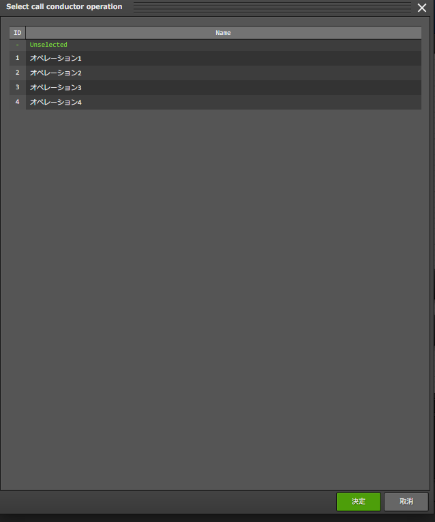
* 1. Movement詳細、入力項目

**表4.1-6　 Conductorクラス編集画面項目一覧：（Movement）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | **説明** | **入力**  **必須** | **入力形式** | **制約事項** |
| Movement ID | 選択したMovemenのIDが表示されます | - | 自動入力 | - |
| Orchestrator | 選択したMovemen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Movemen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作業実行画面にて、変更可能なパラメータ | - | 手動入力 |  |
| Operation | ・Selectボタンを押下すると、表示されるオペレーション一覧から選択します。  ・指定したオペレーション名が表示されます。 | - | 選択 | - |



**図 4.1‑11詳細、入力項目（Movement）**

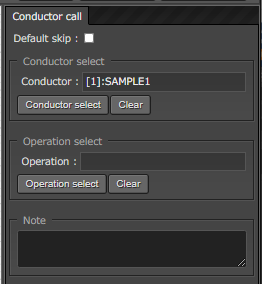


**図 4.1‑12詳細、入力項目（Operation select）**

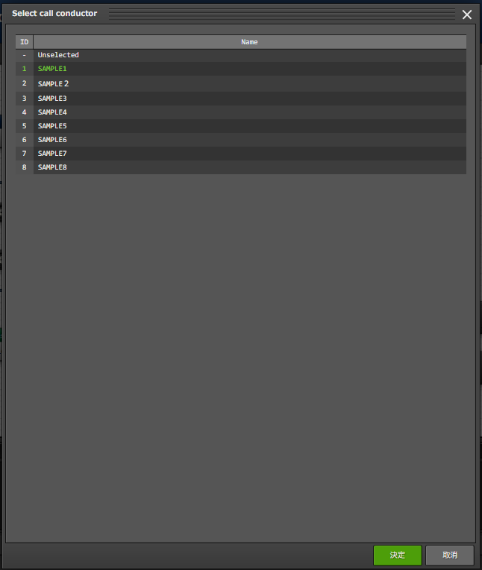
* 1. Conductor call詳細、入力項目

**表4.1-7　 Conductorクラス編集画面項目一覧：（Conductor call）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 | - | 手動入力 | - |
| Conductor | ・Selectボタンを押下すると、表示されるConductor一覧から選択します。  ・指定したConductorクラス名称が表示されます。 | 〇 | 選択 | - |
| Operation | ・Selectボタンを押下すると、表示されるオペレーション一覧から選択します。  ・指定したオペレーション名が表示されます。 | - | 選択 | - |



**図 4.1‑13詳細、入力項目（Conductor call）**

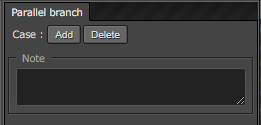


**図 4.1‑14詳細、入力項目（Conductor call select）**

* 1. Parallel branch詳細、入力項目

**表4.1-8　 Conductorクラス編集画面項目一覧：（Parallel branch）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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。  デフォルトは2つで以下を押下して、増減します。  ・Add  ・Delete | - | 選択 |  |



**図 4.1‑15詳細、入力項目（Parallel branch ）**

* 1. Conditional branch詳細、入力項目

**表4.1-9　 Conductorクラス編集画面項目一覧：（Conditional branch）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | **説明** | **入力**  **必須** | **入力形式** | **制約事項** |
| case  (1-6) | Movement、ConductorCall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、  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 | - | 選択 |  |

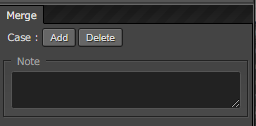


**図 4.1‑16詳細、入力項目（Conditional branch ）**

* 1. Conditional branch詳細、入力項目

**表4.1-10　 Conductorクラス編集画面項目一覧：（Parallel merge）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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。  デフォルトは2つで以下を押下して、増減します。  ・Add  ・Delte | - | 選択 |  |



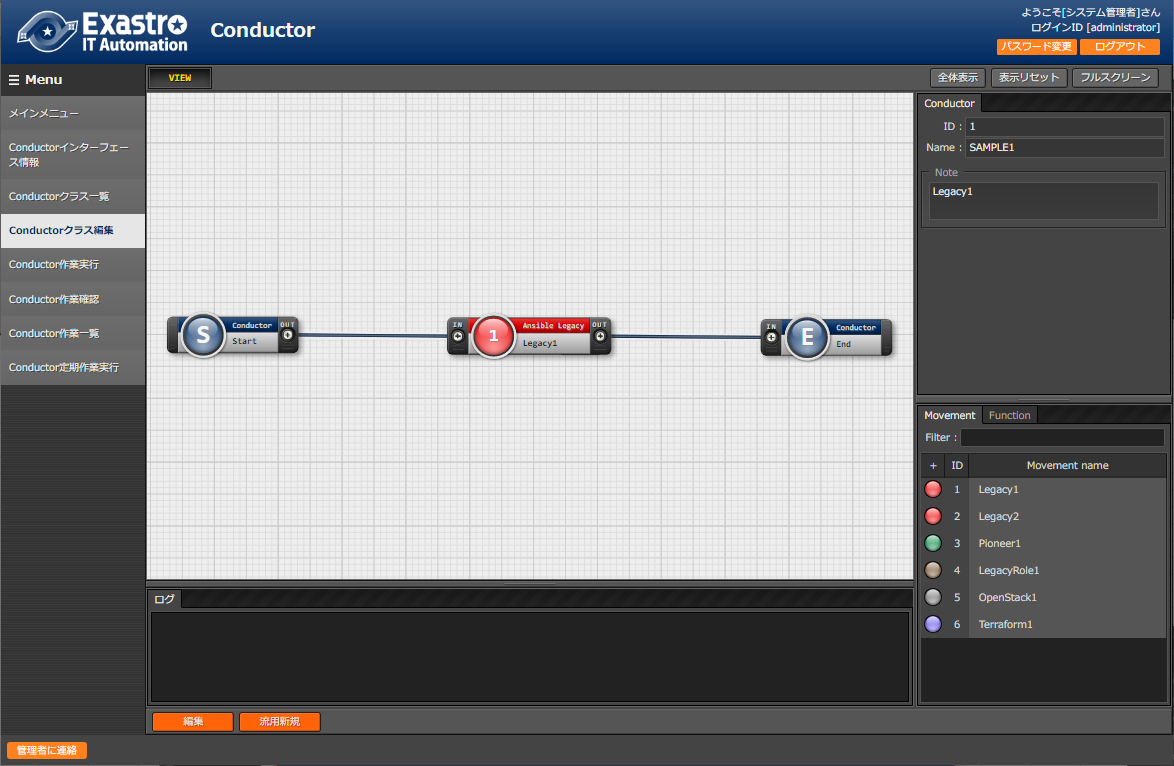
**図 4.1‑17詳細、入力項目（Parallel merge）**

* クラス編集画面で実行可能な操作は以下になります。

**表4.1-11　 Conductorクラス編集画面実行操作一覧**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **項目** | **説明** | **新規**  **（EDIT）** | **更新（VIEW）** | | **更新（EDIT）** | **備考** |
| 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、Consuctor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. [Conductor クラス一覧]画面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 4.1‑18サブメニュー画面（Conductorクラス編集：VIEW）**

admin@nec.amc.jp

~

~

~

~

:set bin noeol

* 「編集」ボタンで、登録済みのConductorを編集できます。

・　「流用新規」ボタンで、登録済みのConductorをコピーして新規作成が行えます。

1. 「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。

admin@nec.amc.jp

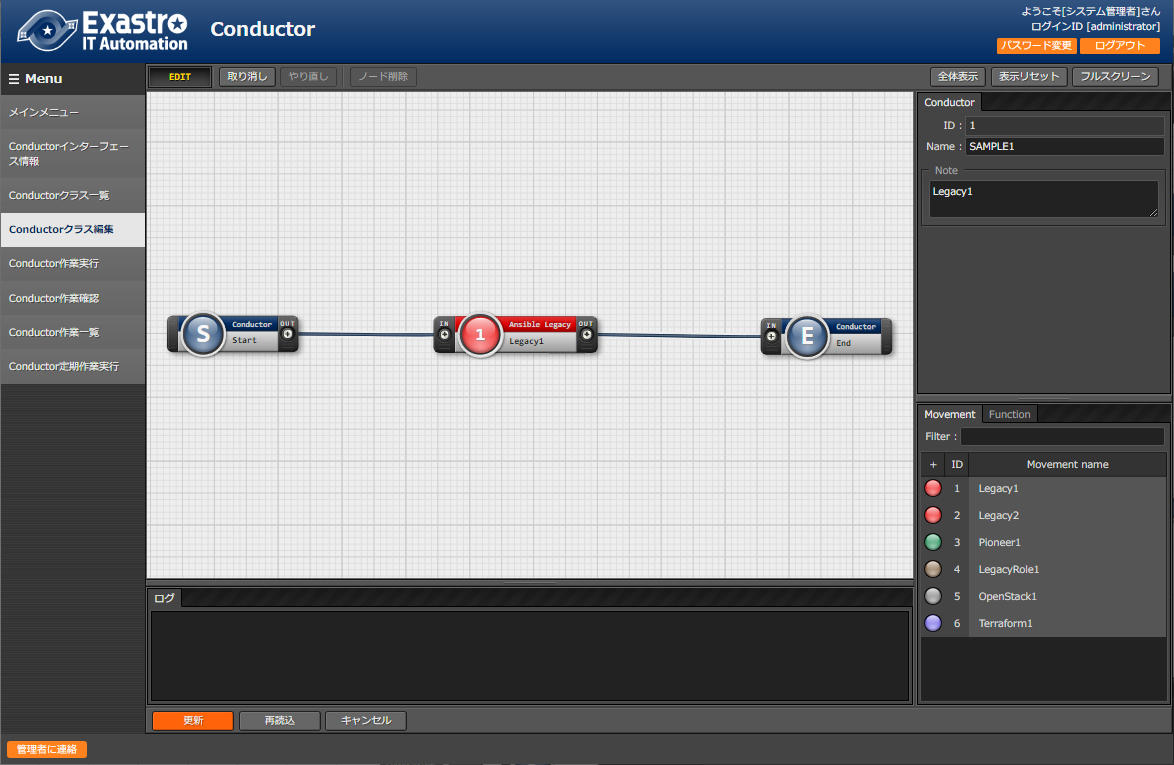
~

~

~

~

:set bin noeol



**図 4.1‑19サブメニュー画面（Conductorクラス編集：EDIT）**

* 「再読込」ボタンで、編集内容が破棄されて登録内容の状態に戻ります。

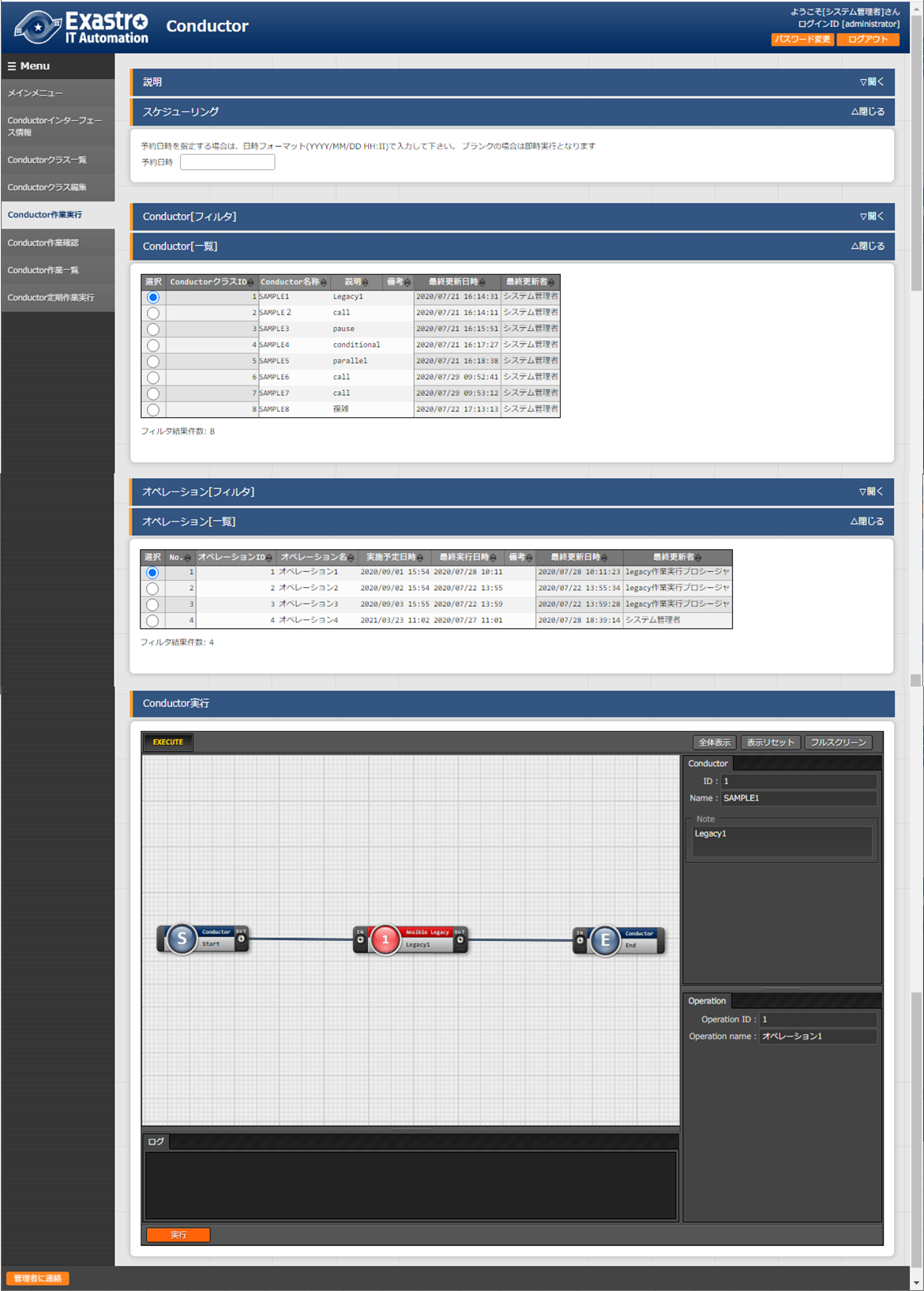
・　「更新」ボタンで、編集内容が保存されます。

・　「キャンセル」ボタンで、「編集」ボタン押下前の状態に戻ります。

Conductor作業実行

1. [Conductor 作業実行]画面では、Conductor実行の指示を行います。

* 「Conductor[一覧]」には「4.1.2 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]」には「基本コンソール」で登録したオペレーションが表示されます。
  1. 別紙、「利用手順マニュアル 基本コンソール」を参照。
* 「Conductor[一覧]、「オペレーション[一覧]」のラジオボタンからそれぞれ選択し、「実行ボタン」を押すと「4.1.5 Conductor作業確認」に遷移し、作業のトレースが始まります。
* 予約日時に入力して「実行ボタン」を押すと、作業予約が作られます。これは、「4.1.6 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のOperation, skipのみ、設定値を変更可能です。
  1. 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。



**図 4.1‑20サブメニュー画面（Conductor作業実行）**

Conductor実行画面の共通項目一覧は以下のとおりです。

**表4.1-12　登録画面項目一覧（Conductor実行）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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「Operationの指定について」参照 | - | 手動入力 |  |
| 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。

Conductor作業確認

1. [Conductor 作業確認]画面では、Conductorの実行状態を表示します。

「4.1.6Conductor作業一覧」の「詳細」表示ボタンをクリックすると、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

~

~

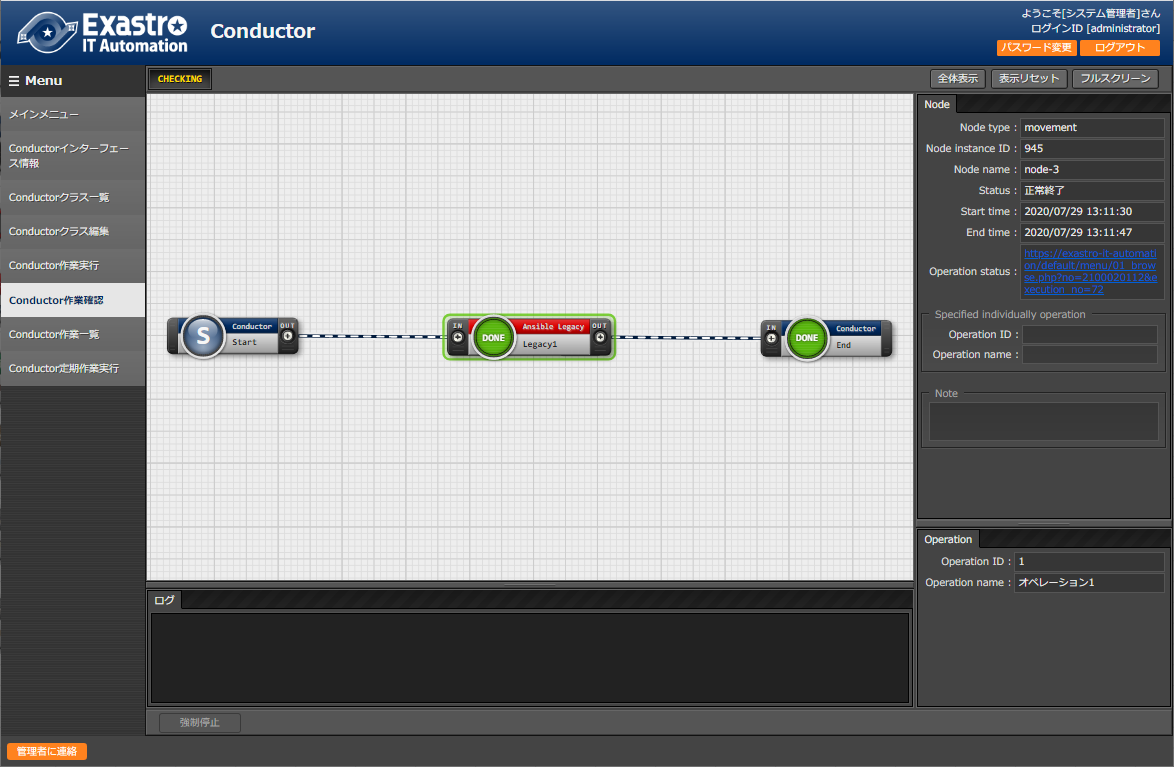
~

~

:set bin noeol

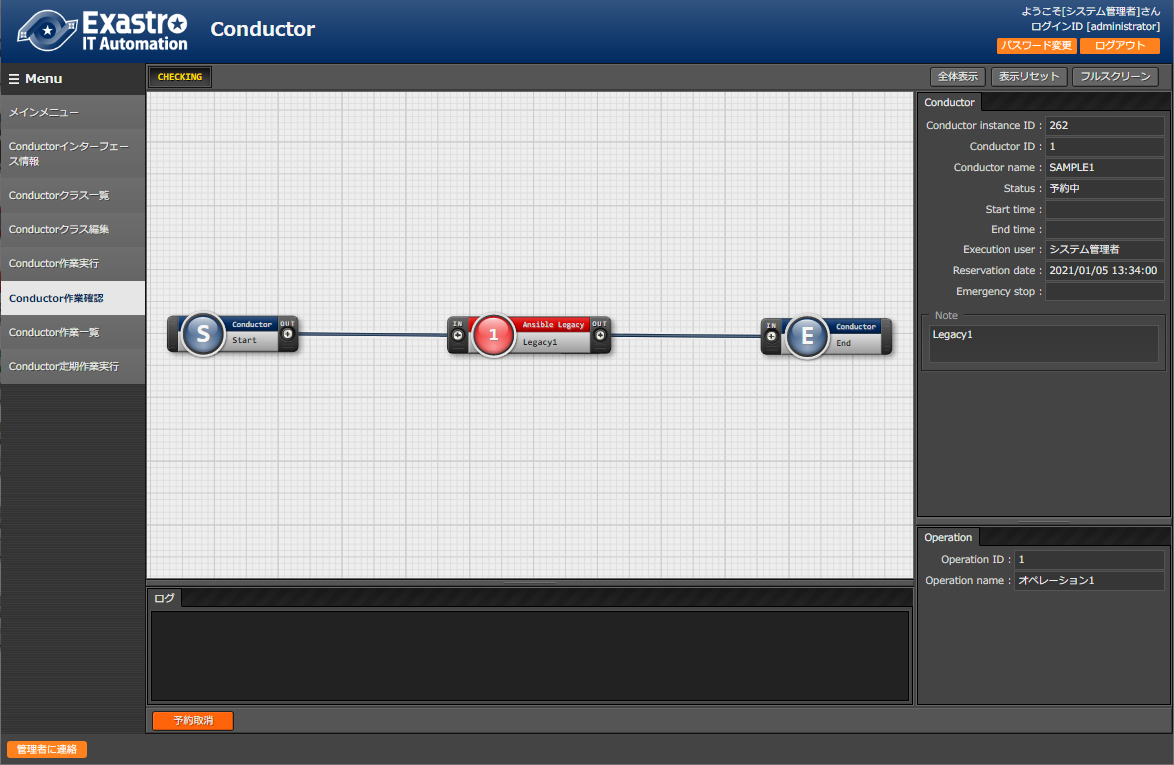
各ノードを選択すると、実行状況を表示します。

作業実行状況の詳細は、「Movement」、「Conductor Call」、の「Operation status」から作業状態確認画面に遷移できます。

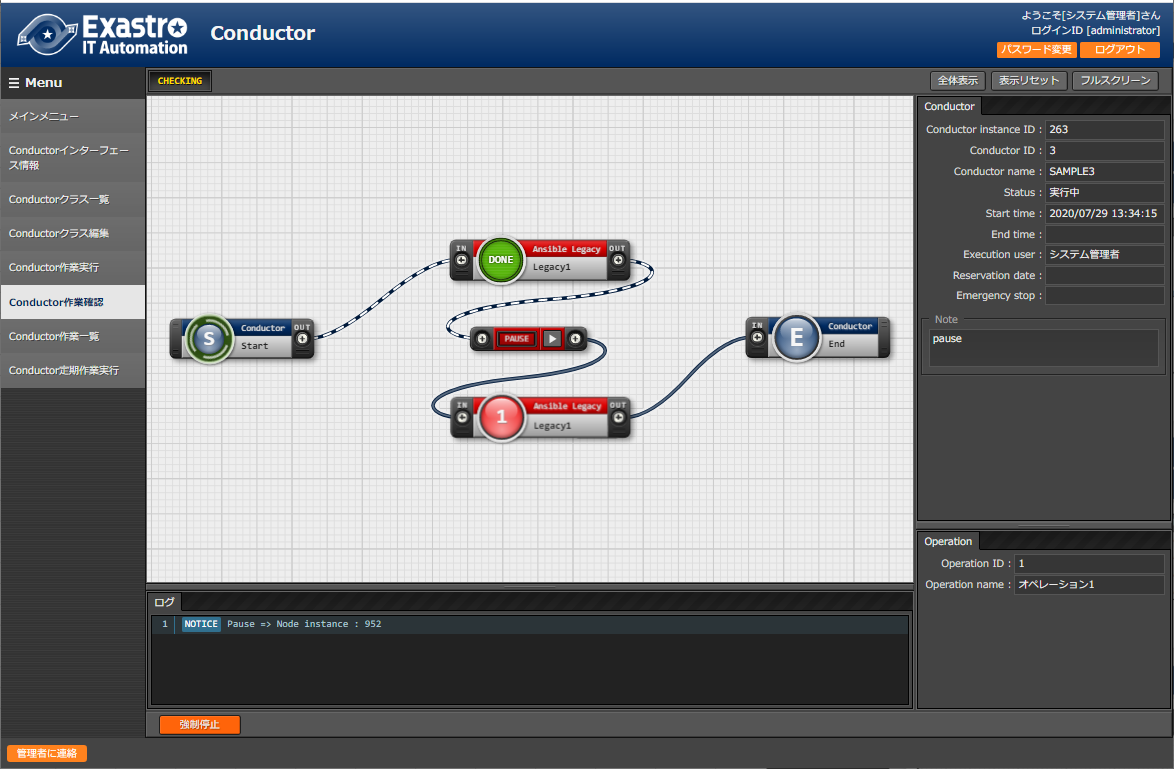


**図 4.1‑21サブメニュー画面（Conductor作業確認）**

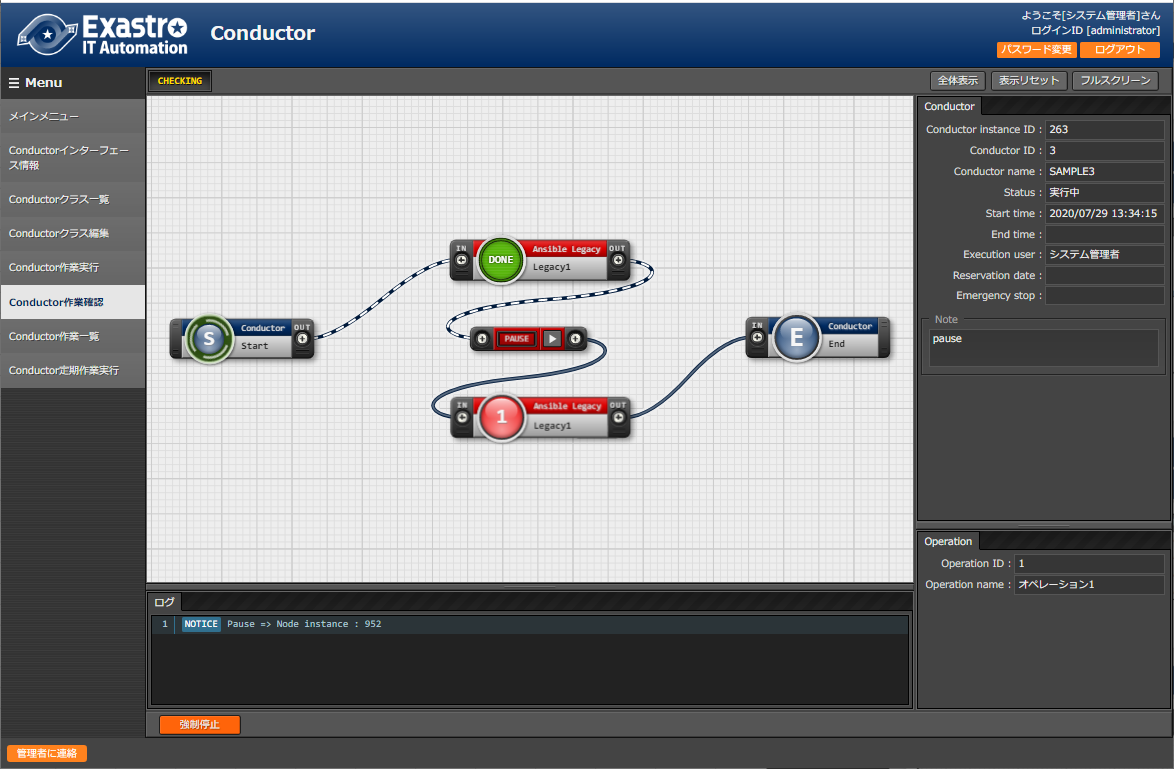
また、選択したConductor作業に予約日時が設定されていて、かつ未実行の場合は、予約取り消しボタンが表示されます。予約取り消しボタンを押すと、「4.1.6Conductor作業一覧」で確認できるステータスが「予約取り消し」となり、実行されなくなります。



**図 4.1‑22サブメニュー画面（Conductor作業確認 – 予約取消）**



**図 4.1‑23サブメニュー画面（Conductor作業確認 – 停止解除）**



**図 4.1‑24サブメニュー画面（Conductor作業確認 – 緊急停止）**

Conductor作業確認の共通項目一覧は以下のとおりです。

**表4.1-13　登録画面項目一覧（Conductor作業確認）**

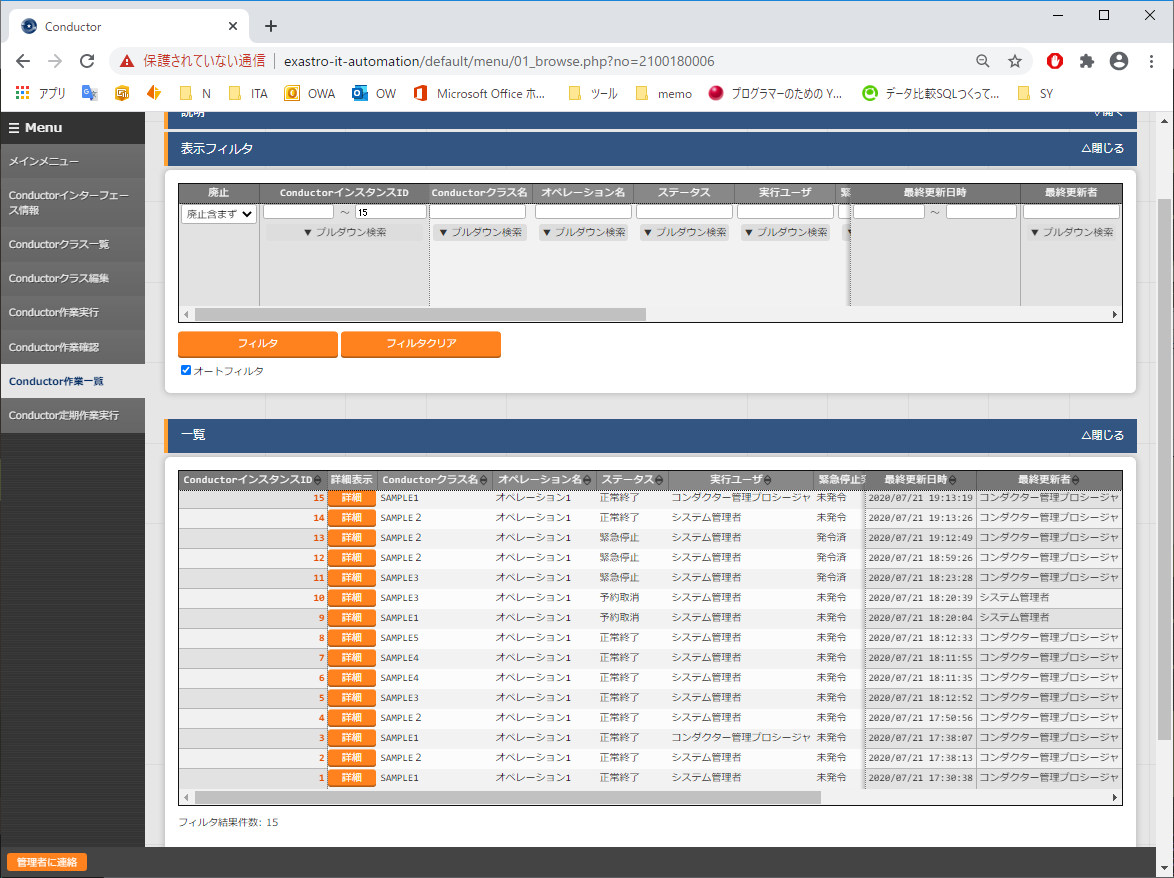
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **項目** | **説明** | **入力**  **必須** | **入力形式** | **制約事項** |
| 停止解除 | 一時停止を解除します | - | ボタン | - |
| 緊急停止 | Conductorの実行を中止します | - | ボタン | - |
| 予約取消 | Conductorの実行予約を取り消します | - | ボタン | 予約日時が設定されていて、かつ未実行の場合に表示される |

### 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、「4.1.5 Conductor作業確認」画面に遷移します。

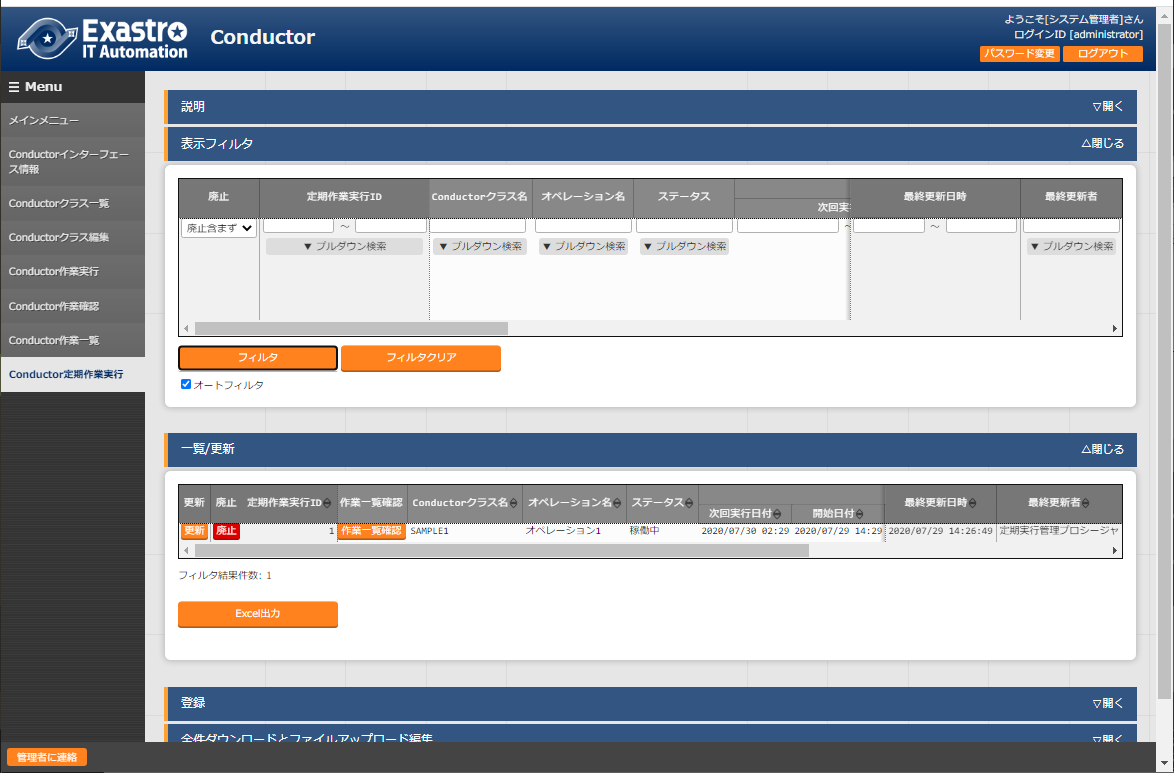


**図 4.1‑25サブメニュー画面（Conductor作業一覧）**

### 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「4.1.6Conductor作業一覧」へ遷移できます。



**図 4.1‑26サブメニュー画面（Conductor定期作業実行）**

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 4.1‑27スケジュール設定画面（定期作業実行）**

1. 登録画面の項目一覧は以下のとおりです

**表4.1-14　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 | | 「4.1.2Conductor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 | - |
| 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4.1-15　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「4.1.6Conductor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 | 「4.1.6Conductor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、「4.1.6Conductor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」の3分前に「4.1.6Conductor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。

※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「4.1.5 Conductor作業確認」より「保留解除」の投入を行わないと「4.1.6Conductor作業一覧」にステータス「実行中」で残り続けます。