拡張カスタムジョブ部品利用の手引き

JobCenter

R16.2

■Windows, Windows Server, Microsoft Azure, Microsoft Excel, Internet Explorer および Microsoft Edge は、米国 Microsoft Corporation 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。

■UNIX は、The Open Group が独占的にライセンスしている米国ならびにほかの国における登録商標です。

■HP-UX は、米国 HP Hewlett Packard Group LLC の商標です。

■AIX は、米国 IBM Corporation の商標です。

■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。

■Oracle Linux, Oracle Clusterware および Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。

■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。

■SUSE は、SUSE LLC 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。

■NQS は、NASA Ames Research Center のために Sterling Software 社が開発した Network Queuing System です。

■SAP ERP, SAP NetWeaver BW および ABAP は、SAP AG の登録商標または商標です。

■Amazon Web Services およびその他の AWS 商標は、Amazon.com, Inc. またはその関連会社の米国およびその他の国における商標です。

■iPad, iPadOS および Safari は、米国およびその他の国で登録された Apple Inc. の商標です。

■iOS は、Apple Inc. のOS名称です。IOS は、Cisco Systems, Inc. またはその関連会社の米国およびその他の国における商標または登録商標であり、ライセンスに基づき使用されています。

■Docker は、米国およびその他の国で登録された Docker, Inc. の登録商標または商標です。

■Firefox は、Mozilla Foundation の米国およびその他の国における商標または登録商標です。

■UiPath は、UiPath 社の米国およびその他の国における商標です。

■Box, boxロゴは、Box, Inc. の米国およびその他の国における商標または登録商標です。

■その他、本書に記載されているソフトウエア製品およびハードウエア製品の名称は、関係各社の登録商標または商標です。

なお、本書内では、R、TM、cの記号は省略しています。

本マニュアルでは、製品名およびサービス名を次のように略称表記しています。

略称	製品名・サービス名
Office	Microsoft Office
Excel	Microsoft Excel
Azure	Microsoft Azure
Internet Explorer	Internet Explorer 11
Firefox	Mozilla Firefox
AWS	Amazon Web Services
EC2	Amazon Elastic Compute Cloud
EBS	Amazon Elastic Block Store
S3	Amazon Simple Storage Service
ELB	Elastic Load Balancing
CloudFormation, CF	AWS CloudFormation
CloudWatch, CW	Amazon CloudWatch
RDS	Amazon Relational Database Service
Glue	AWS Glue
Lambda	AWS Lambda
EKS	Amazon Elastic Kubernetes Service
ECS	Amazon Elastic Container Service
STS	AWS Security Token Service
CloudWatch Logs	Amazon CloudWatch Logs
SNS	Amazon Simple Notification Service

輸出する際の注意事項

本製品（ソフトウエア）は、外国為替令に定める提供を規制される技術に該当いたしますので、日本国外へ持ち出す際には日本国政府の役務取引許可申請等必要な手続きをお取りください。許可手続き等にあたり特別な資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

はじめに

本書は、拡張カスタムジョブ部品の利用方法について説明しています。 なお、本書内に記載されている画面例と実際の画面とは異なることがありますので注意してください。

本書の内容は将来、予告なしに変更する場合があります。あらかじめご了承下さい。

1. マニュアルの読み方

■本バージョンにおける新規機能や変更事項を理解したい場合

→ スタンダードモードの場合は <スタンダードモード用リリースメモ>、 クラシックモードの場合は <クラシックモード用リリースメモ>を参照してください。

■JobCenter を新規にインストール、またはバージョンアップされる場合

→ スタンダードモードの場合は <セットアップガイド>、 クラシックモードの場合は <クラシックモード用インストールガイド>を参照してください。

■JobCenter をコンテナ環境で構築、運用をする場合

→ <クラシックモード用コンテナガイド>を参照してください。

■JobCenter を初めて利用される場合

→ スタンダードモードの場合は <セットアップガイド>、 クラシックモードの場合は <クラシックモード用クイックスタート編>を参照してください。

■JobCenter の基本的な操作方法を理解したい場合

→ スタンダードモードの場合は <スタンダードモード用基本操作ガイド>、 クラシックモードの場合は <クラシックモード用基本操作ガイド>を参照してください。

■環境の構築や各種機能の設定を理解したい場合

→ スタンダードモードの場合は <スタンダードモード用環境構築ガイド>、 クラシックモードの場合は <クラシックモード用環境構築ガイド>を参照してください。

■JobCenter の操作をコマンドラインから行う場合

→ スタンダードモードの場合は <スタンダードモード用コマンドリファレンス>、 クラシックモードの場合は <クラシックモード用コマンドリファレンス>を参照してください。

■JobCenter の運用方法を理解したい場合

→ スタンダードモードの場合は <スタンダードモード用移行ガイド>、 クラシックモードの場合は <クラシックモード用移行ガイド>を参照してください。

■運用中のJobCenter を新環境に移行する場合

→ スタンダードモードの場合は <スタンダードモード用移行ガイド>、 クラシックモードの場合は <クラシックモード用移行ガイド>を参照してください。

■クラスタ環境で運用中のJobCenter をバージョンアップする場合

→ <スタンダードモード用移行ガイド>を参照してください。

■その他機能についてお知りになりたい場合

→ 関連マニュアルの内容をお読みいただき、目的のマニュアルを参照してください。

2. 凡例

本書内での凡例を紹介します。

	気をつけて読んでいただきたい内容です。
	本文中の補足説明
	本文中のヒントとなる説明
注	本文中につけた注の説明
__	UNIX版のインストール画面の説明では、__部分(下線部分)はキーボードからの入力を示します。

3. 関連マニュアル

JobCenter に関するマニュアルです。JobCenter メディア内に格納されています。

最新のマニュアルは、JobCenter 製品サイトのダウンロードのページを参照してください。

https://jpn.nec.com/websam/jobcenter/download.html

【スタンダードモードのマニュアル】

資料名	概要
JobCenter セットアップガイド	JobCenterを新規にインストール、またはバージョンアップする場合の方法について説明しています。
JobCenter 基本操作ガイド	JobCenterの基本機能、操作方法について説明しています。
JobCenter 環境構築ガイド	JobCenterを利用するために必要なジョブ実行マネージャ環境の構築方法や設定方法の詳細、マネージャ環境の運用に役立つ機能について説明しています。
JobCenter ジョブ実行エージェント構築ガイド	JobCenterを利用するために必要なジョブ実行エージェント環境の構築方法や設定方法の詳細について説明しています。
JobCenter コマンドリファレンス	GUIと同様にジョブネットワークの投入、実行状況の参照などをコマンドラインから行うために、JobCenterで用意されているコマンドについて説明しています。
JobCenter クラスタ機能利用の手引き	クラスタシステムでJobCenterを操作するための連携方法について説明しています。
JobCenter Web機能利用の手引き	WebブラウザでジョブJobCenter CL/Webについて説明しています。
JobCenter 移行ガイド	運用中のJobCenterを別の新環境に移行する手順について横断的に説明しています。
JobCenter R16.2 リリースメモ	バージョン固有の情報を記載しています。

【クラシックモードのマニュアル】

資料名	概要
JobCenter インストールガイド	JobCenterを新規にインストール、またはバージョンアップする場合の方法について説明しています。
JobCenter クイックスタート編	初めてJobCenterをお使いになる方を対象に、JobCenterの基本的な機能と一通りの操作を説明しています。
JobCenter 基本操作ガイド	JobCenterの基本機能、操作方法について説明しています。
JobCenter 環境構築ガイド	JobCenterを利用するために必要な環境の構築、環境の移行や他製品との連携などの各種設定方法について説明しています。
JobCenter NQS機能利用の手引き	JobCenterの基盤であるNQSの機能をJobCenterから利用する方法について説明しています。
JobCenter コマンドリファレンス	GUIと同様にジョブネットワークの投入、実行状況の参照などをコマンドラインから行うために、JobCenterで用意されているコマンドについて説明しています。
JobCenter クラスタ機能利用の手引き	クラスタシステムでJobCenterを操作するための連携方法について説明しています。
JobCenter SAP機能利用の手引き	JobCenterをSAPと連携させるための方法について説明しています。
JobCenter WebOTX Batch Server連携機能利用の手引き	JobCenterをWebOTX Batch Serverと連携させるための方法について説明しています。

資料名	概要
JobCenter Web機能利用の手引き	Webブラウザ上でジョブ監視を行うことができるJobCenter CL/Webについて説明しています。
JobCenter クラスタ環境でのバージョンアップ・パッチ適用ガイド	クラスタ環境で運用しているJobCenterのアップデート、パッチ適用手順を説明しています。
JobCenter 運用・構築ガイド	JobCenterの設計、構築、開発、運用について横断的に説明しています。
JobCenter 移行ガイド	運用中のJobCenterを別の新環境に移行する手順について横断的に説明しています。
JobCenter コンテナガイド	JobCenterをコンテナ環境で構築・運用する方法について説明しています。
JobCenter R16.2 リリースメモ	バージョン固有の情報を記載しています。

【共通のマニュアル】

資料名	概要
JobCenter 操作・実行ログ機能利用の手引き	JobCenter CL/Winからの操作ログ、ジョブネットワーク実行ログ取得機能および設定方法について説明しています。
JobCenter Helper機能利用の手引き	Excelを用いたJobCenterの効率的な運用をサポートするJobCenter Definition Helper (定義情報のメンテナンス)、JobCenter Report Helper (帳票作成)、JobCenter Analysis Helper (性能分析)の3つの機能について説明しています。
JobCenter テキスト定義機能の利用手引き	JobCenterの定義情報をテキストファイルで定義する方法について説明しています。
JobCenter 拡張カスタムジョブ部品利用の手引き	拡張カスタムジョブとして提供される各部品の利用方法について説明しています。

4. 改版履歴

版数	変更日付	項目	形式	変更内容
1	2023/06/23	新規作成	－	第1版
2	2023/09/15	修正	－	R16.2.1リリースに伴い版改訂
3	2024/05/24	修正	－	R16.2.2リリースに伴い版改訂

目次

はじめに iv
1. マニュアルの読み方 v
2. 凡例 vi
3. 関連マニュアル vii
4. 改版履歴 ix
1. 概要 1
1.1. 適用方法 2
1.2. 共通仕様 3
1.2.1. 投入キュー(クラシックモードのみ) 3
1.2.2. カスタムパラメータの設定 4
1.2.3. カスタムジョブテンプレートのカスタマイズ 14
1.2.4. 表示アイコン 14
1.2.5. 注意事項 15
2. メール送信部品 16
2.1. メール送信部品 17
2.1.1. 部品情報 17
2.1.2. スクリプト 17
2.1.3. カスタムパラメータ設定 17
2.1.4. 実行結果 18
2.1.5. 主要メッセージ 18
2.1.6. 注意事項 19
3. HTTP接続部品 20
3.1. HTTP接続部品共通仕様 21
3.1.1. リクエスト種別(HTTPメソッド) 21
3.1.2. 制御パラメータ(カスタムパラメータ)一覧 21
3.1.3. HTTPリクエストパラメータ 24
3.1.4. 実行結果 26
3.1.5. 主要メッセージ 27
3.1.6. 注意事項 27
3.2. HTTP-GET部品 28
3.2.1. 部品情報 28
3.3. HTTP-PUT部品 29
3.3.1. 部品情報 29
3.4. HTTP-POST部品 30
3.4.1. 部品情報 30
3.5. HTTP-DELETE部品 31
3.5.1. 部品情報 31
3.6. HTTP-HEAD部品 32
3.6.1. 部品情報 32
3.7. HTTP-PATCH部品 33
3.7.1. 部品情報 33
4. Amazon Web Services 連携部品 34
4.1. 共通仕様 35
4.1.1. スクリプトの記述 35
4.2. EC2 36
4.2.1. EC2部品共通のカスタムパラメータ 36
4.2.2. インスタンス起動 37
4.2.3. インスタンス停止 39
4.2.4. インスタンス情報取得 41
4.2.5. インスタンスプロパティ変更 43
4.2.6. AMI作成 45
4.2.7. 世代管理付きAMI作成 47
4.2.8. タグ作成 52
4.2.9. スナップショット作成 54

4.2.10. スナップショット削除 59
4.2.11. スナップショット情報取得 61
4.2.12. スナップショットコピー 63
4.2.13. EBSボリュームサイズ拡張 65
4.2.14. EBSボリュームタイプ変更 67
4.3. S3 69
4.3.1. S3部品共通のカスタムパラメータ 69
4.3.2. S3部品共通仕様 69
4.3.3. ファイルダウンロード 70
4.3.4. ファイルアップロード 74
4.3.5. ファイル削除 78
4.3.6. S3部品共通の注意事項 79
4.4. ELB 80
4.4.1. ELB部品共通のカスタムパラメータ 80
4.4.2. クラシックロードバランサーへのインスタンス登録 81
4.4.3. クラシックロードバランサーからのインスタンス登録解除 83
4.5. CF 85
4.5.1. CF部品共通のカスタムパラメータ 85
4.5.2. スタック作成 86
4.5.3. スタック削除 88
4.5.4. スタック情報取得 90
4.6. CW 92
4.6.1. CW部品共通のカスタムパラメータ 92
4.6.2. アラーム情報取得 93
4.6.3. アラーム状態待ち合わせ 95
4.7. RDS 98
4.7.1. RDS部品共通のカスタムパラメータ 98
4.7.2. インスタンス起動 99
4.7.3. インスタンス停止 102
4.8. Glue 106
4.8.1. Glue部品共通のカスタムパラメータ 106
4.8.2. ジョブ実行 107
4.9. Lambda 113
4.9.1. Lambda部品共通のカスタムパラメータ 113
4.9.2. 関数実行 114
4.10. EKS 118
4.10.1. EKS部品共通のカスタムパラメータ 118
4.10.2. ジョブ実行 119
4.11. ECS 126
4.11.1. ECS部品共通のカスタムパラメータ 126
4.11.2. タスク実行 127
5. Microsoft Azure(V2) 連携部品 137
5.1. 部品共通のカスタムパラメータ 138
5.1.1. カスタムパラメータ設定 138
5.2. 仮想マシン起動部品 139
5.2.1. 部品情報 139
5.2.2. 部品説明 139
5.2.3. スクリプト 139
5.2.4. カスタムパラメータ設定 139
5.2.5. 実行結果 140
5.2.6. 主要メッセージ 140
5.2.7. 注意事項 140
5.3. 仮想マシン停止およびリソース解放部品 141
5.3.1. 部品情報 141
5.3.2. 部品説明 141
5.3.3. スクリプト 141

5.3.4. カスタムパラメータ設定 141
5.3.5. 実行結果 142
5.3.6. 主要メッセージ 142
5.3.7. 注意事項 142
5.4. 仮想マシン停止部品 143
5.4.1. 部品情報 143
5.4.2. 部品説明 143
5.4.3. スクリプト 143
5.4.4. カスタムパラメータ設定 143
5.4.5. 実行結果 144
5.4.6. 主要メッセージ 144
5.4.7. 注意事項 144
5.5. 仮想マシン情報(モデルビュー)取得 145
5.5.1. 部品情報 145
5.5.2. 部品説明 145
5.5.3. スクリプト 145
5.5.4. カスタムパラメータ設定 145
5.5.5. 実行結果 145
5.5.6. 主要メッセージ 146
5.5.7. 注意事項 146
5.6. 仮想マシン情報(インスタンスビュー)取得 147
5.6.1. 部品情報 147
5.6.2. 部品説明 147
5.6.3. スクリプト 147
5.6.4. カスタムパラメータ設定 147
5.6.5. 実行結果 147
5.6.6. 主要メッセージ 148
5.6.7. 注意事項 148
6. UiPath 連携部品 149
6.1. ジョブ開始 150
6.1.1. 部品情報 150
6.1.2. 部品説明 150
6.1.3. スクリプト 150
6.1.4. カスタムパラメータ設定 150
6.1.5. 部品詳細 151
6.1.6. 実行結果 153
6.1.7. 主要メッセージ 154
6.1.8. 使用API 154
6.1.9. 注意事項・制限事項 155
7. Box連携部品 156
7.1. Box連携部品概要 157
7.1.1. ジョブネットワーク作成方法 157
7.2. 事前準備 161
7.3. Box連携部品共通の仕様 166
7.3.1. Box連携部品共通で設定可能なパラメータ 166
7.3.2. APIトークンを取得する部品を利用する方法でのパラメータ設定 166
7.3.3. Boxの操作を行う部品を単体で利用する方法でのパラメータ設定 167
7.3.4. ワイルドカード設定 167
7.4. BOX_APIトークン取得部品 168
7.4.1. 部品情報 168
7.4.2. 部品説明 168
7.4.3. スクリプト 168
7.4.4. カスタムパラメータ設定 168
7.4.5. 利用例 168
7.4.6. 実行結果 169
7.4.7. 主要メッセージ 170

7.4.8. 注意事項 170
7.5. BOXアップロード部品 171
7.5.1. 部品情報 171
7.5.2. 部品説明 171
7.5.3. スクリプト 171
7.5.4. カスタムパラメータ設定 171
7.5.5. 利用例 172
7.5.6. 実行結果 174
7.5.7. 主要メッセージ 175
7.5.8. 注意事項 176
7.6. BOXダウンロード部品 177
7.6.1. 部品情報 177
7.6.2. 部品説明 177
7.6.3. スクリプト 177
7.6.4. カスタムパラメータ設定 177
7.6.5. 利用例 178
7.6.6. 実行結果 180
7.6.7. 主要メッセージ 181
7.6.8. 注意事項 182
7.7. コンテンツID取得部品 183
7.7.1. 部品情報 183
7.7.2. 部品説明 183
7.7.3. スクリプト 183
7.7.4. カスタムパラメータ設定 183
7.7.5. 実行結果 183
7.7.6. 主要メッセージ 184
7.7.7. 注意事項 184
8. SystemManager G連携部品 185
8.1. メッセージ監視機能概要 186
8.1.1. 事前準備 186
8.2. ACOS監視機能概要 191
8.2.1. 事前準備 191
8.2.2. ジョブネットワークの設定 192
8.3. メッセージ監視登録部品 198
8.3.1. 部品情報 198
8.3.2. 部品説明 198
8.3.3. スクリプト 198
8.3.4. カスタムパラメータ設定 198
8.3.5. 実行結果 199
8.3.6. 主要メッセージ 199
8.3.7. 注意事項 199
8.4. メッセージ待ち合わせ部品 200
8.4.1. 部品情報 200
8.4.2. 部品説明 200
8.4.3. スクリプト 200
8.4.4. カスタムパラメータ設定 200
8.4.5. 実行結果 215
8.4.6. 主要メッセージ 216
8.4.7. 注意事項 216
8.5. ACOSジョブ実行部品 217
8.5.1. 部品情報 217
8.5.2. 部品説明 217
8.5.3. スクリプト 217
8.5.4. カスタムパラメータ設定 217
8.5.5. 実行結果 219
8.5.6. 主要メッセージ 221

8.5.7. 注意事項 221
8.6. SystemManager G連携部品の注意事項 222

1. 概要

本章では拡張カスタムジョブ部品について説明します。

拡張カスタムジョブは、部品の種別毎にJPFファイルとして提供され、ファイルをアップロードすることで利用することができます。

アップロードした部品はマネージャフレーム中のカスタムジョブ定義一覧で参照することができます。種別毎にグループ化され、Systemグループ配下に格納されます。

JobCenter環境へのアップロードの方法は 「1.1 適用方法」を参照してください。アップロードのより詳細な説明については、 スタンダードモードの場合は <スタンダードモード用基本操作ガイド>の11章 「アップロード・ダウンロード機能」 、 クラシックモードの場合は <クラシックモード用基本操作ガイド>の12章 「アップロード・ダウンロード機能」 を参照してください。 また、カスタムジョブ定義の利用方法については、 スタンダードモードの場合は <スタンダードモード用基本操作ガイド>の5章 「カスタムジョブの利用方法」 、 クラシックモードの場合は <クラシックモード用基本操作ガイド>の5章 「カスタムジョブの利用方法」 を参照してください。

本手引きでは各部品の利用方法について説明します。

1.1. 適用方法

1. 拡張カスタムジョブのJPFファイルを、CL/Winがインストールされたマシンの任意の場所にコピーします。

拡張カスタムジョブのJPFファイルはJobCenterメディアに同梱されています。また最新JPFファイルはJobCenter 製品サイトのダウンロードページから入手する事ができます。

JobCenterメディア同梱のJPFファイルの格納場所は、JobCenterメディアのリリースメモ(RELMEMO)を参照してください。

最新JPFファイルをダウンロードできるJobCenter製品サイトのページについては、以下をご覧ください。このページの閲覧をするためには、サポートポータルへのログイン(JobCenterの保守契約)が必要です

https://www.support.nec.co.jp/View.aspx?id=3140103238

2. CL/WinからJobCenter管理者ユーザで対象サーバへログインします。　[ファイル]-[Helper機能]-[アップロード]を選択します。

3. 「アップロード対象のユーザの指定」画面でJobCenter管理者ユーザを選択します。

4. (1)でコピーしたJPFファイルを選択して「アップロード」を行います。

5. CL/Winのマネージャフレームで、カスタムジョブ定義一覧を表示して、Systemグループ配下に拡張カスタムジョブの定義が登録されていることを確認して下さい。

1.2. 共通仕様

1.2.1. 投入キュー(クラシックモードのみ)

拡張カスタムジョブ部品はパイプキューに投入することもできますが、カスタムキュー属性を持つバッチキューで実行する必要があります。カスタムキュー属性を持たないバッチキューに投入した場合には、正常に動作しません。

1.2.1.1. カスタムキュー属性の付与手順

バッチキューにカスタムキュー属性を付与する手順を説明します。

1. CL/WinからJobCenter管理者ユーザで対象サーバへログインし、NQSフレームを開いて［マシン一覧］を表示します。

2. カスタムキュー属性を付与したいバッチキューを選択し、右クリックしたときのポップアップメニューから［キューパラメータ］を選択します。

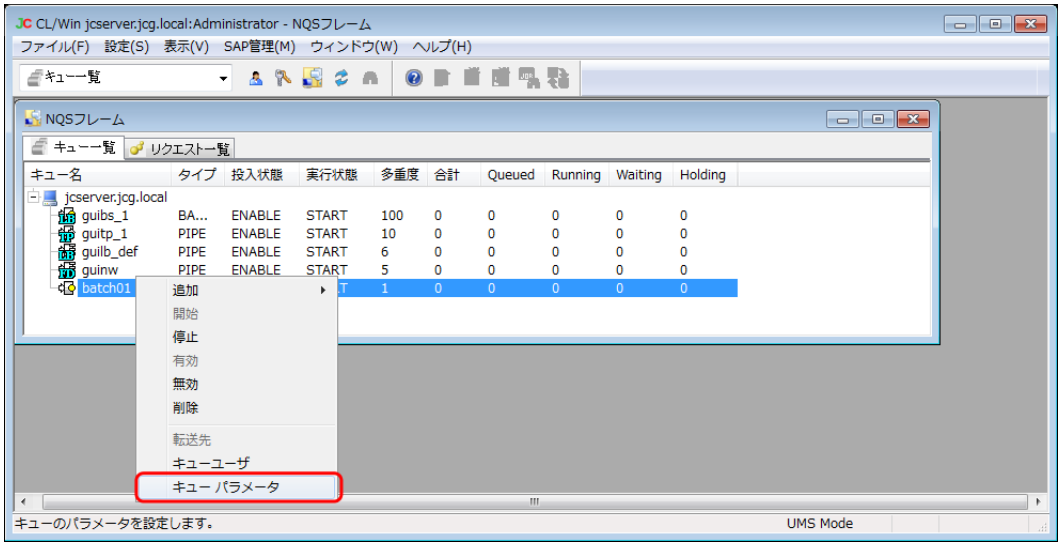


図1.1 バッチキューの［キューパラメータ］選択画面例

3. ［バッチキューパラメータ］ダイアログが表示されるので、［キュー］タブ－［キュー属性］－［Custom］を［ON］に設定します。

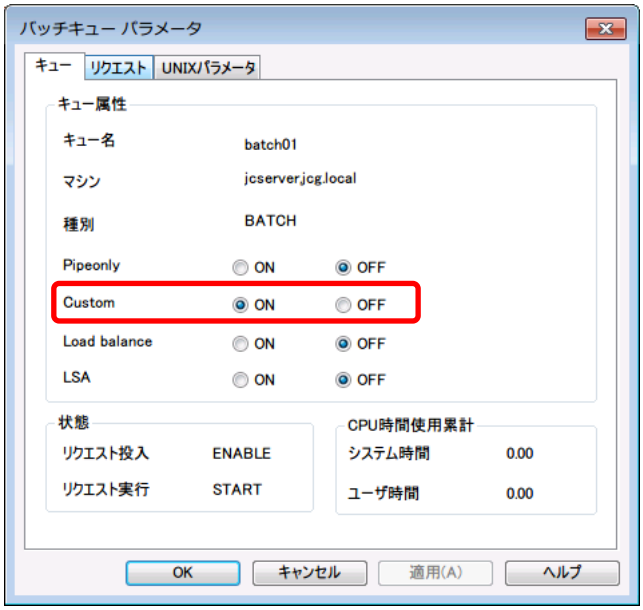
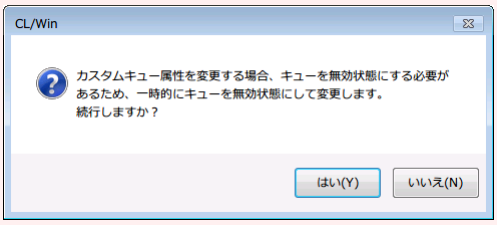


図1.2 バッチキューパラメータの設定画面例

4. 設定後、［OK］ボタンをクリックします。

カスタムキュー属性を変更する場合、キューを無効状態にする必要があります。キューを有効状態のまま変更する場合は、次のような警告ダイアログが表示されます。続行すると、内部動作として一時的にキューを無効化したうえでカスタムキュー属性を変更し、キューを有効状態に戻します。

本ダイアログが表示された場合は、設定変更するキューにジョブが投入されないことを事前に確認してから続行してください。



1.2.2. カスタムパラメータの設定

拡張カスタムジョブ部品の設定はカスタムパラメータで行います。ジョブネットワークフロー画面で対象の拡張カスタムジョブ部品のパラメータ画面を開き、[カスタムパラメータ設定]タブから対象のパラメータを選択し[編集]ボタンを押下して設定します。

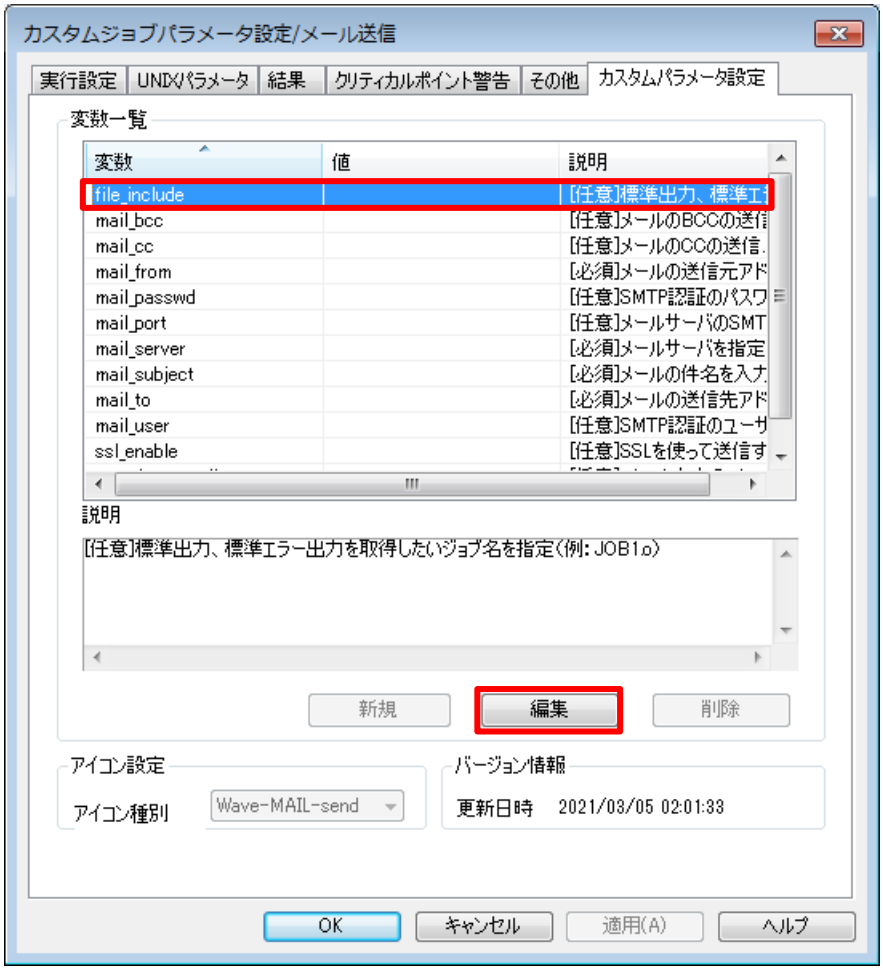


図1.3 カスタムパラメータ設定画面

拡張カスタムジョブ部品では、カスタムパラメータのデータ型によって以下の入力ダイアログを使用します。

データ型	入力ダイアログ	説明
リスト	「1.2.2.1 リスト型の入力ダイアログ」	曜日など、決められた固定値から選択するパラメータで利用します。
数値	「1.2.2.2 数値型の入力ダイアログ」	数値を入力するパラメータで利用します。
日付	「1.2.2.3 日付型の入力ダイアログ」	日付を入力するパラメータで利用します。
時間	「1.2.2.4 時間型の入力ダイアログ」	時間を入力するパラメータで利用します。
文字列	「1.2.2.5 文字列型の入力ダイアログ」	文字列を入力するパラメータで利用します。

カスタムパラメータの詳細およびデータ型については、各拡張カスタムジョブ部品の章を参照してください。

1.2.2.1. リスト型の入力ダイアログ

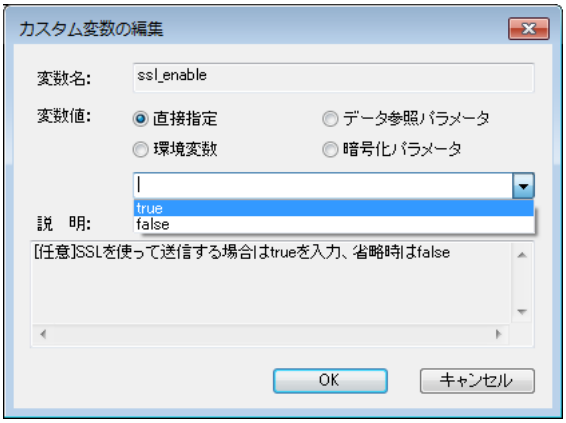


図1.4 リスト型の入力ダイアログ

データ型がリストであるカスタムパラメータを設定する際に利用する入力ダイアログです。

値を直接入力する場合は、[変数値]の[直接指定]を選択します。その他の方式で入力する場合は「1.2.2.6 カスタムパラメータ共通の入力方式」を参照してください。本節では[直接指定]の設定方法について説明します。

■値を設定する方法

プルダウンメニューから値を選択して[OK]ボタンを押下します。

入力欄をクリックして直接入力することも可能ですが、一覧に表示された値以外を入力した場合は[OK]ボタンを押下した際に入力エラーとなります。

■値を削除する方法

入力欄をクリックして値を削除して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

1.2.2.2. 数値型の入力ダイアログ

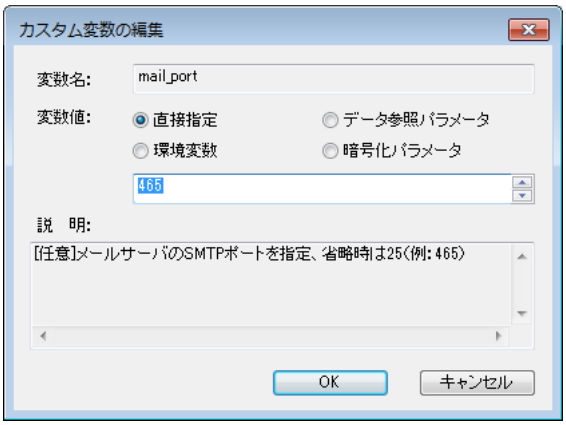


図1.5 数値型の入力ダイアログ

データ型が数値であるカスタムパラメータを設定する際に利用する入力ダイアログです。

値を直接入力する場合は、[変数値]の[直接指定]を選択します。その他の方式で入力する場合は「1.2.2.6 カスタムパラメータ共通の入力方式」を参照してください。本節では[直接指定]の設定方法について説明します。

■値を設定する方法

スピンボタンから値を入力して[OK]ボタンを押下します。

入力欄をクリックして直接入力することも可能ですが、範囲外の値を入力した場合は[OK]ボタンを押下した際に入力エラーとなります。

■値を削除する方法

入力欄をクリックして値を削除して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

1.2.2.3. 日付型の入力ダイアログ

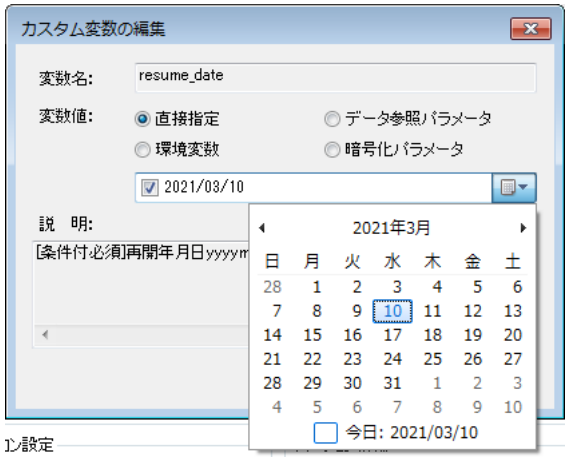


図1.6 日付型の入力ダイアログ

データ型が日付であるカスタムパラメータを設定する際に利用する入力ダイアログです。

値を直接入力する場合は、[変数値]の[直接指定]を選択します。その他の方式で入力する場合は「1.2.2.6 カスタムパラメータ共通の入力方式」を参照してください。本節では[直接指定]の設定方法について説明します。

■値を設定する方法

入力欄右側にあるカレンダボタンから日付を選択して[OK]ボタンを押下してください。対象のカスタムパラメータの書式に従った文字列が自動で設定されます。

入力欄左側にあるチェックボックスにチェックを入れた後に、入力欄の[年][月][日]の部分をクリックして直接入力することも可能ですが、範囲外の期間を入力した場合は[OK]ボタンを押下した際に入力エラーとなります。

■値を削除する方法

入力欄左側にあるチェックボックスのチェックを外して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

1.2.2.4. 時間型の入力ダイアログ

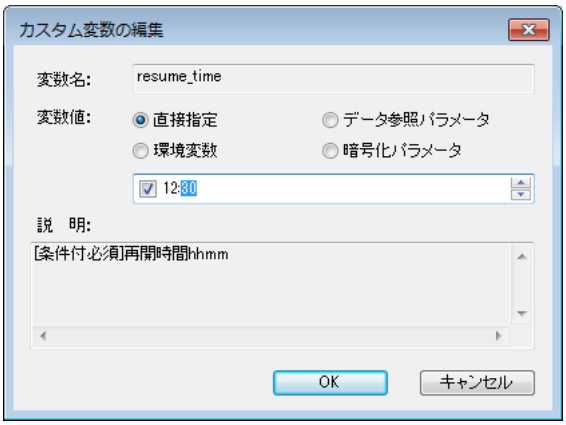


図1.7 時間型の入力ダイアログ

データ型が時間であるカスタムパラメータを設定する際に利用する入力ダイアログです。

値を直接入力する場合は、[変数値]の[直接指定]を選択します。その他の方式で入力する場合は「1.2.2.6 カスタムパラメータ共通の入力方式」を参照してください。本節では[直接指定]の設定方法について説明します。

■値を設定する方法

入力欄左側にあるチェックボックスにチェックを入れた後に、入力欄の[時][分]の部分をクリックし、スピンボタンから値を入力して[OK]ボタンを押下します。対象のカスタムパラメータの書式に従った文字列が自動で設定されます。

入力欄の[時][分]の部分をクリックして直接入力することも可能ですが、範囲外の時間を入力した場合は[OK]ボタンを押下した際に入力エラーとなります。

■値を削除する方法

入力欄左側にあるチェックボックスのチェックを外して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

1.2.2.5. 文字列型の入力ダイアログ

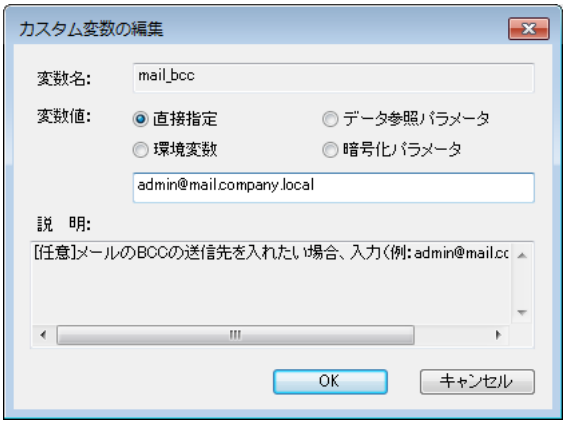


図1.8 文字列型の入力ダイアログ

データ型が文字列であるカスタムパラメータを設定する際に利用する入力ダイアログです。

値を直接入力する場合は、[変数値]の[直接指定]を選択します。その他の方式で入力する場合は「1.2.2.6 カスタムパラメータ共通の入力方式」を参照してください。本節では[直接指定]の設定方法について説明します。

■値を設定する方法

入力欄をクリックして値を入力して[OK]ボタンを押下します。

■値を削除する方法

入力欄をクリックして値を削除して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

1.2.2.6. カスタムパラメータ共通の入力方式

拡張カスタムジョブ部品の全てのカスタムパラメータは、共通で以下の入力方式を利用できます。

入力方式	利用方法	説明
データ参照パラメータ	「1.2.2.6.1 データ参照パラメータの利用」	特定のデータフォーマットで出力される先行ジョブ部品の実行結果を使用する場合に利用します。
暗号化パラメータ	「1.2.2.6.2 暗号化パラメータの利用」	paramctrlコマンドで登録した暗号化パラメータの値を使用する場合に利用します。
環境変数	「1.2.2.6.3 ジョブネットワークの環境変数の利用」	ジョブネットワークの環境変数の値を使用する場合に利用します。

1.2.2.6.1. データ参照パラメータの利用

拡張カスタムジョブ部品ではカスタムパラメータの設定時に、特定のデータフォーマットで出力される先行ジョブ部品の実行結果を参照する事ができます。

先行ジョブ部品の実行結果を参照するにはデータ参照パラメータを使用します。 データ参照パラメータについては、 スタンダードモードの場合は <スタンダードモード用基本操作ガイド>の16章 「先行ジョブ部品の実行結果の参照方法」 、 クラシックモードの場合は <クラシックモード用基本操作ガイド>の17章 「先行ジョブ部品の実行結果の参照方法」 を参照してください。

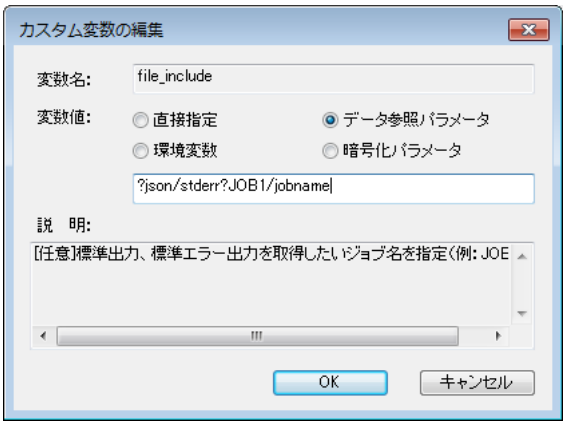


図1.9 データ参照パラメータの設定例

データ参照パラメータを拡張カスタムジョブ部品の実行時に展開するには、対象のカスタムパラメータの入力ダイアログを開き、以下の手順で設定します。

■値を設定する方法

1. [変数値]の[データ参照パラメータ]を選択します。

2. 入力欄をクリックしてデータ参照パラメータを入力して[OK]ボタンを押下します。

   不正なフォーマットを入力した場合は[OK]ボタンを押下した際に入力エラーとなります。データ参照パラメータのフォーマットは以下のとおりです。

   ```
   ?<データフォーマット>/<出力場所>?<部品名><キー>
   ```

■値を削除する方法

1. [変数値]の[直接指定]を選択します。

2. 入力欄が空欄であることを確認して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

1.2.2.6.2. 暗号化パラメータの利用

拡張カスタムジョブ部品ではカスタムパラメータの設定時に、あらかじめJobCenterに登録したパラメータへの展開を指定する事ができます。

JobCenterへ登録したパラメータは、JobCenter内部では暗号化して保存し、カスタムパラメータの展開も実行時に行うため、パスワードなどの秘匿情報の設定はこの暗号化パラメータの利用を推奨いたします。暗号化パラメータの登録方法については、 スタンダードモードの場合は <スタンダードモード用コマンドリファレンス>の「3.26 paramctrl 暗号化DBへのパラメータ登録」、 クラシックモードの場合は <クラシックモード用コマンドリファレンス>の「3.29 paramctrl 暗号化DBへのパラメータ登録」を参照してください。

なお、暗号化パラメータの登録は、拡張カスタムジョブ部品を配置したジョブネットワークフローを持つサイトで行ってください。

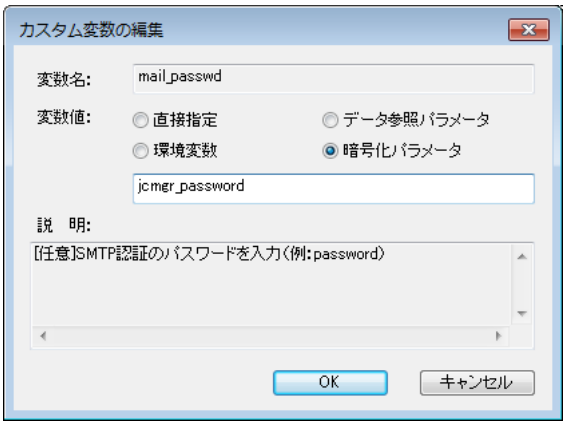


図1.10 暗号化パラメータの設定例

暗号化パラメータを拡張カスタムジョブ部品の実行時に展開するには、対象のカスタムパラメータの入力ダイアログを開き、以下の手順で設定します。

■値を設定する方法

1. [変数値]の[暗号化パラメータ]を選択します。

2. 入力欄をクリックして暗号化パラメータのキー名を入力して[OK]ボタンを押下します。カスタムパラメータ設定画面の変数一覧に、暗号化パラメータの書式($<キー名>$)に従った文字列が自動で設定されます。

■値を削除する方法

1. [変数値]の[直接指定]を選択します。
2. 入力欄が空欄であることを確認して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

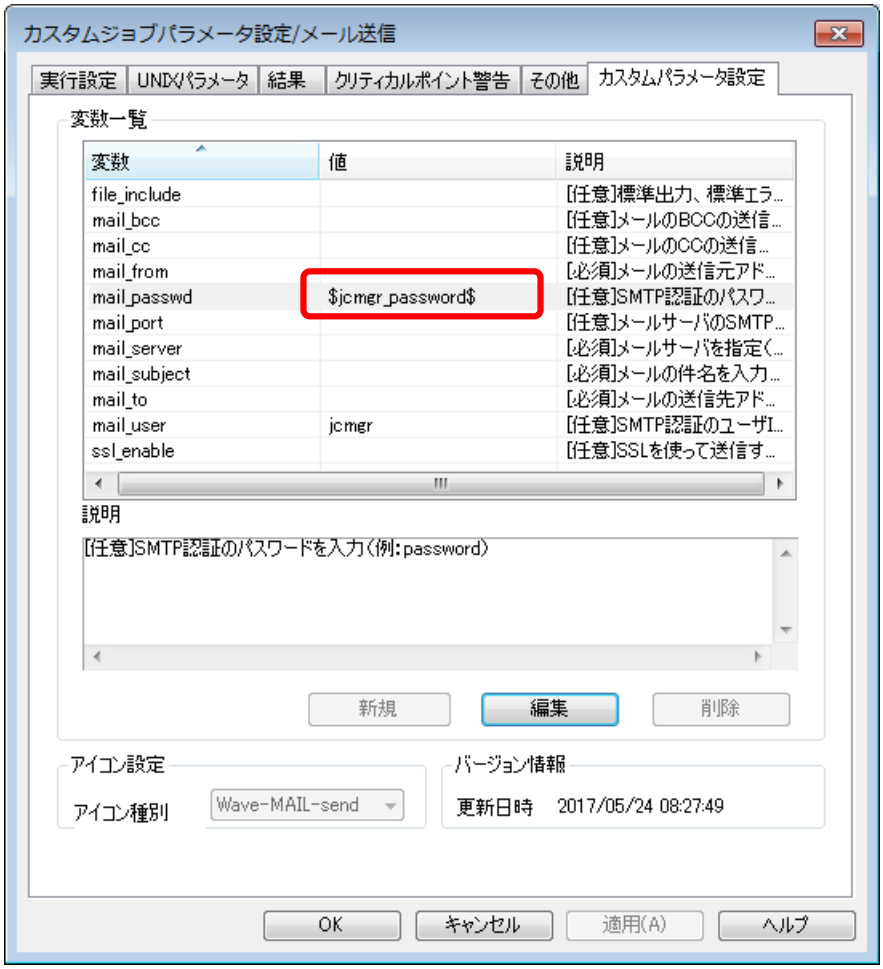


図1.11 暗号化パラメータ設定後のカスタムパラメータ設定画面

1.2.2.6.3. ジョブネットワークの環境変数の利用

拡張カスタムジョブ部品ではカスタムパラメータの設定時に、ジョブネットワークの環境変数の値を参照する事ができます。

変数継承機能と組み合わせて、カスタムパラメータの値を拡張カスタムジョブ部品の実行時に動的に決定することも可能です。 変数継承機能については、 スタンダードモードの場合は <スタンダードモード用基本操作ガイド>の「4.2.3 単位ジョブ間で情報を引き継ぐ」、 クラシックモードの場合は <クラシックモード用基本操作ガイド>の「4.2.3 単位ジョブ間で情報を引き継ぐ」を参照してください。

また、参照した環境変数の値が「key」(keyは展開したい暗号化パラメータのキー名)となっている場合、それを暗号化パラメータの展開の指定であるとみなして暗号化パラメータの値に展開されます。このように環境変数を経由することで、展開する暗号化パラメータのキー名を拡張カスタムジョブ部品の実行時に動的に決定することが可能です。

■環境変数名がマルチバイト文字である環境変数は指定できません。

■環境変数はサイズの上限があります。

- UNIX版JobCenterの場合

  拡張カスタムジョブ部品のパラメータ全体のサイズは128KByteが最大です。

  カスタムパラメータで環境変数を参照している場合、環境変数の値に置き換えた場合のサイズで計算します。

- Windows版JobCenterの場合

  環境変数1つあたりのサイズを、環境変数名(=も含む)と値あわせて8186byte以内にしておく必要があります。

1.2.2.6.3.1. カスタムパラメータに、単一の環境変数の値のみを使用する場合

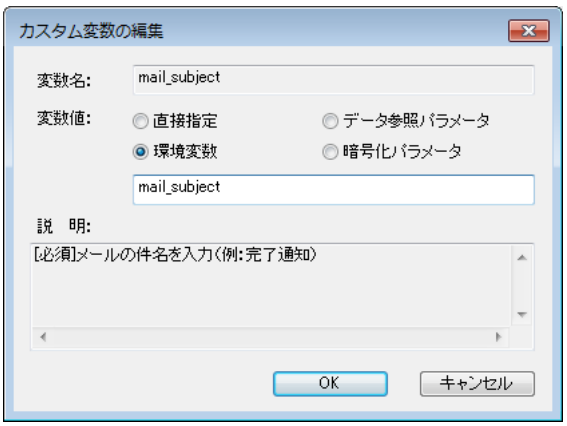


図1.12 ジョブネットワークの環境変数の設定例

ジョブネットワークの環境変数の値を拡張カスタムジョブ部品の実行時に参照するには、対象のカスタムパラメータの入力ダイアログを開き、以下の手順で設定します。

■値を設定する方法

1. [変数値]の[環境変数]を選択します。

2. 入力欄をクリックして環境変数名を入力して[OK]ボタンを押下します。カスタムパラメータ設定画面の変数一覧に、環境変数を参照する書式(%<環境変数名>%)に従った文字列が自動で設定されます。

■値を削除する方法

1. [変数値]の[直接指定]を選択します。

2. 入力欄が空欄であることを確認して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

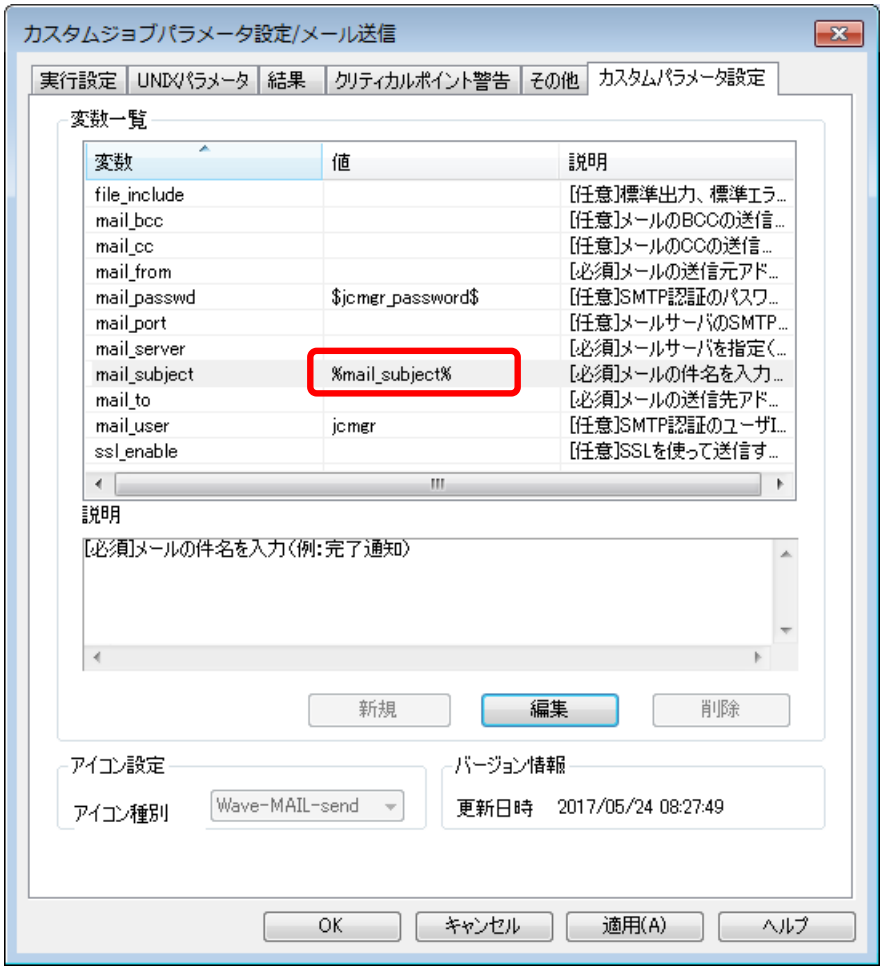


図1.13 環境変数の参照設定後のカスタムパラメータ設定画面

1.2.2.6.3.2. カスタムパラメータの一部に、環境変数の値を使用する場合

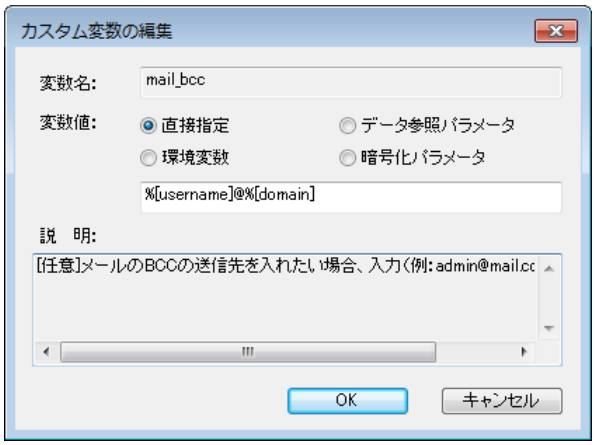


図1.14 ジョブネットワークの環境変数の設定例(部分使用の場合)

複数の環境変数や文字列と組み合わせる場合、対象のカスタムパラメータの入力ダイアログを開き、以下の手順で設定します。

■値を設定する方法

1. [変数値]の[直接指定]を選択します。

2. 入力欄をクリックし、値を入力して[OK]ボタンを押下します。

   環境変数名を指定する場合、以下の書式で記述します。

   - 書式

     ```
     %[<環境変数名>]
     ```

   - カスタムパラメータの入力例

     ```
     %[username]@%[domain]
     ```

■値を削除する方法

1. [変数値]の[直接指定]を選択し、入力欄を空欄にします。
2. 入力欄が空欄であることを確認して[OK]ボタンを押下します。

値を設定または削除した後、カスタムパラメータ設定画面で[適用]または[OK]ボタンを押下して設定を保存してください。

■以下のような値は、環境変数を参照するフォーマットになるため、通常の文字列としては使用できません。

- %aaa%

  先頭と末尾に「%」がある場合、その間にある「aaa」を環境変数として参照します。

- aaa%[bbb]ccc

  「%[」と「]」で囲まれた部分「bbb」を環境変数として参照します。

■値の先頭と末尾に「%」がある場合、%[<環境変数名>]のフォーマットは使用できません。

例：%%[ENV]%

%<環境変数名>%のフォーマットが優先されるため「%[ENV]」を環境変数として参照します。

■環境変数名の中に「]」が含まれる場合、%[<環境変数名>]のフォーマットは正常に使用できません。

例：%[ENV[1].PARAM1]

「ENV[1」を環境変数として参照します。

1.2.3. カスタムジョブテンプレートのカスタマイズ

Systemグループ配下の部品のテンプレートについては製造元から配布されたものであり、スクリプト、パラメータの変更ができません。変更する場合にはUserグループ配下に定義をコピーしてからご利用ください。

カスタムジョブテンプレートはJobCenter管理者ユーザのみ編集可能です。また、設定方法は拡張カスタムジョブ部品と同様です。設定方法については「1.2.2 カスタムパラメータの設定」を参照してください。

1.2.4. 表示アイコン

拡張カスタムジョブの表示アイコンのアップロードは、 スタンダードモードの場合は <スタンダードモード用基本操作ガイド>、 クラシックモードの場合は <クラシックモード用基本操作ガイド>を参照してください。

追加された表示アイコンは、CL/WinおよびCL/Webによる対象サーバへの次のログイン時から反映されます。

1.2.5. 注意事項

■拡張カスタムジョブを実行したとき、以下のようなカスタムパラメータの値は未指定として扱われます。

- カスタムパラメータの設定時に暗号化パラメータを展開するように設定しているが、拡張カスタムジョブの実行時に指定した暗号化パラメータが存在しなかった場合
- カスタムパラメータの設定時に環境変数の値を参照するように設定しているが、拡張カスタムジョブの実行時に指定した環境変数が存在しなかった場合

■R15.3以降では、カスタムパラメータの設定時に環境変数の値を参照するように設定したとき、参照した環境変数の値が「key」のように「$」で始まり「$」で終わる値となっている場合、自動的に暗号化パラメータの値に展開されます。

参照する環境変数の値としてこのような値が想定される場合、先に環境変数の値を暗号化パラメータとして登録し、環境変数を経由して暗号化パラメータの展開を行うようにしてください。

■1.9より前のバージョンの拡張カスタムジョブ部品を使用する場合、データ型に関わらず「1.2.2.5　文字列型の入力ダイアログ」となります。

2. メール送信部品

メール送信部品ついて説明します。

2.1. メール送信部品

2.1.1. 部品情報

■格納パス
/Root/System/Tools/Mail

■定義名
send_mail

■表示アイコン

2.1.2. スクリプト

メール本文として記載したい内容を記述します。

カスタムジョブパラメータでfile_includeを設定している場合、{{{file_include}}}を記述することで内容が展開されます。

また、環境変数についてもメール本文中に展開することができます。利用する場合は{{#getenv}}環境変数名{{/getenv}}と記述してください。

{{{file_include}}} で展開する標準出力・標準エラー出力が巨大なサイズになっている場合、SMTPサーバに設定されているメッセージサイズの上限を超える場合が有ります。

巨大な標準出力・標準エラー出力データの出力は、CL/Winでのトラッカ表示時に時間がかかる等の影響もあるため、単位ジョブスクリプト内で他のファイルに(アプリケーションやコマンドのログとして)リダイレクトするなどして巨大な出力とならないようご注意ください。

2.1.3. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	file_include	文字列	JNWの同一階層にある他の部品の標準出力、標準エラー出力をメールに取り込んで送信する場合に使用します。スクリプトに{{{file_include}}}と記載することで内容が展開されます。ジョブ名.oで標準出力、ジョブ名.eで標準エラー出力を詠み込みます。	JOB1.o 処理1.e
	mail_bcc	文字列	BCCとして送信アドレスを指定します。複数指定する場合は半角スペース、;(セミコロン)または,(カンマ)区切りで指定します。	admin@mail.local user1@mail.local, user2@mail.local
	mail_cc	文字列	CCとして送信アドレスを指定します。複数指定する場合は半角スペース、;(セミコロン)または,(カンマ)区切りで指定します。	admin@mail.local user1@mail.local, user2@mail.local
○	mail_from	文字列	Fromアドレスを指定します。	admin@mail.local
	mail_passwd	文字列	SMTPサーバに認証が必要な場合のパスワードを指定します。 値を直接入力せずに秘匿化したい場合は「1.2.2.6.2　暗号化パラメータの利用」を参照してください。	PASS

必須	変数	データ型	説明	設定例
	mail_port	数値	SMTPサーバのポート番号を指定します。空欄の場合は25を使用します。	25 465
○	mail_server	文字列	SMTPサーバ名を指定します。	smtp.server.local 127.0.0.1
○	mail_subject	文字列	メールの件名を指定します。	テスト 処理結果通知
○	mail_to	文字列	TOとして送信アドレスを指定します。複数指定する場合は半角スペース、;(セミコロン)または,(カンマ)区切りで指定します。	admin@mail.local user1@mail.local, user2@mail.local
	mail_user	文字列	SMTPサーバに認証が必要な場合のユーザ名を指定します。 値を直接入力せずに秘匿化したい場合は「1.2.2.6.2　暗号化パラメータの利用」を参照してください。	USER
	ssl_enable	リスト	SMTPサーバとの通信にSSLによる暗号化が必要な場合はtrueを指定します。空欄またはtrue以外を指定した場合はfalseとして扱います。なお対応しているプロトコルはSMTPsになります。STARTTLSには対応しておりません。	true false
	tls_version	リスト	SMTPSを利用したメール送信時のTLSバージョンを指定します。パラメータ省略時はTLS1.2での通信を行います。指定可能な値は1.0/1.1/1.2/1.3です。なお、1.0および1.1は非推奨です。セキュリティリスクを受容してでも継続利用せざるを得ない場合にのみ指定してください。	1.2
	transfer_encoding	リスト	メール本文のエンコード方式を '8bit' または 'base64'　で指定します。空欄または'8bit'を指定した場合、メール本文はUTF-8でエンコードしたメッセージを送信します。'base64'を指定した場合、メール本文はUTF-8でエンコードした文字列をbase64変換したメッセージを送信します。	8bit base64

2.1.4. 実行結果

メール本文として送信した内容が出力結果に表示されます。異常が発生した場合もエラーメッセージが出力結果に表示されます。

2.1.5. 主要メッセージ

メッセージ	内容
parameter [mail_XXX] is required.	必須パラメータが指定されていない場合に出力されます。
XXX: Not number	ポート番号の指定で数字以外を指定した場合に出力されます。
XXX: No such file or directory	file_includeの指定でファイルを読み込むことができなかった場合に出力されます。

メッセージ	内容
host or service not provided, or not known	メールサーバの指定が間違っている場合に出力されます。
connection refused	メールサーバの指定が間違っている、もしくはポート番号の指定が間違っている場合に出力されます。
timeout	メールサーバの指定が間違っている、もしくはメールサーバが応答しない場合に出力されます。
wrong version number	ssl_enableの指定で、設定が不要なメールサーバに対して設定した場合に出力されることがあります。

2.1.6. 注意事項

■アップロードした部品(Systemグループ配下にある部品)は、スクリプト(メール本文)を編集することができません。Userグループ配下に必要となる定型文の数だけコピーした後、スクリプト編集してご利用ください。

■SMTP認証ではplainとloginのみをサポートしています。

3. HTTP接続部品

HTTP接続部品ついて説明します。

HTTP接続部品は、Get、Put、Post、Delete、Head および Patch の各HTTPメソッドに対応するテンプレート群で構成される機能です。 各HTTP接続部品は、ユーザ認証やProxy設定等の機能を持ち、任意のサービスへのRESTリクエスト発行に利用することができます。

3.1. HTTP接続部品共通仕様

各HTTP接続部品で共通の設定方法、およびパラメータについて説明します。

各HTTP接続部品の設定は、部品そのものに対する制御パラメータと、HTTPリクエストに含めるパラメータによって設定方法が異なります。　部品の制御パラメータはカスタムパラメータ設定画面から、HTTPリクエストパラメータは部品のスクリプト画面から設定します。

3.1.1. リクエスト種別(HTTPメソッド)

HTTP接続部品のリクエスト種別(HTTPメソッド)は、対応する部品ごとに固有です。　HTTPサーバに送信したいリクエスト(HTTPメソッド)に合わせて、 Get、Put、Post、Delete、Head および Patch の各HTTPメソッドに対応するテンプレートを選択し、　制御パラメータおよびHTTPリクエストパラメータの設定を行って下さい。

3.1.2. 制御パラメータ(カスタムパラメータ)一覧

各HTTP接続部品で設定可能なカスタムパラメータは以下のとおりです。 各パラメータの詳細は、「3.1.2.1 基本パラメータ」 ～ 「3.1.3 HTTPリクエストパラメータ」で説明します。

必須	変数	データ型	説明	設定例
○	url	文字列	接続先のURL	http://www.nec.com/
	expected_status	文字列	正常終了とみなすHTTPステータスコード	2xx 302
	retry_count	数値	サーバ接続エラー時のリトライ回数	5
	retry_interval	数値	サーバ接続エラー時リトライの実行間隔 (秒)	3
	tls_version	リスト	HTTPS接続時のTLSバージョン	1.2
	request_body_file	文字列	リクエストボディ格納ファイルパス	/var/app/request.json
	response_header_file	文字列	レスポンスヘッダ情報格納ファイルパス	/var/app/response_header.txt
	response_body_file	文字列	レスポンスボディ格納ファイルパス	/var/app/output.json
	response_body_stdout	数値	レスポンスボディの標準出力への出力制限 (bytes)	1024
	proxy_server	文字列	プロクシサーバ	http://proxy.example.com:8080
	proxy_username	文字列	プロクシサーバの認証ユーザ名	johndoe
	proxy_password	文字列	プロクシサーバの認証パスワード	some-secret-password
	authenticate	リスト	HTTP接続先の認証方式	oauth2
	auth_username	文字列	HTTP接続先の認証ユーザ名	johndoe
	auth_password	文字列	HTTP接続先の認証パスワード	some-secret-password
	auth_token	文字列	認証トークン	some-authenticate-token
	oauth_realm	文字列	OAuth認証のrealm	some-oAuth-realm
	oauth_consumer_key	文字列	OAuth認証のコンシューマキー	some-oauth-consumer-key
	oauth_consumer_secret	文字列	OAuth認証のコンシューマキーシークレット	some-oAuth-consumer-key-sercret
	oauth_grant_type	リスト	OAuth2認証フロー種別	client_credentials

必須	変数	データ型	説明	設定例
	oauth_token_endpoint	文字列	OAuth2認証エンドポイント	https:// api.example.com/ oauth2/endpoints/ authorize

3.1.2.1. 基本パラメータ

HTTP接続部品の基本的動作を制御するパラメータについて説明します。

■url

接続先のURLを指定します。本パラメータの指定は必須であり、以下の形式で指定する必要があります。

http(s)://<host>[:<port>]/<path>

クエリパラメータはURLではなく、スクリプトで指定します。詳細は「3.1.3.1　URLクエリパラメータ」を参照してください。

■retry_count

サーバ接続エラー時のリトライ回数を指定します。retry_interval　と同時に指定し、いずれかのパラメータが省略されている場合はリトライ動作を行いません。

ネットワーク到達不可の場合、コネクション確立エラーの場合、および　HTTPサーバからのレスポンスコードが 503 の場合、 本パラメータで指定した回数までHTTPリクエストの発行を再実行します。

■retry_interval

サーバ接続エラー時のリトライ間隔を秒単位で指定します。retry_count　と同時に指定し、いずれかのパラメータが省略されている場合はリトライ動作を行いません。

retry_count によるHTTPリクエスト再発行を、本パラメータで指定した秒数の間隔を空けてから行います。ただし、HTTPサーバからのレスポンスコードが 503 でリトライを行う場合で、 レスポンスヘッダに Retry-After が設定されている場合は Retry-After の指定が優先されます。

■expected_status

正常終了とみなすHTTPステータスを指定します。

HTTPステータスは ' '(半角スペース)、','(カンマ)、';'(セミコロン) で区切って複数のステータスを指定可能です。 また、任意の10進数1文字のワイルドカードとして 'x'(小文字エックス) を指定することが可能です。

expected_status の設定例は次のようになります。

200 および 201 を正常終了とみなす場合: '200 201'

200番台を正常終了とみなす場合: '2xx'

■tls_version

HTTPS接続時のTLSバージョンを指定します。パラメータ省略時は TLS 1.2 での通信を行います。

指定可能な値は 1.0 / 1.1 / 1.2 です。なお、1.0 および 1.1 は非推奨です。セキュリティリスクを受容してでも継続利用せざるを得ない場合にのみ指定してください。

3.1.2.2. HTTP Proxyパラメータ

HTTP Proxy 経由で接続する場合に指定するパラメータについて説明します。

■proxy_server

ProxyサーバのURLを指定します。パラメータ省略時は、Proxyサーバ経由の接続を行いません。以下の形式で指定する必要があります。

http(s)://<host>[:<port>]

■proxy_username

Proxyサーバの認証ユーザ名を指定します。認証なしProxyサーバを利用する場合は、本パラメータの設定は必要ありません。

■proxy_password

Proxyサーバの認証パスワードを指定します。認証なしProxyサーバを利用する場合は、本パラメータの指定は必要ありません。

3.1.2.3. HTTP認証パラメータ

HTTP 接続先との認証を指定するパラメータについて説明します。

■authenticate

HTTP認証の方式を指定します。パラメータ省略時はHTTP認証処理は行いません。 指定可能な認証方式は、以下のとおりです。

- basic
- digest
- bearer
- oauth
- oauth2

■auth_username

認証ユーザ名を指定します。認証方式が basic, digest の場合は指定が必須です。また、認証方式が oauth2 かつ 認証フロー種別が password の場合は指定が必須です。

■auth_password

認証パスワードを指定します。認証方式が basic, digest 場合は指定が必須です。また、認証方式が oauth2 かつ 認証フロー種別が password の場合は指定が必須です。

■auth_token

認証トークンを指定します。認証方式が bearer の場合は指定が必須です。

■oauth_realm

OAuth認証のrealmを指定します。パラメータ省略時はrealmを空文字列として認証処理を行います。

■oauth_consumer_key

OAuth認証のコンシューマキーを指定します。認証方式が oauth, oauth2 の場合は指定が必須です。

■oauth_consumer_secret

OAuth認証のコンシューマキーシークレットを指定します。認証方式が oauth, oauth2 の場合は指定が必須です。

■oauth_grant_type

OAuth2認証フロー種別を指定します。認証方式が oauth2 の場合は指定が必須です。 指定可能なフロー種別は、以下のとおりです。

- password
- client_credentials

■oauth_token_endpoint

OAuth2トークンエンドポイントのURLを指定します。認証方式が oauth2 の場合は指定が必須です。

HTTP接続部品は、トークンエンドポイントから取得するトークン情報がJSON形式であり、アクセストークンは access_token というメンバー名で参照できることを前提として動作します。

3.1.2.4. 入出力パラメータ

HTTP接続部品の入力または出力に関わるパラメータについて説明します。

■request_body_file

HTTPリクエストのメッセージボディとして送信するファイルを絶対パスで指定します。 パラメータ省略時は、メッセージボディは設定されません。

■response_header_file

HTTPレスポンスのヘッダ情報を保存するファイルを絶対パスで指定します。 パラメータ省略時は、ヘッダ情報のファイルへの出力は行いません。 また、指定ファイルが存在していた場合は上書きします。 ヘッダ情報は、以下の形式で指定ファイルに保存されます。

■response_body_file

HTTPレスポンスのメッセージボディを保存するファイルを絶対パスで指定します。 パラメータ省略時は、メッセージボディのファイルへの出力は行いません。 また、指定ファイルが存在していた場合は上書きします。

■response_body_stdout

HTTPレスポンスのメッセージボディを標準出力に出力する最大量(bytes)を指定します。 パラメータ省略時は、メッセージボディの標準出力への出力は行いません。

3.1.3. HTTPリクエストパラメータ

HTTPリクエストパラメータの設定方法を説明します。

本項目における HTTPリクエストパラメータ は、以下の2種類を指します。

■URLクエリパラメータ

■HTTPヘッダパラメータ

3.1.3.1. URLクエリパラメータ

URLクエリパラメータは、HTTPサーバに情報を送るために、以下のように **変数名=値** の形式でURL末尾に付け加える情報です。

```
http://api.example.com/foo/var.cgi?param1=value1&param2=value2
```

HTTP接続部品でURLクエリパラメータを付与する場合は、スクリプト編集画面に一行一件で **変数名=値** の形式で記述します。

```
param1 = value1
param2 = value2
```

パラメータ名および値は、それぞれの前後空白文字を取り除いた文字列が設定されます。

パラメータ名および値は、HTTPリクエスト発行時にURLエンコードされます。スクリプト編集画面での入力は、URLエンコードされていない値を記述して下さい。

同名パラメータが複数回記述された場合、最後に設定されたパラメータ値が設定されます。

3.1.3.2. HTTPヘッダパラメータ

HTTPヘッダパラメータは、HTTPサーバに情報を送るために、HTTPリクエストヘッダに以下のように **変数名:値** の形式で設定される情報です。

```
param1: value1
param2: value2
```

HTTP接続部品は、標準で以下のHTTPヘッダを設定します。**"**で囲む表記は、固定の値が設定されることを示します。

ヘッダ名	値
Host	接続先のホスト名およびポート番号を設定します。
User-Agent	'JobCenter HTTP Connection Job'
Accept	'*/*'
Content-Length	request_body_file　を指定している場合、当該ファイルのサイズを自動的に設定します。未指定の場合はこのヘッダは設定されません。
Authorization	authorization が basic / digest / bearer の場合、認証情報を設定します。

これらのHTTPヘッダを変更、または新しいHTTPヘッダを追加する場合は、スクリプト編集画面に一行一件で **Header = 変数名 : 値** の形式で記述します。

```
Header = param1 : value1
Header = param2 : value2
```

パラメータ名および値は、それぞれの前後空白文字を取り除いた文字列が設定されます。

同名パラメータが複数回記述された場合、最後に設定されたパラメータ値が設定されます。

3.1.3.3. HTTPリクエストパラメータの環境変数展開

HTTPリクエストパラメータには環境変数を利用することができます。利用方法はスクリプト編集画面に「{{{環境変数名}}}」という記法を用いることで環境変数値の展開ができます。

3.1.3.3.1. HTTPリクエストパラメータの環境変数展開の利用例

環境変数とスクリプト編集画面が以下の設定がされている場合の例を記載します。

環境変数名	環境変数値
request_id	req-99999
HeaderAccept	Header = Accept : application/json

```
{{{HeaderAccept}}}
request_id={{{request_id}}}
```

設定されている環境変数を参照し展開するため、スクリプト編集画面で以下を記載したことと同じ意味になります。

```
Header = Accept : application/json
request_id=req-99999
```

スクリプト編集画面に記載した値が環境変数に存在しなかった場合は空文字列に置き換わります。

{ および } を環境変数展開と認識させないようにするエスケープ記法は用意しないため、パラメータ自体に { や } を入れたい場合には、環境変数の値に { や } を指定する必要があります。

3.1.4. 実行結果

HTTP接続部品の標準出力は、以下の形式で出力されます。**<>**で囲む表記は、実行時の値が反映されます。

```
--- General -------------------------------------
Request URL: <リクエスト先URL>
Request Method: <リクエストメソッド>
Status Code: <レスポンスコード>
--- HTTP response headers ----------------------
<ヘッダ名1>: <ヘッダ値1>
<ヘッダ名2>: <ヘッダ値2>
   : (設定されたヘッダを全て表示)
--- HTTP response body   -----------------------
<レスポンスボディーの内容>
```

response_body_stdout パラメータが未指定、または値が0の場合、HTTP response body 行以降は出力されません。

レスポンスボディーの内容は、response_body_stdout　パラメータで指定したbyte数以内で表示されます。

レスポンスボディーの内容は、文字コード変換の処理は行われませんので、マルチバイト文字を含む内容は文字化けが発生する場合があります。

レスポンスボディーの内容は、制御文字を'?'(クエスチョン)に置換した内容で表示されます。対象となる制御文字は次のとおりです。

0x00 (NUL)	0x08 (BS)	0x12 (DC2)	0x1A (SUB)
0x01 (SOH)	0x09 (TAB)	0x13 (DC3)	0x1B (ESC)
0x02 (STX)	0x0B (VT)	0x14 (DC4)	0x1C (FS)
0x03 (ETX)	0x0C (FF)	0x15 (NAK)	0x1D (GS)
0x04 (EOT)	0x0E (SO)	0x16 (SYN)	0x1E (RS)
0x05 (ENQ)	0x0F (SI)	0x17 (ETB)	0x1F (US)
0x06 (ACK)	0x10 (DLE)	0x18 (CAN)	0x7F (DEL)
0x07 (BEL)	0x11 (DC1)	0x19 (EM)	

3.1.5. 主要メッセージ

メッセージ	内容
Network is unreachable	接続先のHTTPサーバ名、もしくはポート番号の指定が誤っている場合に出力されます。 retry_count および retry_interval のパラメータを指定している場合、再送処理が行われます。
Service Temporarily Unavailable	接続先HTTPサーバから、503エラーが返却された場合に出力されます。
invalid "url" parameter.	url パラメータが指定されていない場合に出力されます。
Unsupported grant_type	認証方式に oauth2 を指定している時、oauth_grant_type パラメータにHTTP接続部品が対応していないOAuth2認証フロー種別を指定した場合に出力されます。
Failed to get access token	認証方式に oauth2 を指定している時、OAuth2アクセストークンが取得できなかった場合に出力されます。

3.1.6. 注意事項

■アップロードした部品(Systemグループ配下にある部品)は、スクリプト(HTTPリクエストパラメータ)を編集することができません。 定型のパラメータを設定した部品を作成する場合は、Userグループ配下に必要となる数だけコピーした後、スクリプト編集してご利用ください。

■Webサーバーからの応答が60秒以内に返ってこなかった場合、タイムアウトが発生し、HTTP接続部品が異常終了します。

タイムアウトまでの時間(60秒)は固定値で、変更することはできません。

3.2. HTTP-GET部品

3.2.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpGet

■表示アイコン

HTTP-GET 部品の詳細は、「3.1 HTTP接続部品共通仕様」を参照してください。

3.3. HTTP-PUT部品

3.3.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpPut

■表示アイコン

HTTP-PUT 部品の詳細は、「3.1 HTTP接続部品共通仕様」を参照してください。

3.4. HTTP-POST部品

3.4.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpPost

■表示アイコン

HTTP-POST 部品の詳細は、「3.1 HTTP接続部品共通仕様」を参照してください。

3.5. HTTP-DELETE部品

3.5.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpDelete

■表示アイコン

HTTP-DELETE 部品の詳細は、「3.1 HTTP接続部品共通仕様」を参照してください。

3.6. HTTP-HEAD部品

3.6.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpHead

■表示アイコン

HTTP-HEAD 部品の詳細は、「3.1 HTTP接続部品共通仕様」を参照してください。

3.7. HTTP-PATCH部品

3.7.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpPatch

■表示アイコン

HTTP-PATCH 部品の詳細は、「3.1 HTTP接続部品共通仕様」を参照してください。

4. Amazon Web Services 連携部品

Amazon Web Services(以下、AWSと記します)連携部品について説明します。

4.1. 共通仕様

各AWS連携部品で共通の仕様について説明します。

4.1.1. スクリプトの記述

AWS連携部品では部品のスクリプトに部品実行時に使用するパラメータなどの入力データや、Mustache言語で部品の実行結果の出力テンプレートを指定できる部品が存在します。部品毎にスクリプトで指定できる内容は異なりますので、スクリプトに何を指定できるかは各部品のスクリプトの説明を参照してください。

出力テンプレートのみを指定する部品の場合は出力テンプレートの内容をそのままスクリプトに記載します。また入力データと出力テンプレートの両方をスクリプトで指定できるような部品については以下のように「---in」、「---out」でセクションを区切ってそれぞれの内容を記述します。

```
---in
{
    "Arguments": {
    }
}

---out
{{#JobCenterResult}}{{#StartJobRunResponse}}JOBRUNID={{JobRunId}}{{/StartJobRunResponse}}
{{#GetJobRunResponse}}{{#JobRun}}JOBRUNSTATE={{JobRunState}}{{/JobRun}}{{/GetJobRunResponse}}{{/
JobCenterResult}}

  ========== error messages ==========
{{#JobCenterErrorMessageList}}{{{.}}}{{/JobCenterErrorMessageList}}
```

この場合、「---in」で始まる行以降に記載したデータが部品の実行時の入力データとして扱われます。また「---out」で始まる行以降に記載した内容が、部品の実行結果の出力テンプレートとして扱われます。なお「---in」より前の行に記述した内容は無視されます。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2. EC2

Amazon Elastic Compute Cloud(以下、EC2と記します)の部品について説明します。

4.2.1. EC2部品共通のカスタムパラメータ

EC2部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.2.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、ec2.ap-northeast-1.amazonaws.comが指定されています。	ec2.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.2.2. インスタンス起動

4.2.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
StartInstances

■表示アイコン

4.2.2.2. 部品説明

インスタンスを起動します。

対象となるインスタンスは、インスタンスIDでの指定の他、タグで検索することができます。

4.2.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
△	InstanceId	文字列	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはtagの指定が必要です。	i-123456ab i-123456ab; i-45678cd
△	tag	文字列	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはInstanceIdの指定が必要です。	AutoShutDown=ON
	interval	数値	EC2インスタンスの状態確認を行う間隔を秒単位で指定します。デフォルト値として5が指定されています。また、空欄の場合は30として扱います。	10
	timeout	数値	EC2インスタンスの起動完了の待ち合わせを行う際のタイムアウト時間を分単位で指定し	30

必須	変数	データ型	説明	設定例
			ます。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	
	StatusCheck	リスト	EC2インスタンスがrunningの状態になりステータスチェックが完了するまでジョブの終了を待ち受ける場合はtrueを指定します。	true false

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.2.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

インスタンス開始対象となるインスタンスを特定するために、DescribeInstancesを実行し、DescribeInstancesResponseに格納されます。インスタンス開始結果は、StartInstancesResponseに格納されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.2.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.2.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	StartInstances
EC2	DescribeInstanceStatus

4.2.2.8. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■InstanceId、tagを両方を指定した場合はANDとして解釈されます。

4.2.3. インスタンス停止

4.2.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
StopInstances

■表示アイコン

4.2.3.2. 部品説明

インスタンスを停止します。

対象となるインスタンスは、インスタンスIDでの指定の他、タグで検索することができます。

4.2.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
△	InstanceId	文字列	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはtagの指定が必要です。	i-123456ab i-123456ab; i-45678cd
△	tag	文字列	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはInstanceIdの指定が必要です。	AutoShutDown=ON
	interval	数値	EC2インスタンスの状態確認を行う間隔を秒単位で指定します。デフォルト値として5が指定されています。また、空欄の場合は30として扱います。	10
	timeout	数値	EC2インスタンスの停止完了の待ち合わせを行う際のタイムアウト時間を分単位で指定し	30

必須	変数	データ型	説明	設定例
			ます。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.2.3.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

インスタンス停止対象となるインスタンスを特定するために、DescribeInstancesを実行し、DescribeInstancesResponseに格納されます。インスタンス停止結果は、StopInstancesResponseに格納されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.2.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.2.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	StopInstances
EC2	DescribeInstanceStatus

4.2.3.8. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■InstanceId、tagを両方を指定した場合はANDとして解釈されます。

4.2.4. インスタンス情報取得

4.2.4.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
DescribeInstances

■表示アイコン

4.2.4.2. 部品説明

インスタンスの情報取得を行います。

インスタンスIDまたはタグを指定することで、取得対象となるインスタンスを絞り込むことができます。

4.2.4.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.4.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
△	InstanceId	文字列	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはtagの指定が必要です。	i-123456ab i-123456ab; i-45678cd
△	tag	文字列	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはInstanceIdの指定が必要です。	AutoShutDown=ON
	noTag	文字列	EC2インスタンスのタグを指定します。検索結果を、指定したタグを持たない、または持っていても値が空であるインスタンスに絞り込みます。複数入力する場合セミコロン ；区切りで指定します。	Owner AutoShutDown; AutoBoot

4.2.4.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeInstancesResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.2.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.2.4.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances

4.2.4.8. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■InstanceId、tagを両方指定した場合はANDとして解釈されます。

■noTagでタグを複数指定した場合はORとして解釈されます。

4.2.5. インスタンスプロパティ変更

4.2.5.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
ModifyInstanceAttribute

■表示アイコン

4.2.5.2. 部品説明

インスタンスのインスタンスタイプを変更します。

変更対象のインスタンスは、インスタンスIDで指定することができます。

4.2.5.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.5.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	InstanceId	文字列	EC2インスタンスIDを指定します。	i-123456ab
○	InstanceType.Value	文字列	EC2インスタンスのタイプを指定します。	t2.micro

4.2.5.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、ModifyInstanceAttributeResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.2.5.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.5.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	ModifyInstanceAttribute

4.2.5.8. 注意事項

特になし

4.2.6. AMI作成

4.2.6.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
CreateImage

■表示アイコン

4.2.6.2. 部品説明

AMIを作成します。

対象となるインスタンスは、インスタンスIDで指定することができます。

4.2.6.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.6.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	InstanceId	文字列	EC2インスタンスIDを指定します。	i-123456ab
○	Name	文字列	AMIの名前を指定します。マルチバイト文字は指定できません。	Test AMI

4.2.6.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CreateImageResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.2.6.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.6.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	CreateImage

4.2.6.8. 注意事項

特になし

4.2.7. 世代管理付きAMI作成

4.2.7.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
CreateImageWithGeneration

■表示アイコン

4.2.7.2. 部品説明

AMIを作成します。

対象となるインスタンスは、インスタンスIDでの指定の他、タグで検索することができます。

この部品で作成されたAMIには世代管理用のタグが付与され、世代数の古いAMIを削除することができます。

4.2.7.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.7.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
△	InstanceId	文字列	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはInstancetagの指定が必要です。	i-123456ab i-123456ab; i-45678cd
△	Instancetag	文字列	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはInstanceIdの指定が必要です。	AMIBackup=Yes
	Description	文字列	AMIの概要説明を指定します。マルチバイト文字は指定できません。 本パラメータは以下のマクロに対応しています。パラメータ中にマクロ文字列を記述した場合、対応する値に置換されます。	CreateImage %Y/%m/%d %H:%M:%S(%i)

必須	変数	データ型	説明	設定例
			%Y：西暦(4桁) %m：月(2桁) %D：日(2桁) %H：時間(2桁) %M：分(2桁) %S：秒(2桁) %i：作成元となるECインスタンスのインスタンスID %%：%自身(エスケープ)	
○	generation	数値	AMIの世代数を指定します。デフォルト値として1が指定されています。 0を指定した場合は古い世代のAMIの削除を行いません。	5
	managementName	文字列	世代管理を区別する名前をUTF-8換算で63バイト以内で指定します。 このパラメータは同じインスタンスに対して複数のパターンで世代管理を別々に行う場合に指定します。	daily 月次バックアップ
○	timeout	数値	AMIの作成完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。	30
○	NoReboot	リスト	AMIを作成する際に作成元のEC2インスタンスを再起動するかどうかを指定します。インスタンスを再起動しない場合はtrue、再起動する場合はfalseを指定します。デフォルト値としてtrueが指定されています。	true

4.2.7.5. 部品詳細

本部品は大きく分けて5つのステップの動作があります。

1. AMI作成対象となるインスタンス一覧の取得
2. AMIの作成要求
3. AMI作成の待ち合わせ
4. 作成要求成功したAMIへ世代管理を行うためのタグの付与
5. AMIの作成完了待ち合わせ
6. 旧世代のAMIの削除

本部品の動作の詳細を説明します。

1. InstanceIdパラメータおよびInstancetagパラメータを用いて、AMI作成対象となるインスタンスの一覧を取得します。

インスタンスの一覧を取得できなかった場合や、インスタンスがひとつも見つからなかった場合、部品はコード1で終了します。

2. 取得したインスタンス一覧に対してAMIの作成要求を行います。

   ここで、AMI名は"YYYYMMDDhhmmss(<インスタンスID>)"という名前が自動的に指定されます。

   また、AMIの説明はDescriptionパラメータで指定された文字列(マクロ文字列を記述していた場合はマクロ展開後の文字列)が設定されます。

   作成要求に成功したAMIの一覧が次の処理の対象となります。

   インスタンス一覧のすべてのインスタンスに対してAMIの作成要求が失敗した場合、部品はコード1で終了します。

3. 作成要求に成功したAMIの作成完了を待ち合わせます。

   約5秒間隔でAMIの状態を取得し、作成完了したかどうかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■AMI一覧のすべてのAMIの状態を取得したとき

   4.以降の処理を行います。

   ■待ち合わせ開始時刻から60秒が経過したとき

   状態を取得したAMIがひとつもなかった場合、部品はコード1で終了します。

   ■AMIの状態取得が"notfound"以外で失敗したとき

   部品はコード1で終了します。

   待ち合わせ終了時に、状態を取得したAMIの一覧が次の処理の対象となります。

4. 作成要求に成功したAMI一覧の各AMIに以下のタグを付与します。

   ■JCSourceInstanceId=<作成元となったインスタンスのインスタンスID>

   このタグは世代管理に必要となります。

   ■JCGenerationManagement

   - managementNameパラメータを指定した場合

     JCGenerationManagement=Yes:<managementNameパラメータで指定した値>

   - managementNameパラメータを指定しない場合

     JCGenerationManagement=Yes

   このタグは世代管理に必要となります。

   ■Name=<作成元となったインスタンスのNameタグの値>

   タグの付与に成功したAMIの一覧が次の処理の対象となります。

   AMI一覧のすべてのAMIに対してタグの付与に失敗した場合、部品はコード1で終了します。

5. タグの付与に成功したAMI一覧に対して、作成完了を待ち合わせます。

約30秒間隔でAMIの状態を取得し、作成完了したかどうかをチェックします。

以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

■AMI一覧のすべてのAMIの状態が"available"または"failed"のいずれかとなったとき

すべてのAMIの状態が"failed"となった場合、部品はコード1で終了します。

■待ち合わせ開始時刻からtimeoutパラメータで指定した時間が経過したとき

状態が"available"であるAMIがひとつもなかった場合、部品はコード1で終了します。

■AMIの状態取得に失敗した場合

この場合、部品はコード1で終了します。

待ち合わせ終了時に、状態が"available"となったAMIの一覧が次の処理の対象となります。

6. generationパラメータが0よりも大きい場合、AMIの世代管理、すなわち古い世代のAMIの削除を行います。

状態が"available"となったAMI一覧の各AMIに対して、以下の処理を行います。

■以下のタグをすべて持つAMIの一覧を取得します。

JCSourceInstanceId=<作成元となったインスタンスのインスタンスID>

- managementNameパラメータを指定した場合

  JCGenerationManagement=Yes:<managementNameパラメータで指定した値>

- managementNameパラメータを指定しない場合

  JCGenerationManagement=Yes

■上記のAMI一覧から、状態が"available"であるもののみを取り出し、AMI名を利用して作成日時順にソートします。

■上記のソートされたAMI一覧のうち、作成日時が新しいものからgenerationパラメータで指定した数だけを残し、他のAMIを削除します。

また、削除したAMIに紐づいたスナップショットを削除します。

7. AMI作成対象となったインスタンスすべてに対して、AMIの作成処理および世代管理処理が成功した場合、部品はコード0で終了します。

そうでない場合、部品はコード1で終了します。

4.2.7.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、オブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.2.7.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.7.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	CreateImage
EC2	DescribeImages
EC2	CreateTags
EC2	DeregisterImage
EC2	DeleteSnapshot

4.2.7.9. 注意事項

■タグの付与に失敗したAMI、および作成に失敗して状態が"failed"となったAMIは、世代管理の対象にはならず残存し続けます。そのようなAMIは手動で削除する必要があります。

4.2.8. タグ作成

4.2.8.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
CreateTags

■表示アイコン

4.2.8.2. 部品説明

EC2の各リソースのタグ作成を行います。

対象となるリソースはリソースIDで指定します。

4.2.8.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.8.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	ResourceId	文字列	リソースIDを指定します。複数入力する場合セミコロン；区切りを指定します。	snap-123456ab i-123456ab; snap-45678cd
○	Tag	文字列	作成するタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン ； 区切りで指定します。	Name=Snapshot1

4.2.8.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CreateTagsResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.2.8.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	Tagの指定方法が間違っている場合に表示されます。

4.2.8.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	CreateTags

4.2.8.8. 注意事項

特になし

4.2.9. スナップショット作成

4.2.9.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
CreateSnapshot

■表示アイコン

4.2.9.2. 部品説明

EBSボリュームのスナップショットを作成します。

対象となるEBSボリュームは、ボリュームIDでの指定の他、インスタンスID、インスタンスタグ、ボリュームタグで検索することができます。

世代数の古いスナップショットを削除することができます。

4.2.9.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.9.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	Description	文字列	スナップショットの概要説明を指定します。マルチバイト文字は指定できません。 本パラメータは以下のマクロに対応しています。パラメータ中にマクロ文字列を記述した場合、対応する値に置換されます。 %Y：西暦(4桁) %m：月(2桁) %D：日(2桁) %H：時間(2桁) %M：分(2桁) %S：秒(2桁)	CreateSnapshot %Y/%m/%d %H:%M:%S(%i, %v)

必須	変数	データ型	説明	設定例
			%i：作成元となるECインスタンスのインスタンスID(ボリュームがインスタンスにアタッチされていない場合など、インスタンスIDが取得できない場合は空文字列) %v：作成元となるボリュームのボリュームID %%：%自身(エスケープ)	
△	VolumeId	文字列	ボリュームIDを指定します。複数入力する場合セミコロン；区切りを指定します。 InstanceId、Instancetag、VolumeId、Volumetagのいずれか1つを指定する必要があります。	vol-123456ab vol-123456ab;vol-45678cd
△	Volumetag	文字列	ボリュームのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン ； 区切りで指定します。 InstanceId、Instancetag、VolumeId、Volumetagのいずれか1つを指定する必要があります。	Name=Vol1
△	InstanceId	文字列	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りを指定します。 InstanceId、Instancetag、VolumeId、Volumetagのいずれか1つを指定する必要があります。	i-123456ab i-123456ab;i-45678cd
△	Instancetag	文字列	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りを指定します。 InstanceId、Instancetag、VolumeId、Volumetagのいずれか1つを指定する必要があります。	AutoShutDown=ON
○	generation	数値	スナップショットの世代数を指定します。デフォルト値として1が指定されています。 0を指定した場合は古い世代のスナップショットの削除を行いません。	5
○	timeout	数値	スナップショットの作成完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。	30
○	checkInterval	数値	スナップショットの作成完了の待ち合わせを行う際のステータスチェックの間隔を秒単位で指定します。デフォルト値として30が指定されています。	60

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.2.9.5. 部品詳細

本部品は大きく分けて5つのステップの動作があります。

1. スナップショット作成対象となるEBSボリューム一覧の取得

2. スナップショットの作成要求

3. スナップショット作成の待ち合わせ

4. 作成要求成功したスナップショットへタグの付与

5. スナップショットの作成完了待ち合わせ

6. 旧世代のスナップショットの削除

本部品の動作の詳細を説明します。

1. InstanceIdパラメータ、Instancetagパラメータ、VolumeIdパラメータ、Volumetagパラメータのうち、指定されたものを用いて、スナップショット作成対象となるEBSボリュームの一覧を取得します。

   EBSボリュームの一覧を取得できなかった場合や、EBSボリュームがひとつも見つからなかった場合、部品はコード1で終了します。

2. 取得したEBSボリューム一覧に対してスナップショットの作成要求を行います。

   スナップショットの説明はDescriptionパラメータで指定された文字列(マクロ文字列を記述していた場合はマクロ展開後の文字列)が設定されます。

   作成要求が成功したスナップショットの一覧が次の処理の対象となります。

   EBSボリューム一覧のすべてのEBSボリュームに対してスナップショットの作成要求が失敗した場合、部品はコード1で終了します。

3. 作成要求が成功したスナップショットに対して、作成完了を待ち合わせます。

   約5秒間隔でスナップショットの状態を取得し、作成完了したかどうかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■スナップショット一覧のすべてのスナップショットの状態を取得したとき

     4.以降の処理を行います。

   ■待ち合わせ開始時刻から60秒が経過したとき

     状態を取得したスナップショットがひとつもなかった場合、部品はコード1で終了します。

   ■スナップショットの状態取得が"notfound"以外で失敗したとき

     部品はコード1で終了します。

   待ち合わせ終了時に、状態を取得したスナップショットの一覧が次の処理の対象となります。

4. 作成要求が成功したスナップショットに以下のタグを付与します。

   ■Name=<作成元となったEBSボリュームのNameタグの値>

   ■InstanceName=<作成元となったEBSボリュームをアタッチしているインスタンスのNameタグの値>

■上記タグは部品利用者がスナップショットの識別を行いやすくする目的で付与しています。そのため、付与されたタグを後から編集、削除しても問題ありません。

■EBSボリュームにNameタグが存在しない、またはNameタグの値が空である場合、スナップショットにNameタグは付与されません。

■EBSボリュームをアタッチしているインスタンスが存在しない、インスタンスにNameタグが存在しない、または値が空である場合、スナップショットにInstanceNameタグは付与されません。

タグの付与に失敗した場合、以下の警告メッセージを標準出力に出力します。

```
Warning: CreateTags request failure. target snapshotId=<スナップショットID>
```

なお、タグの付与に失敗しても部品は実行を継続します。また、タグの付与の結果は部品の終了コードに影響しません。

5. 作成要求が成功したスナップショット一覧に対して、作成完了を待ち合わせます。

   checkIntervalパラメータで指定した秒間隔でスナップショットの状態を取得し、作成完了したかどうかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■スナップショット一覧のすべてのスナップショットの状態が"completed"または"error"のいずれかとなったとき

   すべてのスナップショットの状態が"error"となった場合、部品はコード1で終了します。

   ■待ち合わせ開始時刻からtimeoutパラメータで指定した時間が経過したとき

   状態が"completed"であるスナップショットがひとつもなかった場合、部品はコード1で終了します。

   ■スナップショットの状態取得に失敗した場合

   この場合、部品はコード1で終了します。

   待ち合わせ終了時に、状態が"completed"となったスナップショットの一覧が次の処理の対象となります。

6. generationパラメータが0よりも大きい場合、スナップショットの世代管理、すなわち古い世代のスナップショットの削除を行います。

   状態が"completed"となったスナップショット一覧の各スナップショットに対して、以下の処理を行います。

   ■そのスナップショットの作成元となったEBSボリュームから作られたスナップショット一覧を取得します。

   ■上記のスナップショット一覧から、状態が"completed"であるもののみを取り出し、作成日時順にソートします。

   ■上記のソートされたスナップショット一覧のうち、作成日時が新しいものからgenerationパラメータで指定した数だけを残し、他のスナップショットを削除します。

   削除しようとしたスナップショットがAMIに使用されている場合、そのスナップショットは削除できません。また、その理由によりスナップショットの削除が行えなかった場合、本部品はそれをエラー扱いしません。

7. スナップショット作成対象となったEBSボリュームすべてに対して、スナップショットの作成処理および世代管理処理が成功した場合、部品はコード0で終了します。

   そうでない場合、部品はコード1で終了します。

4.2.9.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

スナップショットの作成対象となるボリュームを特定するために、DescribeInstancesまたはDescribeVolumesを実行しますが、それぞれDescribeInstancesResponse、DescribeVolumesResponseに格納されます。スナップショット作成結果は、JobCenterResultというオブジェクト配下に、VolumeId、CreateSnapshotResponse、DescribeSnapshotsResponseとして格納されます。ローテーション結果として、削除したスナップショットが存在する場合には、jc_deletedという変数がtrueに設定されます。また削除ができなかった場合にはjc_delete_error_messageにエラーメッセージが保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.2.9.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.2.9.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	DescribeVolumes
EC2	CreateSnapshot
EC2	CreateTags
EC2	DescribeSnapshots
EC2	DeleteSnapshot

4.2.9.9. 注意事項

■InstanceId、Instancetag、VolumeId、Volumetagをいずれか1つではなく、複数指定した場合はInstanceId、Instancetagの設定が優先されます。Id、tagを両方を指定した場合はANDとして解釈されます。

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.2.10. スナップショット削除

4.2.10.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
DeleteSnapshot

■表示アイコン

4.2.10.2. 部品説明

スナップショットを削除します。

削除対象となるスナップショットは、スナップショットIDで指定することができます。

4.2.10.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.10.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	SnapshotId	文字列	スナップショットIDを指定します。	snap-123456ab

4.2.10.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DeleteSnapshotResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.2.10.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.10.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DeleteSnapshot

4.2.10.8. 注意事項

特になし

4.2.11. スナップショット情報取得

4.2.11.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
DescribeSnapshots

■表示アイコン

4.2.11.2. 部品説明

スナップショットの情報取得を行います。

スナップショットIDまたはタグを指定することで、取得対象となるスナップショットを絞り込むことができます。

4.2.11.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.11.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	SnapshotId	文字列	スナップショットIDを指定します。複数入力する場合セミコロン；区切りを指定します。	snap-123456ab snap-123456ab;snap-45678cd
	tag	文字列	スナップショットのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。	Name=Snapshot1

4.2.11.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeSnapshotsResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.2.11.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.2.11.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeSnapshots

4.2.11.8. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■Id、tagを両方を指定した場合はANDとして解釈されます。

4.2.12. スナップショットコピー

4.2.12.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
CopySnapshot

■表示アイコン

4.2.12.2. 部品説明

スナップショットをコピーします。

コピー対象となるスナップショットは、スナップショットIDでの指定の他、タグで検索することができます。

コピー先のリージョンは、エンドポイントが表すリージョンとなります。

4.2.12.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.12.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	DryRun	リスト	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	Description	文字列	スナップショットの概要説明を指定します。マルチバイト文字は指定できません。	CopySnapshot
△	SourceSnapshotId	文字列	コピー元のスナップショットIDを指定します。	snap-123456ab
△	SourceSnapshottag	文字列	コピー元のスナップショットのタグを指定します。 このパラメータを指定した場合、SourceRegionパラメータの値はエンドポイントが表すリージョンを指定する必要があります。	
○	SourceRegion	文字列	コピー元のリージョンを指定します。 SourceSnapshottagパラメータを指定した場合、このパラメータの値はエンドポイントが表すリージョンを指定する必要があります。	ap-northeast-1

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.2.12.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CopySnapshotResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.2.12.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.12.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeSnapshots
EC2	CopySnapshot

4.2.12.8. 注意事項

特になし

4.2.13. EBSボリュームサイズ拡張

4.2.13.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
ExtendVolumeSize

■表示アイコン

4.2.13.2. 部品説明

EBSボリュームのサイズを拡張します。

対象となるEBSボリュームは、ボリュームIDで指定します。

4.2.13.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.13.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	VolumeId	文字列	EBSボリュームIDを指定します。	vol-123456ab
○	Size	数値	EBSボリュームの拡張後のサイズをGiB単位で指定します。	200

4.2.13.5. 部品詳細

本部品は大きく分けて2つのステップの動作があります。

1. サイズ拡張対象となるEBSボリュームの情報取得
2. ボリュームサイズの拡張

本部品の動作の詳細を説明します。

1. VolumeIdパラメータを用いて、DescribeVolumes　APIを発行しサイズ拡張対象となるEBSボリュームの情報を取得します。

   情報を取得できなかった場合、部品はコード1で終了します。

2. ModifyVolume APIを発行しボリュームに対してサイズの拡張要求を行います。

   サイズの拡張要求に失敗した場合、またはSizeパラメータで指定した値が現在のボリュームサイズ以下であった場合、部品はコード1で終了します。

4.2.13.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、オブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.2.13.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.13.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeVolumes
EC2	ModifyVolume

4.2.13.9. 注意事項

特になし

4.2.14. EBSボリュームタイプ変更

4.2.14.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
ChangeVolumeType

■表示アイコン

4.2.14.2. 部品説明

EBSボリュームのタイプを変更します。

対象となるEBSボリュームは、ボリュームIDで指定します。

4.2.14.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.2.14.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	VolumeId	文字列	EBSボリュームIDを指定します。	vol-123456ab
○	VolumeType	リスト	EBSボリュームの変更後のタイプを指定します。 ただし、io1 を指定することはできません。	standard, gp2, sc1, st1

4.2.14.5. 部品詳細

本部品は大きく分けて2つのステップの動作があります。

1. タイプ変更対象となるEBSボリュームの情報取得
2. ボリュームタイプの変更

本部品の動作の詳細を説明します。

1. VolumeIdパラメータを用いて、DescribeVolumes　APIを発行しタイプ変更対象となるEBSボリュームの情報を取得します。

   情報を取得できなかった場合、部品はコード1で終了します。

2. ModifyVolume APIを発行しボリュームに対してタイプの変更要求を行います。

   タイプの変更要求に失敗した場合、または現在のボリュームタイプとVolumeTypeパラメータで指定した値が同じであった場合、部品はコード1で終了します。

4.2.14.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、オブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.2.14.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.14.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeVolumes
EC2	ModifyVolume

4.2.14.9. 注意事項

本部品はボリュームタイプ io1 への変更をサポートしていません。

4.3. S3

Amazon Simple Storage Service(以下、S3と記します)の部品について説明します。

4.3.1. S3部品共通のカスタムパラメータ

S3部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.3.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、s3-ap-northeast-1.amazonaws.comが指定されています。	s3-ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.3.2. S3部品共通仕様

S3部品に共通の仕様について説明します。

4.3.2.1. ワイルドカード設定

S3部品の一部パラメータではワイルドカードを利用することができます。

ワイルドカードとして利用できる文字は以下になります。

利用可能文字	説明
*	0文字以上の任意の文字列と一致するパターン
?	任意の1文字と一致するパターン

4.3.3. ファイルダウンロード

4.3.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/SimpleStorageService

■定義名
GetObject

■表示アイコン

4.3.3.2. 部品説明

S3上のファイルをジョブ実行マシン上にダウンロードします。

4.3.3.3. スクリプト

指定する必要はありません。

4.3.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	bucket	文字列	S3のバケット名を指定します。	bucket-directory
△	object	文字列	ダウンロードするS3のオブジェクト名を指定します。 このパラメータまたはprefixパラメータのどちらか一方が必須です。 wild_cardパラメータがtrueの場合、ワイルドカードを使用できます。その場合、パターンと一致したファイルを全てダウンロードします。	test.txt test?.txt
	file_path	文字列	objectパラメータ指定時のオブジェクトの格納先を指定します。 ■ワイルドカードが無効の場合 ダウンロードしたオブジェクトを格納するファイルパスを絶対パスで指定します。指定しない場合は標準出力に内容を表示します。 ■ワイルドカードが有効の場合 ダウンロードしたオブジェクトを格納するディレクトリを絶対パスで指定します。指定しない場合は標準出力に内容を表示します。	C:\test.txt /tmp/test.txt
△	prefix	文字列	ダウンロードするS3フォルダ名を絶対パスで指定します。	test/log/

必須	変数	データ型	説明	設定例
			指定したフォルダ直下のファイルをダウンロードします。サブフォルダのダウンロードには対応していません。 このパラメータまたはprefixパラメータのどちらか一方が必須です。	
	dir_path	文字列	prefixパラメータ指定時のダウンロード先ディレクトリ名を絶対パスで指定します。指定しない場合は標準出力に内容を表示します。 prefixパラメータが入力されていない場合、このパラメータの値は利用されません。	C:\test\log
	wild_card	リスト	objectパラメータでワイルドカードを有効にするかどうかを指定します。有効にする場合はtrueを指定します。無効にする場合は空欄またはfalseを指定します。	true false
	exclude_file	文字列	ダウンロード対象から除外するファイルを指定します。 このパラメータはwild_cardパラメータの設定値に関係なくワイルドカードが有効となります。また、複数指定する場合は,(カンマ)で区切って記述します。	tmp?.txt tmp*.txt tmp1.txt,tmp2.txt

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

prefixパラメータとobjectパラメータの両方に値が入力されている場合、prefixパラメータが優先されます。

4.3.3.5. 実行結果

ダウンロードした結果を出力します。

4.3.3.5.1. 実行例

S3上のbucketに以下のファイルが登録されている場合の例を説明します。

```
+--var/
  +----log/
    +------error1.txt
    +------error2.txt
    +------log2017.tar.gz
    +------log20181031.log
    +------log20181101.log
    +------log20181102.log
    +------log20181103.log
    +------log20181104.log
    +------log20181105.log
```

4.3.3.5.1.1. prefixパラメータを用いた複数ファイルのダウンロード例

var/log配下の「error1.txt」、「error2.txt」、「log2017.tar.gz」ファイル以外のすべてのファイルをダウンロードし、/tmp/log配下のディレクトリに格納したい場合、 カスタムパラメータを以下の値に設定することで目的のファイルを取得できます。

変数	設定値
bucket	<bucket名>
object	<空白>
file_path	<空白>
prefix	var/log
dir_path	/tmp/log
wild_card	<空白>
exclude_file	error?.txt,log2017.tar.gz

このカスタムパラメータで実行した結果、ダウンロードされるファイルは以下になります。

■log20181031.log

■log20181101.log

■log20181102.log

■log20181103.log

■log20181104.log

■log20181105.log

4.3.3.5.1.2. objectパラメータを用いた複数ファイルのダウンロード例

var/log配下のファイル名が「log2018」に前方一致するすべてのファイルをダウンロードし、/tmp/log配下のディレクトリに格納したい場合、 カスタムパラメータを以下の値に設定することで目的のファイルを取得できます。

変数	設定値
bucket	< bucket名>
object	var/log/log2018*
file_path	/tmp/log
prefix	<空白>
dir_path	<空白>
wild_card	true
exclude_file	<空白>

このカスタムパラメータで実行した結果、ダウンロードされるファイルは以下になります。

■log20181031.log

■log20181101.log

■log20181102.log

■log20181103.log

■log20181104.log

■log20181105.log

4.3.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Not a file:XXX.	格納先がファイルではない場合に出力されます。
No such target object.	ダウンロード対象のオブジェクトがひとつも存在しない場合に出力します。

4.3.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
S3	ListBucket
S3	GetObject

4.3.3.8. 注意事項

■取得したオブジェクトの内容を標準出力に表示する場合、JobCenterの言語設定と一致していないと文字化けする可能性があります。

4.3.4. ファイルアップロード

4.3.4.1. 部品情報

■格納パス
/Root/System/AmazonWebService/SimpleStorageService

■定義名
PutObject

■表示アイコン

4.3.4.2. 部品説明

ジョブ実行マシン上のファイルをS3上にアップロードします。

4.3.4.3. スクリプト

指定する必要はありません。

4.3.4.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	bucket	文字列	S3のバケット名を指定します。	bucket-directory
○	object	文字列	S3のオブジェクト名を指定します。 ■file_pathパラメータにファイルを指定した場合(ワイルドカードが無効) S3のオブジェクト名にはobjectパラメータがそのまま利用されます。 ■file_pathパラメータにファイルを指定した場合(ワイルドカードが有効) S3のオブジェクト名にはディレクトリ名にobjectパラメータ、ファイル名にfile_pathパラメータで対象となるファイルが利用されます。 ■file_pathパラメータにディレクトリを指定した場合 S3のオブジェクト名にはディレクトリ名にobjectパラメータ、ファイル名にfile_pathパラメータで対象となるファイルが利用されます。	test.txt var/log
○	file_path	文字列	バケットに格納するジョブ実行マシン上のファイルまたはディレクトリを絶対パスで指定します。 ■ファイルを指定した場合(ワイルドカードが無効)	C:\test.txt /tmp/test.txt

必須	変数	データ型	説明	設定例
			アップロードするファイルパスを絶対パスで指定します。 ■ファイルを指定した場合(ワイルドカードが有効) ファイルパスにワイルドカードを含めて絶対パスで指定します。 条件に一致するファイルが対象となります。 ■ディレクトリを指定した場合 アップロードするディレクトリを絶対パスで指定します。 ディレクトリ直下のファイルが対象となります。サブディレクトリ配下のファイルは対象となりません。	
	wild_card	リスト	file_pathパラメータでワイルドカードを有効にするかどうかを指定します。 有効にする場合はtrueを指定します。無効にする場合は空欄またはfalseを指定します。	true false
	exclude_file	文字列	アップロード対象から除外するファイルを指定します。 このパラメータはwild_cardパラメータの設定値に関係なくワイルドカードが有効となります。また、複数指定する場合は,(カンマ)で区切って記述します。	tmp?.txt tmp*.txt

4.3.4.5. 実行結果

アップロードした結果を出力します。

4.3.4.5.1. 実行例

JobCenterサーバ上に以下のファイルが保存されている場合の例を説明します。

```
+--/tmp/
  +----log/
    +------error1.txt
    +------error2.txt
    +------log2017.tar.gz
    +------log20181031.log
    +------log20181101.log
    +------log20181102.log
    +------log20181103.log
    +------log20181104.log
    +------log20181105.log
```

4.3.4.5.1.1. file_pathパラメータにディレクトリを指定した場合の複数ファイルのアップロード例

/tmp/log配下の「error1.txt」、「error2.txt」、「log2017.tar.gz」ファイル以外のすべてのファイルをアップロードし、S3上のbucketのvar/log配下のフォルダに格納したい場合、カスタムパラメータを以下の値に設定することで目的のファイルをアップロードできます。

変数	設定値
bucket	<bucket名>
object	var/log
file_path	/tmp/log
wild_card	<空白>
exclude_file	error?.txt,log2017.tar.gz

このカスタムパラメータで実行した結果、アップロードされるファイルは以下になります。

■log20181031.log

■log20181101.log

■log20181102.log

■log20181103.log

■log20181104.log

■log20181105.log

4.3.4.5.1.2. file_pathパラメータにワイルドカードを指定した場合の複数ファイルのアップロード例

/tmp/log配下のファイル名が「log2018」に前方一致するすべてのファイルをアップロードし、S3上のbucketのvar/log配下のフォルダに格納したい場合、カスタムパラメータを以下の値に設定することで目的のファイルをアップロードできます。

変数	設定値
bucket	<bucket名>
object	var/log/
file_path	/tmp/log/log2018*
wild_card	true
exclude_file	<空白>

このカスタムパラメータで実行した結果、アップロードされるファイルは以下になります。

■log20181031.log

■log20181101.log

■log20181102.log

■log20181103.log

■log20181104.log

■log20181105.log

4.3.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
No such file or directory.	格納するファイルがファイルまたはディレクトリではない、もしくは存在しないパスである場合に出力されます。
No such target object.	アップロード対象のオブジェクトがひとつも存在しない場合に出力します。

4.3.4.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
S3	PutObject

4.3.4.8. 注意事項

特になし

4.3.5. ファイル削除

4.3.5.1. 部品情報

■格納パス
/Root/System/AmazonWebService/SimpleStorageService

■定義名
DeleteObject

■表示アイコン

4.3.5.2. 部品説明

S3上のファイルを削除します。

4.3.5.3. スクリプト

指定する必要はありません。

4.3.5.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	bucket	文字列	S3のバケット名を指定します。	bucket-directory
○	object	文字列	S3のオブジェクト名を指定します。	test.txt

4.3.5.5. 実行結果

ファイル削除した結果を出力します。

4.3.5.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.3.5.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
S3	DeleteObject

4.3.5.8. 注意事項

■存在しないオブジェクトを削除した場合、ジョブとしては正常終了します。

4.3.6. S3部品共通の注意事項

■UNIX環境では各ジョブはroot権限で実行されるため、S3部品のファイルアクセス権限はrootとなります。

4.4. ELB

Amazon Elastic Load Balancing(以下、ELBと記します)の部品について説明します。

4.4.1. ELB部品共通のカスタムパラメータ

ELB部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.4.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、elasticloadbalancing.ap-northeast-1.amazonaws.comが指定されています。	elasticloadbalancing.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.4.2. クラシックロードバランサーへのインスタンス登録

4.4.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticLoadBalancing

■定義名
RegisterInstancesWithLoadBalancer

■表示アイコン

4.4.2.2. 部品説明

クラシックロードバランサーに指定したインスタンスを登録します。

4.4.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.4.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	InstanceId	文字列	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りを指定します。	i-123456ab i-123456ab; i-45678cd
○	LoadBalancerName	文字列	ロードバランサー名を指定します。	my-http-loadbalancer

4.4.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、RegisterInstancesWithLoadBalancerResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.4.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.4.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
ELB	RegisterInstancesWithLoadBalancer

4.4.2.8. 注意事項

特になし

4.4.3. クラシックロードバランサーからのインスタンス登録解除

4.4.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticLoadBalancing

■定義名
DeregisterInstancesFromLoadBalancer

■表示アイコン

4.4.3.2. 部品説明

クラシックロードバランサーから指定したインスタンスを登録解除します。

4.4.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.4.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	InstanceId	文字列	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りを指定します。	i-123456ab i-123456ab;i-45678cd
○	LoadBalancerName	文字列	ロードバランサー名を指定します。	my-http-loadbalancer

4.4.3.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DeregisterInstancesFromLoadBalancerResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.4.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.4.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
ELB	DeregisterInstancesFromLoadBalancer

4.4.3.8. 注意事項

特になし

4.5. CF

AWS CloudFormation(以下、CFと記します)の部品について説明します。

4.5.1. CF部品共通のカスタムパラメータ

CF部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.5.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、cloudformation.ap-northeast-1.amazonaws.comが指定されています。	cloudformation.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.5.2. スタック作成

4.5.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudFormation

■定義名
CreateStack

■表示アイコン

4.5.2.2. 部品説明

指定したテンプレートからスタックを作成します。

4.5.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.5.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	StackName	文字列	作成するスタック名を指定します。	
○	TemplateURL	文字列	スタック作成に使用するテンプレートファイルがあるURLを指定します。 URLはAmazon　S3バケット上にあるファイルを指定する必要があります。	https://s3-ap-northeast-1.amazonaws.com/cf-templates-xxxxxxxxxxxx-ap-northeast-1/cf_template.json

4.5.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CreateStackResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.5.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.5.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudFormation	CreateStack
CloudFormation	DescribeStacks

4.5.2.8. 注意事項

特になし

4.5.3. スタック削除

4.5.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudFormation

■定義名
DeleteStack

■表示アイコン

4.5.3.2. 部品説明

指定したスタックを削除します。

4.5.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.5.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	StackName	文字列	削除するスタック名を指定します。	

4.5.3.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DeleteStackResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.5.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.5.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudFormation	DeleteStack

4.5.3.8. 注意事項

特になし

4.5.4. スタック情報取得

4.5.4.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudFormation

■定義名
DescribeStacks

■表示アイコン

4.5.4.2. 部品説明

指定したスタックの情報取得を行います。

4.5.4.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.5.4.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	StackName	文字列	情報取得するスタック名を指定します。	

4.5.4.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeStacksResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.5.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.5.4.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudFormation	DescribeStacks

4.5.4.8. 注意事項

特になし

4.6. CW

Amazon CloudWatch(以下、CWと記します)の部品について説明します。

4.6.1. CW部品共通のカスタムパラメータ

CW部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.6.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、monitoring.ap-northeast-1.amazonaws.comが指定されています。	monitoring.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.6.2. アラーム情報取得

4.6.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudWatch

■定義名
DescribeAlarms

■表示アイコン

4.6.2.2. 部品説明

アラームの情報取得を行います。

アラーム名を直接指定またはアラーム名のプレフィックスを指定することで、取得対象となるアラームを絞り込むことができます。

4.6.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.6.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AlarmName	文字列	アラーム名を指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータとAlarmNamePrefixパラメータを同時に指定した場合、AlarmNamePrefixパラメータは無視されます。	alarm001 instanceA_CPUUtil; instanceB_CPUUtil
	AlarmNamePrefix	文字列	アラーム名を前方一致で検索します。 このパラメータとAlarmNameパラメータを同時に指定した場合、このパラメータは無視されます。	instanceA_

4.6.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeAlarmsResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.6.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.6.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudWatch	DescribeAlarms

4.6.2.8. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.6.3. アラーム状態待ち合わせ

4.6.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudWatch

■定義名
WaitAlarmState

■表示アイコン

4.6.3.2. 部品説明

アラームの状態がOKとなるまで待ち合わせます。

対象となるアラームは、アラーム名での指定の他、アラーム名のプレフィックスで検索することができます。

4.6.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.6.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
△	AlarmName	文字列	アラーム名を指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータとAlarmNamePrefixパラメータを同時に指定した場合、AlarmNamePrefixパラメータは無視されます。 このパラメータまたはAlarmNamePrefixの指定が必要です。	alarm001 instanceA_CPUUtil;instanceB_CPUUtil
△	AlarmNamePrefix	文字列	アラーム名を前方一致で検索します。 このパラメータとAlarmNameパラメータを同時に指定した場合、このパラメータは無視されます。 このパラメータまたはAlarmNameの指定が必要です。	instanceA_
	interval	数値	アラームの状態確認を行う間隔を秒単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	10
	timeout	数値	アラームの状態がOKとなるまで待ち合わせを行う際のタイムアウト時間を分単位で指定し	60

必須	変数	データ型	説明	設定例
			ます。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。	

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.6.3.5. 部品詳細

本部品は大きく分けて2つのステップの動作があります。

1. 状態チェック対象となるアラーム一覧の取得
2. アラームの状態がOKになるまで待ち合わせ

本部品の動作の詳細を説明します。

1. AlarmNameパラメータ、AlarmNamePrefixパラメータのうち、指定されたものを用いて、状態チェック対象となるアラームの一覧を取得します。

   アラームの一覧を取得できなかった場合や、アラームがひとつも見つからなかった場合、部品はコード1で終了します。

2. 前ステップで取得したアラーム一覧に対して、状態がOKになるのを待ち合わせます。

   intervalパラメータで指定した秒間隔でアラームの状態を取得し、状態がOKかどうかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■アラーム一覧のすべてのアラームの状態が"OK"となったとき

   この場合、部品はコード0で終了します。

   ■待ち合わせ開始時刻から状態が"OK"とならないアラームが存在したままtimeoutパラメータで指定した時間が経過したとき

   部品はコード1で終了します。

   ■アラームの状態取得に失敗した場合

   この場合、部品はコード1で終了します。

4.6.3.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

状態チェック対象となるアラームを特定するために、DescribeAlarmsを実行し、DescribeAlarmsResponseに格納されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.6.3.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.6.3.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudWatch	DescribeAlarms

4.6.3.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.7. RDS

Amazon Relational Database Service(以下、RDSと記します)の部品について説明します。

4.7.1. RDS部品共通のカスタムパラメータ

RDS部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.7.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、rds.ap-northeast-1.amazonaws.comが指定されています。	rds.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.7.2. インスタンス起動

4.7.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/RelationalDatabaseService

■定義名
StartDBInstance

■表示アイコン

4.7.2.2. 部品説明

DBインスタンスを起動します。

対象とするDBインスタンスはDBインスタンスの名前で指定します。

4.7.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.7.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	DBInstanceIdentifier	文字列	起動させたいDBインスタンスの名前を指定します。	db001
	interval	数値	DBインスタンスのステータス確認を行う間隔を秒単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。	10
	timeout	数値	DBインスタンスが利用可能になるまでの待ち合わせを行うタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	30

4.7.2.5. 部品詳細

本部品は大きく分けて3つのステップの動作があります。

1. DBインスタンスのステータスの確認

2. DBインスタンスの起動要求

3. DBインスタンスが利用可能になるまで待ち合わせ

本部品の動作の詳細を説明します。

1. DBInstanceIdentifierで指定されたDBインスタンスについて、DescribeDBInstances　APIでステータスを取得します。

   DBインスタンスのステータスが取得できなかった場合は、部品はコード1で終了します。

   DBインスタンスのステータスが既にavailable(利用可能)である場合、、部品はコード0で終了します。

   DBインスタンスのステータスが既にstarting(開始中)である場合、DBインスタンスがavailable(利用可能)になるまで待ち合わせます。

   DBインスタンスのステータスが他のステータスである場合、DBインスタンスの起動要求を行います。

2. DBInstanceIdentifierで指定されたDBインスタンスについて、StartDBInstance　APIで起動の要求を行います。

   StartDBInstance APIによる起動の要求が失敗した場合、部品はコード1で終了します。

3. DBInstanceIdentifierで指定されたDBインスタンスについて、ステータスがavailable(利用可能)になるまで待ち合わせます。

   intervalパラメータで指定した秒の間隔で、DescribeDBInstances　APIでDBインスタンスのステータスを取得し、available(利用可能)かどうかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■DBインスタンスのステータスがavailable(利用可能)となったとき

   この場合、部品はコード0で終了します。

   ■待ち合わせ開始時刻からtimeoutパラメータで指定した時間が経過したとき

   この場合、部品はコード1で終了します。

   ■DBインスタンスのステータスの取得に失敗したとき

   この場合、部品はコード1で終了します。

4.7.2.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下の通りオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

■DBインスタンスのステータスの確認

DescribeDBInstances APIの結果がBeforeDescribeDBInstancesResponseオブジェクトに保存されます。

■DBインスタンスの起動要求

StartDBInstance APIの結果がStartDBInstanceResponseオブジェクトに保存されます。

■DBインスタンスが利用可能になるまで待ち合わせ

待ち合わせ完了時点の最終的なDescribeDBInstances　APIの結果がDescribeDBInstancesResponseオブジェクトに保存されます。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには表示されない場合があります。

4.7.2.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is not a number.	数値を指定する必要があるパラメータに数値以外が指定された場合に出力されます。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のDescribeDBInstances　APIについては出力されず、待ち合わせ完了時点での最後のDescribeDBInstances　APIについてのみ出力されます。
Failed to XXX. Message：XXX Code：XXX	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.7.2.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
RDS	DescribeDBInstances
RDS	StartDBInstance

4.7.2.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには表示されない場合があります。

■部品実行時点でDBインスタンスのステータスがavailable(利用可能)の場合、その時点で部品は正常終了します。

■部品実行時点でDBインスタンスのステータスがstarting(開始中)の場合、DBインスタンスの起動要求は行わずにステータスがavailable(利用可能)まで待ち合わせを行います。

■待ち合わせの処理では、intervalで指定された間隔でDBインスタンスのステータスのチェックを行っています。

この間隔の間にステータスがavailable(利用可能)になり、さらに他のステータスに遷移した場合、ステータスのチェックの時点でステータスがavailable(利用可能)ではないため、起動部品は実行中のままとなります。

例えばDBインスタンスがavailable(利用可能)となった直後に停止操作が行われた場合、別途操作がない限りステータスがavailable(利用可能)に戻らないため、起動部品はtimeoutで指定した時間で打ち切られるまで実行中となる可能性があります。

■CL/Webを用いてInternet ExplorerからRDS部品を含むジョブネットワークのフローを参照した場合、RDS部品の表示アイコンは黒色で表示されます。

4.7.3. インスタンス停止

4.7.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/RelationalDatabaseService

■定義名
StopDBInstance

■表示アイコン

4.7.3.2. 部品説明

DBインスタンスを停止します。

対象とするDBインスタンスはDBインスタンスの名前で指定します。

4.7.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.7.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	DBInstanceIdentifier	文字列	停止させたいDBインスタンスの名前を指定します。	db001
	interval	数値	DBインスタンスのステータス確認を行う間隔を秒単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。	10
	timeout	数値	DBインスタンスが停止になるまでの待ち合わせを行うタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	30
	createSnapshot	リスト	DBインスタンスの停止時にスナップショットを作成する場合はtrueを指定します。trueが指定された場合は「<DBインスタンスID>-<YYYYYMMDDHHmmSS>」をスナップショット名としてスナップショットを作成します。 このとき<YYYYMMDDHHmmSS>の時刻はUTC時刻となります。	true

必須	変数	データ型	説明	設定例
			また、空欄の場合やtrue以外の値が設定されている場合はスナップショットは作成しません。	

4.7.3.5. 部品詳細

本部品は大きく分けて3つのステップの動作があります。

1. DBインスタンスのステータスの確認
2. DBインスタンスの停止要求
3. DBインスタンスが停止になるまで待ち合わせ

本部品の動作の詳細を説明します。

1. DBInstanceIdentifierで指定されたDBインスタンスについて、DescribeDBInstances　APIでステータスを取得します。

   DBインスタンスのステータスが取得できなかった場合は、部品はコード1で終了します。

   DBインスタンスのステータスが既にstopped(停止)である場合、、部品はコード0で終了します。

   DBインスタンスのステータスが既にstopping(停止中)である場合、DBインスタンスがstopped(停止)になるまで待ち合わせます。

   DBインスタンスのステータスが他のステータスである場合、DBインスタンスの停止要求を行います。

2. DBInstanceIdentifierで指定されたDBインスタンスについて、StopDBInstance　APIで停止の要求を行います。

   StopDBInstance APIによる停止の要求が失敗した場合、部品はコード1で終了します。

3. DBInstanceIdentifierで指定されたDBインスタンスについて、ステータスがstopped(停止)になるまで待ち合わせます。

   intervalパラメータで指定した秒の間隔で、DescribeDBInstances　APIでDBインスタンスのステータスを取得し、stopped(停止)かどうかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■DBインスタンスのステータスがstopped(停止)となったとき

   この場合、部品はコード0で終了します。

   ■待ち合わせ開始時刻からtimeoutパラメータで指定した時間が経過したとき

   この場合、部品はコード1で終了します。

   ■DBインスタンスのステータスの取得に失敗したとき

   この場合、部品はコード1で終了します。

4.7.3.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下の通りオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

■DBインスタンスのステータスの確認

DescribeDBInstances APIの結果がBeforeDescribeDBInstancesResponseオブジェクトに保存されます。

■DBインスタンスの停止要求

StopDBInstance APIの結果がStopDBInstanceResponseオブジェクトに保存されます。

■DBインスタンスが停止になるまで待ち合わせ

待ち合わせ完了時点の最終的なDescribeDBInstances　APIの結果がDescribeDBInstancesResponseオブジェクトに保存されます。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎると表示されない場合があります。

4.7.3.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is not a number.	数値を指定する必要があるパラメータに数値以外が指定された場合に出力されます。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のDescribeDBInstances　APIについては出力されず、待ち合わせ完了時点での最後のDescribeDBInstances　APIについてのみ出力されます。
Failed to XXX. Message：XXX Code：XXX	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.7.3.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
RDS	DescribeDBInstances
RDS	StopDBInstance

4.7.3.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには表示されない場合があります。

■部品実行時点でDBインスタンスのステータスがstopped(停止)の場合、その時点で部品は正常終了します。

■部品実行時点でDBインスタンスのステータスがstopping(停止中)の場合、DBインスタンスの停止要求は行わずにステータスがstopped(停止)まで待ち合わせを行います。

■待ち合わせの処理では、intervalで指定された間隔でDBインスタンスのステータスのチェックを行っています。

この間隔の間にステータスがstopped(停止)になり、さらに他のステータスに遷移した場合、ステータスのチェックの時点でステータスがstopped(停止)ではないため、停止部品は実行中のままとなります。

例えばDBインスタンスがstopped(停止)となった直後に起動操作が行われた場合、別途操作がない限りステータスがstopped(停止)に戻らないため、停止部品はtimeoutで指定した時間で打ち切られるまで実行中となる可能性があります。

■AWS側の仕様としてDBインスタンスを停止した後、7日経過すると自動的に停止したインスタンスが起動されます。これについてはRDSの停止部品を用いてJobCenterからDBインスタンスを停止した場合も同様の動作となります。

■CL/Webを用いてInternet ExplorerからRDS部品を含むジョブネットワークのフローを参照した場合、RDS部品の表示アイコンは黒色で表示されます。

4.8. Glue

Amazon Glue(以下、Glueと記します)の部品について説明します。

4.8.1. Glue部品共通のカスタムパラメータ

Glue部品で共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.8.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、glue.ap-northeast-1.amazonaws.comが指定されています。	glue.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.8.2. ジョブ実行

4.8.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/Glue

■定義名
StartJob

■表示アイコン

4.8.2.2. 部品説明

Glueジョブを実行します。

実行するGlueジョブはGlueジョブの名前で指定します。

4.8.2.3. スクリプト

本部品のスクリプトでは部品実行時に使用する入力データと部品の実行結果の出力テンプレートを指定できます。

スクリプト中に入力データと出力テンプレートの両方を記載する場合の記述方法については「4.1.1　スクリプトの記述」を参照してください。

■入力データ

以下のようにjson形式でArguments配下に任意のパラメータを指定することで、Glueジョブの実行時のジョブパラメータとして渡すパラメータを指定できます。

```
{
  "Arguments": {
    "--job-param-name1": "value1",
    "--job-param-name2": "value2",
  },
}
```

また環境変数の値を利用したい場合、{{{環境変数名}}}の形式で記述することで環境変数が展開されます。
なお、対象の環境変数が存在しない場合は空文字として扱われます。

■出力テンプレート

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.8.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	JobName	文字列	実行したいGlueジョブの名前を指定します。	job1
	job-bookmark-option	リスト	ジョブ実行時にGlueジョブのブックマークを処理する方法を指定します。	disable enable pause
	enable-metrics	リスト	Glueジョブの実行時にCloudWatchメトリクスの作成を有効にする場合はtrueを指定します。	true
	enable-continuous-cloudwatch-log	リスト	Glueジョブのリアルタイムな継続的なロギングを有効にしてCloudWatchでApacheSparkジョブログを表示したい場合はtrueを指定します。	true false
	enable-continuous-log-filter	リスト	enable-continuous-cloudwatch-logがtrueの場合に継続的なロギングで標準フィルターを使用する場合はtrueを指定します。	true false
	enable-spark-ui	リスト	Apache Spark Web UIを使用する場合はtrueを指定します。本パラメータでtrueを指定する場合、spark-event-logs-pathも指定する必要があります。	true false
	spark-event-logs-path	文字列	enable-spark-uiがtrueの場合にGlueジョブのSparkイベントログを保存するためのAmazon S3のパスを指定します。	S3://bucket/prefix/object
	SecurityConfiguration	文字列	Glueジョブの実行時に使用するセキュリティ設定の名前を指定します。	SecurityConfig1
	extra-py-files	文字列	Glueジョブがスクリプトを実行する前にパスに追加するAmazon S3上のPythonモジュールを指定します。	S3://bucket/prefix/object
	extra-jars	文字列	Glueジョブがスクリプトを実行する前にパスに追加するAmazon S3上のJavaファイルを指定します。	S3://bucket/prefix/object
	extra-files	文字列	Glueジョブがスクリプトを実行する前に作業ディレクトリにコピーするAmazon S3上のファイルを指定します。	S3://bucket/prefix/object
	MaxCapacity	数値	Glueジョブの実行時に割り当てられるDPUの数を指定します。本パラメータはバージョン1.0以前でのGlueジョブにのみ有効です。また本パラメータをWorkerTypeと一緒に指定することはできません。 Python シェルジョブの場合、0.0625 または 1 が指定できます。 Apache Spark ETL ジョブの場合、2 から 100 の範囲の整数値が指定できます。	2
	WorkerType	リスト	Glueジョブの実行時に割り当てられるWrokerのタイプを指定します。本パラメータを指定する場	Standard G.1X

必須	変数	データ型	説明	設定例
			合、NumberOfWorkersも指定する必要があります。	G.2X
	NumberOfWorkers	数値	Glueジョブの実行時に割り当てられるWorkerの数を指定します。本パラメータを指定する場合、WorkerTypeも指定する必要があります。 Standardタイプの場合、2 から 299 の範囲の整数値が指定できます。 G.1Xタイプの場合、2 から 299 の範囲の整数値が指定できます。 G.2Xタイプの場合、2 から 149 の範囲の整数値が指定できます。	2
	Timeout	数値	Glueジョブのタイムアウト時間を指定します。	2880
	NotifyDelayAfter	数値	Glueジョブの実行が開始された後、実行遅延通知を送信するまでの待機時間を指定します。	10
	interval	数値	実行したGlueジョブの状態の確認を行う間隔を秒単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。	10

4.8.2.5. 部品詳細

本部品は大きく分けて2つのステップの動作があります。

1. Glueジョブの実行要求
2. Glueジョブが終了するで待ち合わせ

本部品の動作の詳細を説明します。

1. JobNameで指定されたGlueジョブについて、StartJobRun APIで実行の要求を行います。

   StartJobRun APIによる実行の要求が失敗した場合、部品はコード1で終了します。

2. 実行したGlueジョブについて、終了するまで待ち合わせます。

   intervalパラメータで指定した秒の間隔で、GetJobRun APIでGlueジョブのステータスを取得し、Glueジョブが終了したかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■GlueジョブのステータスがSUCCEEDEDとなったとき

   この場合、部品はコード0で終了します。

   ■GlueジョブのステータスがFAILED、ERROR、TIMEOUT、STOPPEDのいずれかとなったとき

   この場合、部品はコード1で終了します。

   ■Glueジョブのステータスがその他の不正なステータスとなったとき

   この場合、部品はコード1で終了します。

■GetJobRun APIによるGlueジョブのステータスの取得に失敗したとき

この場合、部品はコード1で終了します。

4.8.2.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下のとおりオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

```
{
  "JobCenterResult": {
    "GetJobRunResponse": {}
    "StartJobRunResponse": {}
  },
  "JobCenterErrorMessageList": []
}
```

■JobCenterResult

JobCenterResultオブジェクトには以下のとおり各APIの結果が保存されます。各APIの結果の詳細についてはAWS側のドキュメントを参照してください。

- StartJobRunResponse

  StartJobRunResponseにはStartJobRun APIの結果が保存されます。

- GetJobRunResponse

  GetJobRunResponseには待ち合わせ完了時点の最終的なGetJobRun APIの結果が保存されます。

■JobCenterErrorMessageList

JobCenterErrorMessageList配列にはエラーが発生した場合のメッセージが保存されます。保存されるエラーメッセージについては「4.8.2.7 主要メッセージ」を参照してください。

4.8.2.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is not a number.	数値を指定する必要があるパラメータに数値以外が指定された場合に出力されます。
parameter [XXX] is invalid. the parameter with dependency is not specified.	依存関係があるほかのパラメータが指定されていない場合に出力されます。
parameter [XXX] is mutually exclusive.	同時に指定できないパラメータが指定された場合に出力されます。
parameter [XXX] is invalid. The value is out of range.	パラメータに範囲外の値を指定した場合に出力されます。
input script must be json format and type Object.	スクリプトの入力データに誤りがある場合に出力されます。スクリプトの入力データはjson形式のオブジェクト形式である必要があります。

メッセージ	内容
GetJobRun request failure.	GetJobRun APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
StartJobRun request failure.	StartJobRun APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
JobRun XXX status check time out.	Glueジョブの完了待ちがタイムアウトした場合、JobCenterErrorMessageList配列に格納されます。
JobRunState is not SUCCEEDED.	GlueジョブのステータスがSUCCEEDED以外であった場合、JobCenterErrorMessageList配列に格納されます。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のGetJobRun APIについては出力されず、待ち合わせ完了時点での最後のGetJobRun APIについてのみ出力されます。
Failed to XXX. Message：XXX Code：XXX	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.8.2.8. 使用API

本部品が使用するAWSのAPIの一覧は以下のとおりです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
Glue	StartJobRun
Glue	GetJobRun

4.8.2.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには全てのデータが表示されず途中で切れる場合があります。

■スクリプト中の入力データのjsonで未対応のパラメータを指定された場合は無視します。

■スクリプト中の入力データのjsonで環境変数を使用する場合、json自体の中括弧と連続した場合に不具合が起きる可能性があります。

たとえば「{"KEY":{{{環境変数名}}}}」のように値を環境変数名で使用する場合、jsonの中括弧が環境変数展開用のものと誤認されるため正しく展開されない可能性があります。

このような場合「{"KEY":{{{環境変数名}}} }」のように中括弧の間にスペースや改行を入れることで回避できます。

■スクリプト中の入力データで指定するジョブパラメータの値としてstring以外の形式が指定された場合、暗黙でstringに変換して扱います。

■AWS側の制約として、実行したGlueジョブが終了した後、同時実行数にカウントされなくなるまでにラグが存在しています。このため、この間に同じGlueジョブを実行する部品を実行して同時実行数の上限に達した場合、StartJobRun APIがエラーとなり部品は終了コード1で終了します。

■オプションのパラメータを指定しない場合、Glueのジョブの定義で指定されたパラメータで実行されます。

- オプションを指定した場合は、指定したオプションのパラメータが優先されます。

- オプションを指定せずGlueのジョブの定義で指定されたパラメータで実行される場合、実行されたGlueのジョブの結果を参照したとき実際の動作に使用されたパラメータと表示が一致しない場合があります。（AWS側のStartJobRun APIの動作の仕様の問題となります）

4.9. Lambda

Amazon Lambda(以下、Lambdaと記します)の部品について説明します。

4.9.1. Lambda部品共通のカスタムパラメータ

Lambda部品で共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.9.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、lambda.ap-northeast-1.amazonaws.comが指定されています。	lambda.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.9.2. 関数実行

4.9.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/Lambda

■定義名
InvokeFunction

■表示アイコン

4.9.2.2. 部品説明

Lambda関数を実行します。

実行するLambda関数はLambda関数の名前で指定します。

4.9.2.3. スクリプト

本部品のスクリプトでは部品実行時に使用する入力データと部品の実行結果の出力テンプレートを指定できます。

スクリプト中に入力データと出力テンプレートの両方を記載する場合の記述方法については「4.1.1　スクリプトの記述」を参照してください。

■入力データ

以下のようにjson形式でEvent配下に任意の値を指定することで、Lambda関数呼び出し時のイベントとして渡す値を指定できます。

また同様にClientContext配下に任意の値を指定することで、Lambda関数呼び出し時のClientContextとして渡す値を指定できます。

```
{
  "Event": {
    "event_param": "event_value"
  },
  "ClientContext": {
    "client_context_param": "client_context_value"
  }
}
```

また環境変数の値を利用したい場合、{{{環境変数名}}}の形式で記述することで環境変数が展開されます。なお、対象の環境変数が存在しない場合は空文字として扱われます。

■出力テンプレート

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.9.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	FunctionName	文字列	実行したいLambda関数の名前を指定します。	function1
	LogType	リスト	Lambda関数の実行ログを取得したい場合、Tailを指定します。	None Tail
	Qualifier	文字列	実行したい関数のバージョン、またはエイリアスを指定したい場合に指定します。	1 alias1
	InvocationType	リスト	Lambda関数の呼び出し方式を指定します。 Eventを指定した場合、部品はLambda関数の終了を待ち合わせません。またこの場合、LogTypeをTailに設定していてもログは取得できません。 RequestResponseを指定した場合、部品はLambda関数の終了を待ち合わせます。 DryRunを指定した場合、Lambda関数の実際の処理は行われません。	Event RequestResponse DryRun

4.9.2.5. 部品詳細

本部品の動作の詳細を説明します。

1. FunctionNameで指定された関数について、Invoke APIで実行の要求を行います。

   Invoke APIによる実行の要求が失敗した場合、部品はコード1で終了します。

2. InvocationTypeで指定した値によって以下のとおりInvoke APIの動作が変わります。

   ■Eventの場合

   この場合、関数の実行を要求した時点でInvoke APIは終了します。

   ■RequestResponseの場合

   この場合、Invoke APIはLambda関数の実行が完了した後に終了します。

   ■DryRunの場合

   この場合、Invoke APIはLambda関数の実行を実際には行わないで終了します。

3. Invoke APIの結果からLambda関数の実行中にエラーが発生したかを確認します。

   Invoke APIの結果としてLambda関数の実行中に例外などのエラーが発生していた場合、部品はコード1で終了します。

   エラーが無い場合、部品はコード0で終了します。

4.9.2.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下のとおりオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

```
{
  "JobCenterResult": {
    "InvokeResponse": {
      "Header": {},
      "Body": {}
      "BodyString": ""
    }
  },
  "JobCenterErrorMessageList": []
}
```

■JobCenterResult

JobCenterResultオブジェクトには以下のとおり各APIの結果が保存されます。各APIの結果の詳細についてはAWS側のドキュメントを参照してください。

- InvokeResponse

  InvokeResponseオブジェクトにはInvoke APIの結果が保存されます。

  - Header

    HeaderオブジェクトにはAPIの結果のHTTPヘッダーが保存されます。ただし保存されるHTTPヘッダーはx-amzで始まるヘッダーのみとなります。

  - Body

    BodyにはAPIの結果のBodyが保存されます。

  - BodyString

    BodyStringにはBodyの内容をjson文字列にした値が保存されます。

■JobCenterErrorMessageList

JobCenterErrorMessageList配列にはエラーが発生した場合のメッセージが保存されます。保存されるエラーメッセージについては「4.9.2.7 主要メッセージ」を参照してください。

4.9.2.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is invalid. The value is out of range.	パラメータに範囲外の値を指定した場合に出力されます。
input script must be json format and type Object.	スクリプトの入力データに誤りがある場合に出力されます。スクリプトの入力データはjson形式のオブジェクト形式である必要があります。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のGetJobRun APIについては出力されず、待ち合わせ完了時点での最後のGetJobRun APIについてのみ出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

メッセージ	内容
Message：XXX Code：XXX	
POST XXXX wantread Failed to XXX. Message：wantread not expected Code：Internal error	APIの通信がソケットのタイムアウトにより終了した場合に出力されます。invoke　APIで発生した場合、Lambda関数の実行が続いている可能性があります。
Invoke request failure.	Invoke　APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
Error occurred during function execution.	関数の実行中に例外などのエラーが発生した場合、JobCenterErrorMessageList配列に格納されます。

4.9.2.8. 使用API

本部品が使用するAWSのAPIの一覧は以下のとおりです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
Lambda	Invoke

4.9.2.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには全てのデータが表示されず途中で切れる場合があります。

■スクリプト中の入力データのjsonで未対応のパラメータを指定された場合は無視します。

■スクリプト中の入力データのjsonで環境変数を使用する場合、json自体の中括弧と連続した場合に不具合が起きる可能性があります。

たとえば「{"KEY":{{{環境変数名}}}}」のように値を環境変数名で使用する場合、jsonの中括弧が環境変数展開用のものと誤認されるため正しく展開されない可能性があります。

このような場合「{"KEY":{{{環境変数名}}} }」のように中括弧の間にスペースや改行を入れることで回避できます。

■InvocationTypeでRequestResponseを指定した場合、Invoke　APIはLambda関数の実行が完了するまで終了せず、この間セッションを保持したままとなります。この通信においてKeepAliveは実行されません。

セッションを切断されるとLambda関数の実行中であっても部品が異常終了しますので、ネットワーク中にファイアウォールなどが存在する場合設定に注意頂く必要があります。

4.10. EKS

Amazon Elastic Kubernetes Service(以下、EKSと記します)の部品について説明します。

4.10.1. EKS部品共通のカスタムパラメータ

EKS部品で共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.10.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	eks-endpoint	文字列	EKSについてAPIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、eks.ap-northeast-1.amazonaws.comが指定されています。	eks.ap-northeast-1.amazonaws.com
○	sts-endpoint	文字列	AWS　STSについてAPIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、sts.ap-northeast-1.amazonaws.comが指定されています。	sts.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.10.2. ジョブ実行

4.10.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/EKS

■定義名
RunJob

■表示アイコン

4.10.2.2. 部品説明

EKSのクラスタ上にKubernetesのJobリソースを登録して完了を待ち合わせます。

リソースを登録するクラスタはEKSのクラスタ名で指定します。また登録するJobリソースの内容は部品のスクリプト中に記載します。

4.10.2.3. スクリプト

本部品のスクリプトでは部品実行時に使用する入力データと部品の実行結果の出力テンプレートを指定できます。

スクリプト中に入力データと出力テンプレートの両方を記載する場合の記述方法については「4.1.1　スクリプトの記述」を参照してください。

■入力データ

以下のようにEKS上に登録するJobリソースをyaml形式で記載します。Jobリソースの詳細はKubernetes側のドキュメントを参照してください。

```
apiVersion: batch/v1
kind: Job
metadata:
  generateName: job-name-
spec:
  template:
    spec:
      containers:
      - name: container-name
        image: cintainer-image
      restartPolicy: Never
```

また環境変数の値を利用したい場合、{{{環境変数名}}}の形式で記述することで環境変数が展開されます。なお、対象の環境変数が存在しない場合は空文字として扱われます。

■出力テンプレート

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.10.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AssumeRoleArn	文字列	EKS上への操作を行うユーザとしてAssumeRoleを使用したい場合、使用するRoleのArnを指定します。	arn:aws:iam::xxxxx:role/role-name
	AssumeRoleSessionName	文字列	AssumeRoleを使用する場合のセッション名を指定します。AssumeRoleが指定されていない場合、本パラメータは無視されます。	EKSGetTokenAuth
	getPodsDetails	リスト	登録したJobリソースの終了後、Jobリソースから生成されたPodの詳細情報を取得する場合はtrueを指定します。	true false
	getPodsLogs	リスト	登録したJobリソースの終了後、Jobリソースから生成されたPodのコンテナのログを取得する場合はtrueを指定します。	true false
	deleteFinishedJob	リスト	登録したJobリソースの終了時に登録したJobリソースの削除を行うかを指定します。 Complete:　Jobリソースが正常終了した場合のみ削除します。 Failed: Jobリソースが異常終了した場合のみ削除します。 All:　状態に関係なく、Jobリソースが終了した時点で削除します。 None: 登録したリソースの削除を行いません。	Complete Failed All None
	create.fieldManager	文字列	Jobリソースの登録時、登録を行った操作者の名前として特定の名前を使用したい場合に指定します。	true false
	log.limitBytes	数値	Jobリソースから生成されたPodのコンテナのログを取得する場合に取得するログのサイズの上限をByte単位で指定します。指定が無い場合は無制限となります。	true false
	log.tailLines	数値	Jobリソースから生成されたPodのコンテナのログを取得する場合に末尾から何行目までのログを取得するかを指定します。指定が無い場合は無制限となります。	100
	log.timestamps	リスト	Jobリソースから生成されたPodのコンテナのログを取得する場合にログの各行の先頭にタイムスタンプを付与するときはtrueを指定します。	true false
○	cluster	文字列	Jobリソースを登録するEKSのクラスタ名を指定します。	default

必須	変数	データ型	説明	設定例
○	namespace	文字列	Jobリソースを登録するEKSのクラスタ上のnamespaceを指定します。	namespace1
	api-server-endpoint	文字列	EKSのクラスタのAPIサーバのエンドポイントを指定します。指定された場合、指定されたAPIサーバのエンドポイントを利用してJobリソースの登録などの操作を行います。	https://xxxxx.gr7.ap-northeast-1.eks.amazonaws.com
	interval	数値	登録したJobリソースの状態の確認を行う間隔を秒単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。	10

4.10.2.5. 部品詳細

本部品は大きく分けて5つのステップの動作があります。

1. EKS上のクラスタのAPIサーバのエンドポイントの取得
2. EKS上のクラスタへリクエストするための認証情報の生成
3. EKS上のクラスタへのJobリソースの登録
4. 登録したJobリソースの終了待ち合わせ
5. Jobリソースの終了後の処理

本部品の動作の詳細を説明します。

1. api-server-endpointが未指定の場合、EKSのクラスタのAPIサーバのエンドポイントを取得します。

   clusterで指定されたEKSのclusterについてDescribeCluster APIを実行し、クラスタのAPIサーバのエンドポイントを取得します。

   DescribeCluster APIによる実行の要求が失敗した場合、部品はコード1で終了します。

   api-server-endpointが指定されている場合、EKSのクラスタのAPIサーバのエンドポイントとしてapi-server-endpointで指定された値を利用します。

2. EKS上のクラスタへJobリソースの登録などのリクエストを送るための認証情報を生成します。

   この時AssumeRoleArnが指定されている場合、AssumeRoleArnで指定されたRoleを利用して認証情報を生成します。

   AssumeRoleArnが指定されていない場合、AccessKeyID、SecretAccessKeyで指定されたユーザを利用して認証情報を生成します。

   以降の処理で認証情報の期限が切れた場合は、内部的に再生成を行います。認証情報の再生成に失敗した場合、部品はコード1で終了します。

3. EKSのクラスタのAPIサーバのエンドポイントにJob V1 BatchのCreateのAPIを実行し、Jobリソースの登録を行います。

   Job V1 BatchのCreateのAPIが失敗した場合、部品はコード1で終了します。

4. 登録したJobリソースについて、終了するまで待ち合わせます。

   intervalパラメータで指定した秒の間隔で、Job V1 BatchのReadのAPIでJobリソースのステータスを取得し、Jobリソースが終了したかをチェックします。

以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

■JobリソースのStatus.Conditions配下にtypeがCompleteのオブジェクトがあるとき

この場合、Jobリソースの終了後の処理を行います。

■JobリソースのStatus.Conditions配下にtypeがFailedのオブジェクトがあるとき

この場合、Jobリソースの終了後の処理を行った後に部品はコード1で終了します。

■JobリソースのStatus.Conditions配下にその他不正なtypeのオブジェクトが存在するとき

この場合、Jobリソースの終了後の処理を行った後に部品はコード1で終了します。

■Job V1 BatchのReadのAPIによるJobリソースのステータスの取得に失敗したとき

この場合、部品はコード1で終了します。

5. 部品のカスタムパラメータの設定に応じてJobリソースの終了後の処理を行います。処理の後、途中でエラーが発生していなければ部品はコード0で終了します。

■getPodsDetailsまたはgetPodsLogsでtrueが指定されている場合

Pod V1 CoreのListのAPIでJobリソースから生成されたPodの詳細情報を取得します。

Pod V1 CoreのListのAPIが失敗した場合、後述のJobリソースの削除処理を行った後に部品はコード1で終了します。

■getPodsLogsでtrueが指定されている場合

Pod V1 CoreのReadLogのAPIでPodのコンテナのログを取得します。このとき、複数のPodやコンテナがある場合は全てのPodの全てのコンテナのログを取得します。

Pod V1 CoreのReadLogのAPIが失敗した場合、後述のJobリソースの削除処理を行った後に部品はコード1で終了します。

■deleteFinishedJobでComplete、Failed、Allのいずれかが指定されている場合

指定された値に従って条件に一致する場合、Job V1 BatchのDeleteのAPIで登録したJobリソースを削除します。

Job V1 BatchのDeleteのAPIが失敗した場合、部品はコード1で終了します。

4.10.2.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下のとおりオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

```
{
  "JobCenterResult": {
    "DescribeCluster": {},
    "AssumeRoleResponse": {},
    "JobV1BatchCreateResponse": {},
    "JobV1BatchReadResponse" : {},
    "JobV1BatchDeleteResponse" : {},
    "PodV1CoreListResponse" : {},
```

```
    "logs": [
      podName: pod-name
      containers: [
        {
          podName: pod-name
          containerName: container-name
          PodV1CoreReadLogResponse:  []
        }
      ]
    ]
  "JobCenterErrorMessageList": []
}
```

■JobCenterResult

JobCenterResultオブジェクトには以下のとおり各APIの結果が保存されます。各APIの結果の詳細についてはAWS側のドキュメントまたはKubernetes側のドキュメントを参照してください。

- DescribeClusterResponse

  DescribeClusterResponseオブジェクトにはAWS EKSのDescribeCluster APIの結果が保存されます。

- AssumeRoleResponse

  AssumeRoleResponseオブジェクトにはAWS STSのAssumeRole APIの結果が保存されます。

- JobV1BatchCreateResponse

  JobV1BatchCreateResponseオブジェクトにはKubernetesのJob V1 BatchのCreateのAPIの結果が保存されます。

- JobV1BatchReadResponse

  JobV1BatchReadResponseオブジェクトには待ち合わせ完了時点の最終的なKubernetesのJob V1 BatchのReadのAPIの結果が保存されます。

- JobV1BatchDeleteResponse

  JobV1BatchDeleteResponseオブジェクトにはKubernetesのJob V1 BatchのDeleteのAPIの結果が保存されます。

- PodV1CoreListResponse

  PodV1CoreListResponseオブジェクトにはKubernetesのPod V1 CoreのListのAPIの結果が保存されます。

- logs

  logs配列にはPod毎のログの情報が保存されます。

  - podname

    podnameには対象のPodの名前が保存されます。

  - containers

    containres配列には対象のPodのコンテナの情報が保存されます。

    - podName

podNameにはコンテナの所属するPodの名前が保存されます。

- containerName

  containerNameにはコンテナの名前が保存されます。

- PodV1CoreReadLogResponse

  PodV1CoreReadLogResponse配列にはKubernetesのPod V1 CoreのReadLogのAPIで取得したログを改行で分割した配列が保存されます。

■JobCenterErrorMessageList

JobCenterErrorMessageList配列にはエラーが発生した場合のメッセージが保存されます。保存されるエラーメッセージについては「4.10.2.7 主要メッセージ」を参照してください。

4.10.2.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is invalid. The value is out of range.	パラメータに範囲外の値を指定した場合に出力されます。
input script must be yaml format and type Object.	スクリプトの入力データに誤りがある場合に出力されます。スクリプトの入力データはyaml形式である必要があります。
DescribeCluster request failure.	DescribeCluster APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
AssumeRole request failure.	AssumeRole APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
JobV1BatchCreate request failure.	Job V1 BatchのCreateのAPIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
JobV1BatchRead request failure.	Job V1 BatchのReadのAPIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
JobV1BatchDelete request failure.	Job V1 BatchのDeleteのAPIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
PodV1CoreList request failure. request failure.	Pod V1 CoreのListのAPIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
PodV1CoreLog request failure. request failure.	Pod V1 CoreのLogのAPIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
Included Failed type in job conditions.	Jobリソースが失敗して終了した場合、JobCenterErrorMessageList配列に格納されます。
Failed to get token again.	認証情報の期限切れによる再認証時に失敗した場合、JobCenterErrorMessageList配列に格納されます。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のJob V1 BatchのReadのAPIについては出力されず、待ち合わせ完了時点での最後のJob V1 BatchのReadのAPIについてのみ出力されます。 また認証情報の期限切れの場合の再認証時のAssumeRole APIについても出力されません。
Failed to XXX. Message：XXX	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

メッセージ	内容
Code：XXX	

4.10.2.8. 使用API

本部品が使用するAWS、KubernetesのAPIの一覧は以下のとおりです。

部品の実行前にあらかじめIAMのアクセス許可、KubernetesのRBACの許可を設定してください。

サービス	アクション
AWS EKS	DescribeCluster
AWS STS	AssumeRole
Kubernetes	Job V1 Batch Create
Kubernetes	Job V1 Batch Read
Kubernetes	Job V1 Batch Delete
Kubernetes	Pod V1 Core List
Kubernetes	Pod V1 Core ReadLog

4.10.2.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには全てのデータが表示されず途中で切れる場合があります。

4.11. ECS

Amazon Elastic Container Service(以下、ECSと記します)の部品について説明します。

4.11.1. ECS部品共通のカスタムパラメータ

ECS部品で共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.11.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	AccessKeyID	文字列	APIを利用するためのアクセスキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	文字列	APIを利用するためのシークレットキーを指定します。 IAM　Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	文字列	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、ecs.ap-northeast-1.amazonaws.comが指定されています。	ecs.ap-northeast-1.amazonaws.com
	proxy	文字列	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.example.com:8080

4.11.2. タスク実行

4.11.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ECS

■定義名
RunTask

■表示アイコン

4.11.2.2. 部品説明

ECSのクラスタ上でタスクを実行して完了を待ち合わせます。実行するタスク定義は事前にAWS側で作成しておく必要があります。

4.11.2.3. スクリプト

本部品のスクリプトでは部品実行時に使用する入力データと部品の実行結果の出力テンプレートを指定できます。

スクリプト中に入力データと出力テンプレートの両方を記載する場合の記述方法については「4.1.1　スクリプトの記述」を参照してください。

■入力データ

タスク実行時のパラメータについてカスタムパラメータで指定できるものでは不十分な場合、スクリプトでも以下のようにjson形式で指定できます。

指定したjson形式のデータがそのままタスク実行時に使用するECSのRunTask APIのリクエストのパラメータとして使用されますので、指定形式の詳細についてはAWSのドキュメントを参照してください。

なお、スクリプトで指定した設定がカスタムパラメータで指定した設定と重複している場合、以下のようにマージされます。

- enableECSManagedTagsなどの配列以外の値の場合、スクリプト側の設定が優先されます。
- tagsやcontainerOverridesなどの配列の値の場合、スクリプト側の設定の後にカスタムパラメータで指定した設定が追加されます。

```
{
  "enableECSManagedTags": true,
  "overrides": {
    "containerOverrides" : [
      {
        "name": "container-name",
        "command": ["./example.sh" ]
      }
    ]
  },
  "tags": [
    "key": "tag-key",
    "value": "tag-value"
  ]
```

```
}
```

また環境変数の値を利用したい場合、{{{環境変数名}}}の形式で記述することで環境変数が展開されます。
なお、対象の環境変数が存在しない場合は空文字として扱われます。

■出力テンプレート

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.11.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
	capacityProviderStrategy	文字列	タスクに使用するキャパシティープロバイダー戦略を指定したい場合に以下の形式で指定します。 `capacityProvider=string,weight=integer,base=integer` 複数の値を入力したい場合はスペース区切りで指定できます。	capacityProvider=provider,weight=1,base=0
○	cluster	文字列	タスクを実行するECSのクラスターの指定します。	cluster-name
	count	数値	同一のタスク定義から同時に複数のタスクを実行したい場合、実行するタスクの数を指定します。最大10個までのタスクを同時に実行できます。	2
	enableECSManagedTags	リスト	実行するタスクに対してAmazon ECSマネージドタグを有効にしたい場合はtrueを指定します。	true false
	enableExecuteCommand	リスト	実行するタスクでコンテナのコマンド実行機能を有効にしたい場合はtrueを指定します。	true false
	group	文字列	実行するタスクに関連付けるグループの名前を指定したい場合にグループ名を指定します。	group1
	launchType	リスト	タスクを実行するインフラストラクチャのタイプを指定します。	EC2 FARGATE EXTERNAL
	networkConfiguration.subnets	文字列	タスクに関連付けるサブネットのIDを指定します。タスク定義のネットワークモードがawsvpcの場合に指定が必要となります。 複数の値を入力したい場合はスペース区切りで指定できます。	subnet-xxxxxxxxxxxxxxxxx

必須	変数	データ型	説明	設定例
	networkConfiguration. securityGroups	文字列	タスクに関連付けるセキュリティグループのIDを指定します。タスク定義のネットワークモードがawsvpcの場合に指定が必要となります。 複数の値を入力したい場合はスペース区切りで指定できます。	sg- xxxxxxxxxxxxxxxx
	networkConfiguration. assignPublicIp	文字列	タスクのElastic Network Interfaceにパブリックアドレスを持たせるかを指定します。タスク定義のネットワークモードがawsvpcの場合に指定できます。	ENABLED DISABLED
	containerOverrides. name	文字列	タスクの実行時にタスク定義のコンテナ設定をオーバーライドしたい場合にオーバーライドするコンテナの名前を指定します。 実際にオーバライドする設定については、containerOverrides.で始まる名前のパラメータで指定します。 カスタムパラメータでは単一のコンテナのオーバーライドのみ指定できます。複数のコンテナのオーバーライドをしたい場合、スクリプトの入力データを使用して設定する必要があります。	container1
	containerOverrides. command	文字列	コンテナのデフォルトコマンドをオーバーライドする場合に指定します。 コマンドと引数は,(カンマ)区切りで入力します。	ping,192.168.0.1
	containerOverrides. environment	文字列	コンテナで使用する環境変数をオーバーライドする場合に以下の形式で指定します。 `name=string,value=string` 複数の値を入力したい場合はスペース区切りで指定できます。	name=env1, value=val1
	containerOverrides. environmentFiles	文字列	コンテナで使用する環境変数をオーバーライドする場合に環境変数を記載したS3上のファイルのリストを以下の形式で指定します。 `value=string,type=s3` 複数の値を入力したい場合はスペース区切りで指定できます。 S3上に配置するファイルの詳細はAWS側のドキュメントを参照してください。	value=arn:aws:s3::: bucket_name/ key_name,type=s3
	containerOverrides. cpu	数値	コンテナの予約CPUユニット数をオーバーライドしたい場合に指定します。	10
	containerOverrides. memory	数値	コンテナのメモリのハード制限をオーバーライドしたい場合にMiB単位で指定します。	2048

必須	変数	データ型	説明	設定例
	containerOverrides. memoryReservation	数値	コンテナのメモリのソフト制限をオーバーライドしたい場合にMiB単位で指定します。	2048
	containerOverrides. resourceRequirements	文字列	コンテナに割り当てるリソースのタイプと量をオーバーライドしたい場合に以下の形式で指定します。 `value=string,type=string` 複数の値を入力したい場合はスペース区切りで指定できます。 指定できるタイプや量の詳細はAWS側のドキュメントを参照してください。	value=2,type=GPU
	overrides.cpu	文字列	タスク定義のCPUの設定をオーバーライドしたい場合に指定します。	1 vCPU
	overrides. inferenceAccelerator Overrides	文字列	タスク定義のinferenceAcceleratorをオーバーライドする場合に以下の形式で指定します。 `deviceName=string,deviceType=string` 複数の値を入力したい場合はスペース区切りで指定できます。 指定できるデバイスタイプの詳細はAWS側のドキュメントを参照してください。	deviceName=dev1, deviceType=eia1. medium
	overrides. executionRoleArn	文字列	タスク定義のタスク実行時に使用するIAMロールの設定をオーバーライドする場合にIAMロールのARNを指定します。	arn:aws:iam::xxxxx: role/taskExecRole
	overrides.memory	文字列	タスク定義のメモリの設定をオーバーライドしたい場合に指定します。	2GB
	overrides.taskRoleArn	文字列	タスク定義のタスク実行時にコンテナに適用するIAMロールの設定をオーバーライドする場合にIAMロールのARNを指定します。	arn:aws:iam::xxxx: role/ecsTaskRole
	overrides. ephemeralStorage. sizeInGiB	数値	タスク定義のタスクに設定するストレージの合計量をオーバーライドする場合にGiB単位で指定します。	40

	placementConstraints	文字列	タスク実行時のタスク配置の制約事項を以下の形式で指定します。タスク配置の制約事項の詳細はAWS側のドキュメントを参照してください。
			
			`type=memberOf,expression=string`
			
			`type=distinctInstance`
			
			複数の値を入力したい場合はスペース区切りで指定できます。
				type=memberOf,
				expression=
				attribute:
				ecs.instance-
				type=~t2.*

	placementStrategy	文字列	タスク実行時のタスク配置戦略を以下の形式で指定します。タスク配置戦略の詳細は	type=spread, field=attribute:

必須	変数	データ型	説明	設定例

			AWS側のドキュメントを参照してください。
			
			`type=binpack,field=string`
			
			`type=random`
			
			`type=spread,field=string`
			
			複数の値を入力したい場合はスペース区切りで指定できます。
				ecs.availability-zone

	platformVersion	文字列	タスク実行時に使用するプラットフォームのバージョンを指定します。Fargateでホストされているタスクに対してのみ指定できます。指定が無い場合、デフォルト最新のバージョンが使用されます。	1.4.0
	propagateTags	リスト	タスク定義のタグを実行するタスクで使用する場合にTASK_DEFINITIONを指定します。	TASK_DEFINITION
	startedBy	文字列	タスクの開始時にタスクに適用されるタグを指定できます。	started-tag
	tags	文字列	タスクに適用するタグを以下の形式で指定できます。 `key=string,value=string` 複数の値を入力したい場合はスペース区切りで指定できます。	key=tag1, value=val1
○	taskDefinition	文字列	実行するタスク定義の名前を指定します。リビジョンを指定指定したい場合は定義名:リビジョンの形式で指定できます。	task-definition task-definition:3
	getCloudWatchLogs	リスト	実行したタスクの実行ログをCloudWatch Logsから取得したいときにtrueを指定します。 本パラメータはタスク定義のlogDriverとしてawslogsを使用している場合にのみ有効です。logDriverの設定がawslogs以外の場合はtrueを指定しても実行ログの取得は行われません。	true false
	interval	数値	実行したタスクの状態の確認を行う間隔を秒単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。	10
	api.enableRetry	リスト	副作用の無いAWS APIの自動リトライを有効にするかを指定します。デフォルト値としてtrue(有効)が指定されています。falseの場合は無効として扱います。	true false
	api.retryInterval	数値	副作用の無いAWS APIの自動リトライの間隔を秒数で指定します。デフォルト値として5秒が指定されています。1から60までの値を指定可能です。	10

必須	変数	データ型	説明	設定例
	api.retryMax	数値	副作用の無いAWS APIの自動リトライの最大回数を指定します。デフォルト値として3回が指定されています。1から10までの値を指定可能です。	5
	api.withSideEffects.enableRetry	リスト	副作用のあるAWS APIの自動リトライを有効にするかを指定します。デフォルト値としてfalse(無効)が指定されています。trueの場合は有効として扱います。	true false
	api.withSideEffects.retryInterval	数値	副作用のあるAWS APIの自動リトライの間隔を秒数で指定します。デフォルト値として5秒が指定されています。1から60までの値を指定可能です。	10
	api.withSideEffects.retryMax	数値	副作用のあるAWS APIの自動リトライの最大回数を指定します。デフォルト値として3回が指定されています。1から10までの値を指定可能です。	5

4.11.2.5. 部品詳細

本部品は大きく分けて3つのステップの動作があります。

1. タスクの実行要求
2. 実行したタスクが終了するで待ち合わせ
3. タスクの終了後の処理

本部品の動作の詳細を説明します。

1. TaskDefinitionで指定されたタスク定義について、RunTask APIで実行の要求を行います。

   RunTask APIによる実行の要求が失敗した場合、部品はコード3で終了します。

2. 実行したタスクが終了するまで待ち合わせます。

   intervalパラメータで指定した秒の間隔で、DescribeTasks APIでタスクのステータスを取得し、タスクが終了したかをチェックします。

   以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

   ■全てのTaskのJobRunStateがSTOPPEDでかつ各コンテナのexitCodeが0のとき

   この場合、タスクの終了後の処理を行います。

   ■全てのTaskのJobRunStateがSTOPPEDでかつ各コンテナのexitCodeに0以外が含まれるとき

   この場合、タスクの終了後の処理を行った後に部品はコード1で終了します。

   ■TaskのJobRunStateがその他不正な値であったとき

   この場合、部品はコード4で終了します。

   ■DescribeTasks APIによるタスクのステータスの取得に失敗したとき

   この場合、部品はコード4で終了します。

3. 部品のカスタムパラメータの設定に応じてタスクの終了後の処理を行います。処理の後、途中でエラーが発生していなければ部品はコード0で終了します。

■getCloudWatchLogsでtrueが指定されている場合

a. DescribeTaskDefinition APIでタスク定義の情報を取得します。

DescribeTaskDefinition APIが失敗した場合、部品はコード5で終了します。

b. 全てのコンテナのlogDriverを確認してawslogsが存在しない場合、部品はコード0で終了します。

logDriverにawslogsが設定されているコンテナが存在する場合、対象のコンテナについてGetLogEvents APIで実行ログを取得します。

GetLogEvents APIが失敗した場合、部品はコード5で終了します。

処理中に呼び出す各種AWS　APIがエラーとなった場合や通信が切断された場合、カスタムパラメータの設定に応じて自動的にリトライを行います。

自動リトライは副作用のあるAPIと無いAPIそれぞれについて個別に設定します。副作用のあるAPIとはRunTask　APIを、副作用の無いAPIとはそれ以外のDescribeTasks　API、DescribeTaskDifinition API、GetLogEvents APIを指します。

副作用のあるAPIについて自動リトライを有効にした場合、リトライによってタスクが重複実行される可能性があるため、実行するタスクについて重複実行が起きうる前提で設計を行ってください。

4.11.2.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下のとおりオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

```
{
  "JobCenterResult": {
    "RunTaskResponse": {},
    "DescribeTasksResponse": {},
    "DescribeTaskDefinitionResponse": {},
    "Logs": [
      {
        "taskArn": "arn:aws:ecs:ap-northeast-1:xxxxxxxxxxxx:task/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx",
        "containers": [
          {
            "taskArn": "arn:aws:ecs:ap-northeast-1:xxxxxxxxxxxx:task/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx",
            "taskId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
            "name": "container-name"
            "GetLogEventsResponse": {}
          }
        ]
      }
    ]
  },
  "JobCenterErrorMessageList": []
}
```

■JobCenterResult

JobCenterResultオブジェクトには以下のとおり各APIの結果が保存されます。各APIの結果の詳細についてはAWS側のドキュメントを参照してください。

- RunTaskResponse

  RunTaskResponseオブジェクトにはECSのRunTask APIの結果が保存されます。

- DescribeTasksResponse

  DescribeTasksResponseオブジェクトにはECSのDescribeTasks APIの結果が保存されます。

- DescribeTaskDefinitionResponse

  DescribeTaskDefinitionResponseオブジェクトにはECSのDescribeTaskDefinition APIの結果が保存されます。

- Logs

  Logs配列にはタスク毎のログの情報が保存されます。

  - taskArn

    taskArnには対象のタスクのArnが保存されます。

  - containers

    containres配列には対象のタスクのコンテナの情報が保存されます。

    - taskArn

      taskArnには対象のタスクのArnが保存されます。

    - taskId

      taskIdには対象のタスクのIDが保存されます。

    - name

      nameにはコンテナの名前が保存されます。

    - GetLogEventsResponse

      GetLogEventsResponseオブジェクトにはCloudWatch LogsのGetLogEvents APIの結果が保存されます。

■JobCenterErrorMessageList

JobCenterErrorMessageList配列にはエラーが発生した場合のメッセージが保存されます。保存されるエラーメッセージについては「4.11.2.8 主要メッセージ」を参照してください。

4.11.2.7. 終了コード

終了コード	説明
0	タスク実行と終了待ち合わせが正常に行われ、すべてのコンテナの終了コードが0だった場合
1	タスク実行と終了待ち合わせが正常に行われ、いずれかのコンテナの終了コードが非0だった場合
2	カスタムパラメータまたはスクリプトの内容が不正でエラーとなった場合
3	タスク実行に失敗した場合

終了コード	説明
4	タスク実行は成功したが、終了待ち合わせに失敗した場合
5	タスク実行と終了待ち合わせは正常に行われたが、いずれかのログ取得に失敗した場合

4.11.2.8. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is invalid. The value is out of range.	パラメータに範囲外の値を指定した場合に出力されます。
parameter [XXX] is invalid foramt.	指定されたパラメータのフォーマットが不正な場合に出力されます。
input script must be yaml format and type Object.	スクリプトの入力データに誤りがある場合に出力されます。スクリプトの入力データはjson形式のオブジェクトである必要があります。
RunTask request failure.	RunTask APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
DescribeTasks request failure.	DescribeTasks APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
DescribeTaskDefinition request failure.	DescribeTaskDefinition APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
GetLogEvents request failure.	GetLogEvents APIのリクエストに失敗した場合、JobCenterErrorMessageList配列に格納されます。
Container returned non-zero exit code.	exitCodeが0以外のコンテナが存在する場合、JobCenterErrorMessageList配列に格納されます。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のDescribeTasks APIについては出力されず、待ち合わせ完了時点での最後のDescribeTasks APIについてのみ出力されます。
Failed to XXX. Message：XXX Code：XXX	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.11.2.9. 使用API

本部品が使用するAWSのAPIの一覧は以下のとおりです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
ECS	RunTask
ECS	DescribeTasks
ECS	DescribeTaskDefinition
CloudWatch Logs	GetLogEvents

4.11.2.10. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには全てのデータが表示されず途中で切れる場合があります。

■スクリプト中の入力データのjsonで環境変数を使用する場合、json自体の中括弧と連続した場合に不具合が起きる可能性があります。

たとえば「{"KEY":{{{環境変数名}}}}」のように値を環境変数名で使用する場合、jsonの中括弧が環境変数展開用のものと誤認されるため正しく展開されない可能性があります。

このような場合「{"KEY":{{{環境変数名}}} }」のように中括弧の間にスペースや改行を入れることで回避できます。

■副作用のあるAWS　APIの自動リトライを有効にした場合(api.withSideEffects.enableRetryをtrueに設定した場合)、リトライによってタスクが重複実行される可能性があります。

自動リトライを有効にする場合は、実行するタスクについて重複実行が起きうる前提での設計を行ってください。

5. Microsoft Azure(V2) 連携部品

本章ではAzure Resource Manager(以下、Azure(V2)と記します)連携部品について説明します。

5.1. 部品共通のカスタムパラメータ

Azure(V2)連携部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

5.1.1. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	client_id	文字列	AzureのAPIの認証に必要です。あらかじめ取得したクライアントIDを指定してください。	
○	username	文字列	AzureのAPIの認証に必要です。あらかじめ作成したユーザの名前を指定してください。	jobcenteruser@<ドメイン名>
○	password	文字列	AzureのAPIの認証に必要です。あらかじめ作成したユーザのパスワードを指定してください。	
○	endpoint	文字列	AzureのAPIのエンドポイントを指定します。デフォルト値としてhttps://management.azure.com/が指定されています。	https://management.azure.com/
○	token_endpoint	文字列	AzureのAPIを使用するためのトークンを取得するためのエンドポイントを指定します。あらかじめ取得したOAUTH 2.0トークン エンドポイントを指定してください。	https://login.microsoftonline.com/xxxxxxxx/oauth2/token
	proxy	文字列	プロキシ経由でAzureのAPIを使用する場合に指定してください。	http://proxy.example.com:8080

5.2. 仮想マシン起動部品

5.2.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
StartVM

■表示アイコン

5.2.2. 部品説明

指定した仮想マシンの起動要求を行います。

対象となる仮想マシンは、名前を直接指定する他、タグで検索することができます。

5.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.2.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	subscription_id	文字列	Azureのサブスクリプション IDを指定します。	
△	resource_group	文字列	起動したい仮想マシンが属するリソースグループの名前を指定します。 vm_name/パラメータを指定する場合は指定が必須です。	ResourceGroup1
△	vm_name	文字列	起動したい仮想マシンの名前を指定します。 複数指定する場合は、名前をセミコロン(;)で区切ってください。 このパラメータまたはtag/パラメータのどちらかが必須です。	vmtest1 vmtest1;vmtest2
△	tag	文字列	起動したい仮想マシンをタグで検索します。 タグ名=値 の形式で指定してください。 このパラメータまたはvm_name/パラメータのどちらかが必須です。	AutoStart=ON
	wait	リスト	仮想マシンの起動要求を行った後、実際に仮想マシンが起動完了するまで部品の終了を待ち合わせる場合はtrueを指定してください。 待ち合わせない場合は空欄またはfalseを指定してください。	true false

必須	変数	データ型	説明	設定例
			デフォルト値としてtrueが指定されています。	
	interval	数値	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの起動完了チェックを行う間隔を秒数で指定します。未指定時はデフォルト値として5が使用されます。	10
	timeout	数値	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの起動完了チェックのタイムアウト時間を秒数で指定します。未指定時はデフォルト値として300が使用されます。	600

5.2.5. 実行結果

■waitパラメータが未指定、またはfalseを指定した場合

起動要求が正常終了した場合は0、エラー時には1が終了値となります。

なお、複数の仮想マシンに対して起動要求を行った場合、すべての起動要求が成功した場合は0、そうでない場合は1が終了値となります。

■waitパラメータにtrueを指定した場合

起動要求および起動チェックが正常終了した場合は0、タイムアウトした場合やエラー時には1が終了値となります。

なお、複数の仮想マシンが対象の場合、すべての起動要求および起動チェックが成功した場合は0、そうでない場合は1が終了値となります。

5.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

5.2.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■vm_nameパラメータ、tagパラメータの両方を指定した場合はvm_nameが優先されます。

5.3. 仮想マシン停止およびリソース解放部品

5.3.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
StopVM

■表示アイコン

5.3.2. 部品説明

指定した仮想マシンの停止要求を行います。

対象となる仮想マシンは、名前を直接指定する他、タグで検索することができます。

停止に成功した場合、仮想マシンの状態は"Stopped (deallocated)"の状態となります。

5.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	subscription_id	文字列	Azureのサブスクリプション IDを指定します。	
△	resource_group	文字列	停止したい仮想マシンが属するリソースグループの名前を指定します。 vm_nameパラメータを指定する場合は指定が必須です。	ResourceGroup1
△	vm_name	文字列	停止したい仮想マシンの名前を指定します。 複数指定する場合は、名前をセミコロン(;)で区切ってください。 このパラメータまたはtagパラメータのどちらかが必須です。	vmtest1 vmtest1;vmtest2
△	tag	文字列	停止したい仮想マシンをタグで検索します。 タグ名=値 の形式で指定してください。 このパラメータまたはvm_nameパラメータのどちらかが必須です。	AutoStart=ON
	wait	リスト	仮想マシンの停止要求を行った後、実際に仮想マシンが停止完了するまで部品の終了を待	true

必須	変数	データ型	説明	設定例
			ち合わせる場合はtrueを指定してください。 待ち合わせない場合は空欄またはfalseを指定してください。 デフォルト値としてtrueが指定されています。	false
	interval	数値	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックを行う間隔を秒数で指定します。未指定時はデフォルト値として5が使用されます。	10
	timeout	数値	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックのタイムアウト時間を秒数で指定します。未指定時はデフォルト値として300が使用されます。	600

5.3.5. 実行結果

■waitパラメータが未指定、またはfalseを指定した場合

停止要求が正常終了した場合は0、エラー時には1が終了値となります。

なお、複数の仮想マシンに対して停止要求を行った場合、すべての停止要求が成功した場合は0、そうでない場合は1が終了値となります。

■waitパラメータにtrueを指定した場合

停止要求および停止チェックが正常終了した場合は0、タイムアウトした場合やエラー時には1が終了値となります。

なお、複数の仮想マシンが対象の場合、すべての停止要求および停止チェックが成功した場合は0、そうでない場合は1が終了値となります。

5.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

5.3.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■vm_nameパラメータ、tagパラメータの両方を指定した場合はvm_nameが優先されます。

5.4. 仮想マシン停止部品

5.4.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
PowerOffVM

■表示アイコン

5.4.2. 部品説明

指定した仮想マシンの停止要求を行います。

対象となる仮想マシンは、名前を直接指定する他、タグで検索することができます。

停止に成功した場合、仮想マシンの状態は"Stopped"の状態となります。

"Stopped"の状態ではネットワークリソースなどが解放されず、課金状態が維持されます。

5.4.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.4.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	subscription_id	文字列	Azureのサブスクリプション IDを指定します。	
△	resource_group	文字列	停止したい仮想マシンが属するリソースグループの名前を指定します。 vm_nameパラメータを指定する場合は指定が必須です。	ResourceGroup1
△	vm_name	文字列	停止したい仮想マシンの名前を指定します。 複数指定する場合は、名前をセミコロン(;)で区切ってください。 このパラメータまたはtagパラメータのどちらかが必須です。	vmtest1 vmtest1;vmtest2
△	tag	文字列	停止したい仮想マシンをタグで検索します。 タグ名=値 の形式で指定してください。 このパラメータまたはvm_nameパラメータのどちらかが必須です。	AutoStart=ON

必須	変数	データ型	説明	設定例
	wait	リスト	仮想マシンの停止要求を行った後、実際に仮想マシンが停止完了するまで部品の終了を待ち合わせる場合はtrueを指定してください。待ち合わせない場合は空欄またはfalseを指定してください。 デフォルト値としてtrueが指定されています。	true false
	interval	数値	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックを行う間隔を秒数で指定します。未指定時はデフォルト値として5が使用されます。	10
	timeout	数値	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックのタイムアウト時間を秒数で指定します。未指定時はデフォルト値として300が使用されます。	600

5.4.5. 実行結果

■waitパラメータが未指定、またはfalseを指定した場合

停止要求が正常終了した場合は0、エラー時には1が終了値となります。

なお、複数の仮想マシンに対して停止要求を行った場合、すべての停止要求が成功した場合は0、そうでない場合は1が終了値となります。

■waitパラメータにtrueを指定した場合

停止要求および停止チェックが正常終了した場合は0、タイムアウトした場合やエラー時には1が終了値となります。

なお、複数の仮想マシンが対象の場合、すべての停止要求および停止チェックが成功した場合は0、そうでない場合は1が終了値となります。

5.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

5.4.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■vm_nameパラメータ、tagパラメータの両方を指定した場合はvm_nameが優先されます。

5.5. 仮想マシン情報(モデルビュー)取得

5.5.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
ModelView

■表示アイコン

5.5.2. 部品説明

指定した仮想マシンの情報取得を行います。

対象となる仮想マシンは名前で指定します。

情報取得に成功した場合、ジョブの出力結果に情報を出力します。

取得できる情報としては、タグ、仮想マシンのサイズなどがあります。

その他、取得できる情報については下記サイトを参照してください。

`https://msdn.microsoft.com/ja-jp/library/azure/mt163682.aspx`

5.5.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.5.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	subscription_id	文字列	Azureのサブスクリプション IDを指定します。	
○	resource_group	文字列	情報取得したい仮想マシンが属するリソースグループの名前を指定します。	ResourceGroup1
○	vm_name	文字列	情報取得したい仮想マシンの名前を指定します。	vmtest1 vmtest1;vmtest2

5.5.5. 実行結果

情報取得が正常終了した場合は0、エラー時には1が終了値となります。

以下に、取得した情報の出力例を示します。

```
Successfully model_view
```

```
---------- ModelViewResult ----------
type: Microsoft.Compute/virtualMachines
name: WS2012R2-test
id: /subscriptions/XXXXXXXXXX/resourceGroups/rg-test1/providers/Microsoft.Compute/virtualMachines/
WS2012R2-test
location: japaneast
----- tags -----
AutoShutdown=Yes
AutoStart=Yes

----- properties -----
osProfile:
  computerName: WS2012R2-test
  adminUsername: JobCenter
networkProfile:
  id: /subscriptions/XXXXXXXXXX/resourceGroups/rg-test1/providers/Microsoft.Compute/
virtualMachines/WS2012R2-test
hardwareProfile:
  vmSize: Standard_DS1
storageProfile:
  dataDisks:
  imageReference:
    version: latest
    publisher: MicrosoftWindowsServer
    sku: 2012-R2-Datacenter
    offer: WindowsServer
  osDisk:
    name: WS2012R2-test
    vhd:
      uri: https://XXXXXXXXXX.blob.core.windows.net/vhds/WS2012R2-test.vhd
    caching: ReadWrite
    osType: Windows
    createOption: FromImage

provisioningState: Succeeded
```

5.5.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。

5.5.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

5.6. 仮想マシン情報(インスタンスビュー)取得

5.6.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
InstanceView

■表示アイコン

5.6.2. 部品説明

指定した仮想マシンの情報取得を行います。

対象となる仮想マシンは名前で指定します。

情報取得に成功した場合、ジョブの出力結果に情報を出力します。

取得できる情報としては、仮想マシンの電源状態、ゲスト エージェントの状態などがあります。

その他、取得できる情報については下記サイトを参照してください。

`https://msdn.microsoft.com/ja-jp/library/azure/mt163682.aspx`

5.6.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.6.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	subscription_id	文字列	Azureのサブスクリプション IDを指定します。	
○	resource_group	文字列	情報取得したい仮想マシンが属するリソースグループの名前を指定します。	ResourceGroup1
○	vm_name	文字列	情報取得したい仮想マシンの名前を指定します。	vmtest1 vmtest1;vmtest2

5.6.5. 実行結果

情報取得が正常終了した場合は0、エラー時には1が終了値となります。

以下に、取得した情報の出力例を示します。

```
Successfully instance_view
```

```
---------- vmAgent ----------
message: VM Agent is unresponsive.
time: 2016-01-19T08:32:50+00:00
code: ProvisioningState/Unavailable
level: Warning
displayStatus: Not Ready
vmAgentVersion: Unknown

---------- disks ----------
name: WS2012R2-test

level: Info
time: 2016-01-18T07:19:59.2814519+00:00
displayStatus: Provisioning succeeded
code: ProvisioningState/succeeded
-----

---------- statuses ----------
level: Info
time: 2016-01-18T07:19:59.343958+00:00
displayStatus: Provisioning succeeded
code: ProvisioningState/succeeded
-----

level: Info
displayStatus: VM deallocated
code: PowerState/deallocated
-----
```

5.6.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。

5.6.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

6. UiPath 連携部品

UiPath 連携部品ついて説明します。

6.1. ジョブ開始

6.1.1. 部品情報

■格納パス
/Root/System/UiPath/Jobs

■定義名
StartJobs

■表示アイコン

6.1.2. 部品説明

UiPath Orchestratorで定義済みのプロセスからジョブを起動し、その実行を完了まで待ち合わせます。

なお、本部品でジョブを起動できるのはUnattendedロボットのみです。

6.1.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

6.1.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	endpoint	文字列	UiPath OrchestratorのAPIエンドポイントのURLを指定します。	http://myuipath.local
	proxy	文字列	プロキシサーバ経由でAPIリクエストを送信する場合に指定します。 プロトコル名://サーバ名:ポート番号 の形式で指定します。	http://proxy:8080
	tenancyName	文字列	テナント名を指定します。 省略時は default となります。	tenant1
○	usernameOrEmailAddress	文字列	ユーザ名、またはメールアドレスを指定します。	user1
○	password	文字列	パスワードを指定します。 値を直接入力せずに秘匿化したい場合は「1.2.2.6.2 暗号化パラメータの利用」を参照してください。	password123
	expirationTime	数値	APIアクセストークンの有効期限を分単位で指定します。 UiPath Orchestratorの認証トークンの有効期限の設定値を指定してください。	60

必須	変数	データ型	説明	設定例
			省略時はUiPath　Orchestratorのデフォルト値である30分となります。	
	interval	数値	ジョブの状態確認を行う間隔を秒単位で指定します。　省略時は30秒となります。	60
	timeout	数値	ジョブの実行完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。 省略時は1440分となります。	120
○	FolderPath	文字列	フォルダパスを指定します。	Shared/folder1
○	ProcessName	文字列	プロセスの名前を指定します。	process1
○	JobsCount	数値	実行するジョブ数を指定します。　省略時は1となります。	3
	RobotAccount	文字列	ロボットアカウントを指定します。	3
	MachineName	文字列	標準マシンまたはマシンテンプレートの名前を指定します。	jcserver
	HostName	文字列	接続済みマシンのホスト名を指定します。 MachineName にマシンテンプレート名を指定した時のみ指定できます。	host1
	InputArgumentsFile	文字列	入力パラメータを記載したJSONファイルを絶対パスで指定します。 なお、入力パラメータファイルの文字コードはJobCenterのセットアップ言語としてください。	C:\input\param1.json
	OutputJobIdsFile	文字列	ジョブIDを出力するファイルを絶対パスで指定します。　省略時はファイル出力を行いません。　既にファイルが存在した場合はファイルを上書きします。 なお、出力するジョブIDが複数ある場合はセミコロン(;)区切りで出力します。	C:\output\jobid1.txt

6.1.5. 部品詳細

本部品の動作の流れを説明します。

1. パラメータの事前チェック

   各パラメータのチェックを行います。不正な指定があった場合、部品はここでコード1で終了します。

2. フォルダIDの取得

   フォルダ情報取得APIリクエストを発行してフォルダ情報を取得してFolderPathパラメータと一致するフォルダのIDを取得します。

   APIリクエストに失敗した場合や、APIレスポンスにFolderPathパラメータと一致するフォルダ情報が含まれていなかった場合は部品はここでコード1で終了します。

3. リリースキーの取得

ジョブを起動するためのリリースキーを取得します。

ProcessNameパラメータを使用してリリース情報取得APIリクエストを発行し、そのレスポンスからリリースキーを取得します。

APIリクエストに失敗した場合や、APIレスポンスにリリース情報が含まれていなかった場合は部品はここでコード1で終了します。

4. ロボットIDの取得

   RobotAccountパラメータが指定されていない場合、この処理はスキップします。

   RobotAccountパラメータが指定された場合、ロボット情報取得APIリクエストを発行してRobotAccountパラメータで指定したロボットアカウントに対応するロボットIDを取得します。

   APIリクエストに失敗した場合や、指定したアカウントのロボットが見つからなかった場合は部品はここでコード1で終了します。

5. マシンIDの取得

   MachineNameパラメータが指定されていない場合、この処理はスキップします。

   MachineNameパラメータが指定された場合、マシン情報取得APIリクエストを発行してMachineNameパラメータで指定したマシン名に対応するマシンIDを取得します。

   APIリクエストに失敗した場合や、指定したマシン名のマシンが見つからなかった場合は部品はここでコード1で終了します。

6. マシンセッションIDの取得

   HostNameパラメータが指定されていない場合、この処理はスキップします。

   HostNameパラメータが指定された場合、マシンセッション情報取得APIリクエストを発行してHostNameパラメータで指定したホスト名に対応するマシンセッションIDを取得します。

   APIリクエストに失敗した場合や、指定したホスト名のマシンセッションが見つからなかった場合は部品はここでコード1で終了します。

7. ジョブの起動

   「リリースキーの取得」ステップで取得したリリースキーを使用して、ジョブ起動APIリクエストを発行してジョブを起動します。

   APIリクエストに失敗した場合、部品はここでコード1で終了します。

8. ジョブIDの出力

   OutputJobIdsFileパラメータが指定されていない場合、この処理はスキップします。

   OutputJobIdsFileパラメータで指定したファイルに「ジョブの起動」ステップで起動したジョブIDを出力します。

   ジョブIDの出力が失敗した場合、部品はここでコード1で終了します。

9. ジョブのステータス監視

   intervalパラメータで指定した秒数間隔でジョブの情報を繰り返し取得し、ジョブのステータスを監視します。

   ジョブのステータスが以下のいずれかとなる、timeoutパラメータで指定した時間を超える、またはAPIリクエストに失敗した場合にステータス監視処理を終了します。

■Successful

■Stopped

■Faulted

10. 結果出力

部品の「スクリプト」に記述されたテンプレートを使用して結果を出力します。

部品の終了コードは「ジョブのステータス監視」ステップの終了理由によって以下になります。

終了コード	説明
0	■部品から実行されたジョブがすべてSuccessfulで終了した場合
1	■部品から実行されたジョブが一つでもStopped, Faultedで終了した場合 ■部品から実行されたジョブがステータス監視処理でtimeoutした場合

6.1.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

以下に、出力例を示します。

```
========== summary ==========
JobIds=76
JobCount=1
SuccessJobIds=76
SuccessJobCount=1
FaultedJobIds=
FaultedJobCount=0
StoppedJobIds=
StoppedJobCount=0
TimeoutJobIds=
TimeoutJobCount=0

========== error messages ==========

========== detail ==========
---------- start jobs result ----------
 ***** Id: 76 *****
  State: Pending
  CreationTime: 2019-11-01T08:10:03.421368Z
  Key: c1d629eb-9db3-4ef7-9a61-c10bccb8eedd
  ReleaseName: process1_TESTENV
  BatchExecutionKey: af9fd7e0-70d3-410c-b1c8-afb93caf564e

---------- get jobs result ----------
 ***** Id: 76 *****
  State: Successful
  CreationTime: 2019-11-01T08:10:03.42Z
  StartTime: 2019-11-01T08:10:03.67Z
  EndTime: 2019-11-01T08:12:11.39Z
  Key: c1d629eb-9db3-4ef7-9a61-c10bccb8eedd
  ReleaseName: process1_TESTENV
  BatchExecutionKey: af9fd7e0-70d3-410c-b1c8-afb93caf564e
  RobotName: robot1
```

```
Info: ジョブは完了しました
OutputArguments: {"Division":"abc","total":1000}
```

6.1.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] should be a number.	数値を指定する必要があるパラメータに数値以外が指定された場合に出力されます。
parameter [XXX] is invalid.	パラメータに指定した値が不正である場合に出力されます。
Failed to read file. (InputArgumentsFile=< InputArgumentsFileパラメータの値 >)	入力パラメータファイルの読み込みに失敗した場合に出力されます。
File format is invalid. (InputArgumentsFile=< InputArgumentsFileパラメータの値 >)	入力パラメータファイルのフォーマットが不正である場合に出力されます。
script is invalid.	部品の「スクリプト」に記述されたテンプレートのフォーマットが不正である場合に出力されます。
Failed to get folder information. [Failed to authenticate.] { Status Code : < ステータスコード > message : < メッセージ > | ErrMsg : < メッセージ > }	フォルダ情報の取得に失敗した場合に出力されます。認証エラーの場合、"Failed to authenticate." も出力されます。
Folder < フォルダパス > not found.	フォルダが見つからない場合に出力されます。
Failed to get XXX. [Failed to authenticate.] { Status Code : < ステータスコード > message : < メッセージ > | ErrMsg : < メッセージ > }	情報の取得に失敗した場合に出力されます。XXXはAPIに応じてRelease, Robot, Machine, Sesssionのいずれかになります。認証エラーの場合、"Failed to authenticate." も出力されます。
Failed to start jobs. [Failed to authenticate.] { Status Code : < ステータスコード > message : < メッセージ > | ErrMsg : < メッセージ > }	ジョブ起動APIリクエストに失敗した場合に出力されます。認証エラーの場合、"Failed to authenticate." も出力されます。
Failed to write file. (OutputJobIdsFile=< OutputJobIdsFileパラメータの値 >) ErrMsg : < メッセージ >	ジョブIDのファイルの書き込みに失敗した場合に出力されます。
Failed to get jobs. [Failed to authenticate.] { Status Code : < ステータスコード > message : < メッセージ > | ErrMsg : < メッセージ > }	ジョブ情報取得APIリクエストに失敗した場合に出力されます。認証エラーの場合、"Failed to authenticate." も出力されます。
Job state check time out.	timeoutパラメータで指定した時間を超えたためジョブのステータス監視処理をタイムアウトした場合に出力されます。

6.1.8. 使用API

本部品が使用するUiPath OrchestratorのAPIの一覧は以下の通りです。

部品の実行前にあらかじめアクセス許可を設定してください。

API	使用用途
POST /api/Account/Authenticate	認証 アクセストークンの取得のため
GET /odata/Folders	フォルダ情報の取得 フォルダIDの取得のため

API	使用用途
GET /odata/Releases	リリース情報取得 リリースキーの取得のため
GET /odata/Robots/UiPath.Server.Configuration.OData.GetRobotsFromFolder(folderId=<フォルダID>)	ロボット情報取得 ロボットIDの取得のため
GET /odata/Machines/UiPath.Server.Configuration.OData.GetAssignedMachines(folderId=<フォルダID>)	マシン情報取得 マシンIDの取得のため
GET /odata/Sessions/UiPath.Server.Configuration.OData.GetMachineSessionRuntimesByFolderId(folderId=<フォルダID>)	マシンセッション情報取得 マシンセッションIDの取得のため
POST /odata/Jobs/UiPath.Server.Configuration.OData.StartJobs	ジョブ起動 ジョブの起動のため
GET /odata/Jobs	ジョブ情報取得 ジョブのステータス監視のため

6.1.9. 注意事項・制限事項

6.1.9.1. 注意事項

■本部品は以下の環境で動作を確認しています。

- UiPath Orchestrator 2021.10.4

■本部品は以下の環境での動作に対応していません。

- UiPath Automation Cloud

■本部品はモダンフォルダのみをサポートします。クラシックフォルダには対応しません。

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには表示されない場合があります。

6.1.9.2. 注意事項

■WindowsのUNICODEモードでの動作については未サポートです。

7. Box連携部品

本章ではBox連携部品について説明します。

7.1. Box連携部品概要

Box連携部品はBox社が提供しているBox APIを利用し、Boxへファイルのアップロードやダウンロードなどの処理を行う機能を提供しています。

これらの部品を利用し、日次処理で作成したファイルをBoxにアップロードするなどの処理を自動化することができます。

本部品はWindowsで利用する場合、非UNICODEモードでセットアップを行い、利用してください。UNICODEモードでセットアップを行った場合、正しく動作しない、文字化けするなどの問題が発生します。

7.1.1. ジョブネットワーク作成方法

Box連携部品を利用したジョブネットワーク作成方法について説明します。

Box連携部品は下記に示す「APIトークンを取得する部品」と「Boxの操作を行う部品」の2種類に分けることができます。

■APIトークンを取得する部品

定義名	機能概要	詳細参照先
box_gettoken	APIトークンを取得する	「7.4 BOX_APIトークン取得部品」

■Boxの操作を行う部品

定義名	機能概要	詳細参照先
box_upload	Boxへファイルをアップロードする	「7.5 BOXアップロード部品」
box_download	Boxからファイルをダウンロードする	「7.6 BOXダウンロード部品」
box_getContentID	Box上のフォルダ、ファイルに対するコンテンツIDを取得する	「7.7 コンテンツID取得部品」

APIトークンを取得する部品を利用する方法、またはBoxの操作を行う部品を単体で利用する方法のいずれかでジョブネットワークを作成します。

7.1.1.1. APIトークンを取得する部品を利用する方法

APIトークンを取得する部品とBoxの操作を行う部品を組み合わせてジョブネットワークを作成します。

Boxへファイルのアップロードを行うジョブネットワークを作成する場合、以下の図のようなジョブネットワークを作成します。APIトークンを取得する部品であるBOX_APIトークン取得部品を配置し、その後にBoxの操作を行う部品であるBOXアップロード部品を配置します。

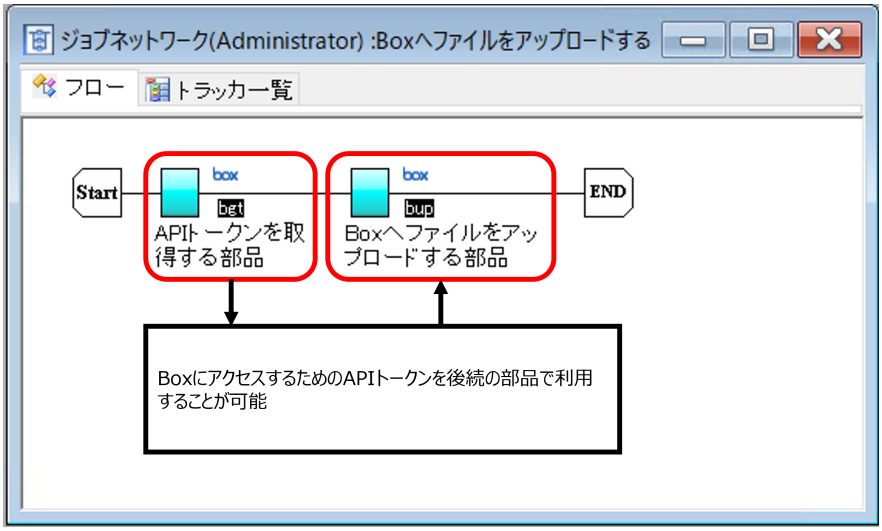


図7.1 Boxへファイルのアップロードを行うジョブネットワーク

この方法でジョブネットワークを作成する場合、BOX_APIトークン取得部品でBoxの操作に必要なAPIトークンを取得し、そのAPIトークンを後続の部品で利用することでBoxの操作を行います。

これにより、APIトークンの設定のみでBoxの操作が可能になるため、複数のBoxの操作を行う部品を配置する場合は、Boxにアクセスするための情報の設定を簡略化できます。

また、Boxの操作を行う部品と同様にHTTP接続部品にAPIトークンを設定し、Box　APIを利用することで、Box連携部品では提供していないBoxの操作を実現することもできます。

Boxにフォルダ作成を行い、そのフォルダに対してファイルをアップロードする場合、以下の図のようなジョブネットワークを作成します。

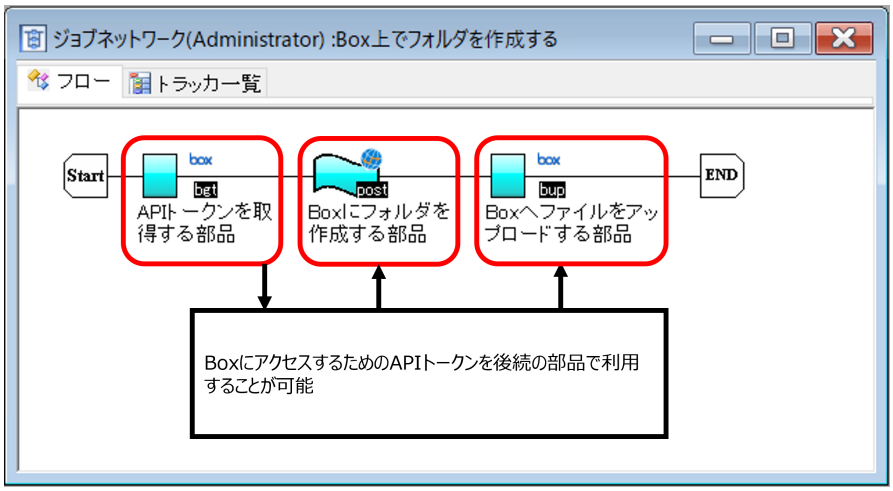


図7.2 Boxにフォルダを作成し、ファイルのアップロードを行うジョブネットワーク

図7.2「Boxにフォルダを作成し、ファイルのアップロードを行うジョブネットワーク」の例では、HTTP-POST部品を利用してフォルダ作成APIを実行し、Box上へフォルダを作成します。その後、BOXアップロード部品を使ってBoxにファイルをアップロードしています。

このようにBOX_APIトークン取得部品で取得したAPIトークンは、HTTP接続部品、BOXアップロード部品のそれぞれに設定することができ、様々なBoxの操作を行うことができます。

以上から、APIトークンを取得する部品を利用する方法は以下のようなケースで有用です。

■推奨ケース

- ジョブネットワーク上に複数のBoxの操作を行う部品を配置する場合
- Box連携部品で提供していない操作をHTTP接続部品を使って実現する場合

また、以下のような作成手順でジョブネットワークを作成します。

■作成手順

1. APIトークンを取得する部品、Boxの操作を行う部品（HTTP接続部品）の前後関係で、部品をジョブネットワーク上に配置します。
2. APIトークンを取得する部品およびBoxの操作を行う部品のパラメータを設定します。設定の詳細は「7.3.2 APIトークンを取得する部品を利用する方法でのパラメータ設定」を参照してください。

■HTTP接続部品のカスタムパラメータ設定例については、「7.4 BOX_APIトークン取得部品」の利用例を参照してください。

■HTTP接続部品の仕様については、「3.1 HTTP接続部品共通仕様」を参照してください。

■利用できるBox APIについては、Boxのドキュメントを参照してください。

7.1.1.2. Boxの操作を行う部品を単体で利用する方法

Boxの操作を行う部品を単体で配置し、ジョブネットワークを作成します。

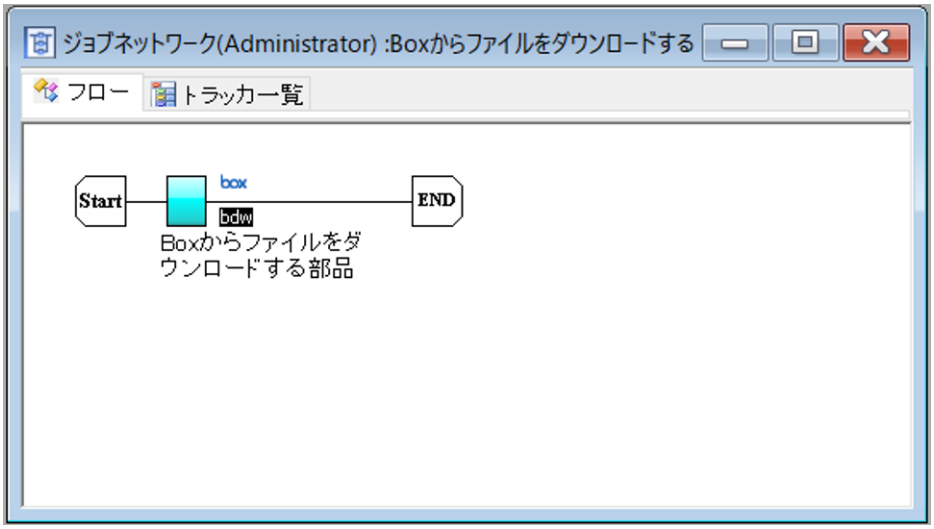


図7.3 Boxからファイルのダウンロードを行うジョブネットワーク

この方法でジョブネットワークを作成する場合、配置した各部品において、Boxにアクセスするための情報を設定し、Boxの操作を行います。

そのため、APIトークンを取得する部品を配置する必要がなく、Box連携部品をジョブネットワークに1つだけ配置する、または、Box連携部品同士を離れて配置する場合にジョブネットワークをシンプルにすることができます。

図7.3「Boxからファイルのダウンロードを行うジョブネットワーク」の例では、BOXダウンロード部品にBoxへアクセスするための情報を設定し、Boxからファイルをダウンロードを行う機能を実現しています。

以上から、Boxの操作を行う部品を単体で利用する方法は以下のようなケースで有用です。

■推奨ケース

- ジョブネットワーク上に単体でBoxの操作を行う部品を配置する場合

また、以下のような作成手順でジョブネットワークを作成します。

■作成手順

1. Boxの操作を行う部品をジョブネットワーク上に配置します。
2. 配置したBoxの操作を行う部品のパラメータを設定します。設定の詳細は「7.3.3 Boxの操作を行う部品を単体で利用する方法でのパラメータ設定」を参照してください。

7.2. 事前準備

Box連携部品で利用するパラメータをBoxで発行するために必要な事前準備について説明します。

本手順ではBoxの管理者アカウントでログインし、Boxアプリケーションを作成してパラメータを取得します。

本手順で取得するパラメータは、APIトークンを取得する部品を利用する方法におけるBOX_APIトークン取得部品、または、Boxの操作を行う部品を単体で利用する方法におけるBoxの操作を行う部品の必須パラメータとなります。そのため、ジョブネットワークの作成方法を問わず、本手順を実施してください。

本手順において、取得するパラメータを以下に記載します。各パラメータの詳細については、「7.3.1 Box連携部品共通で設定可能なパラメータ」を参照してください。

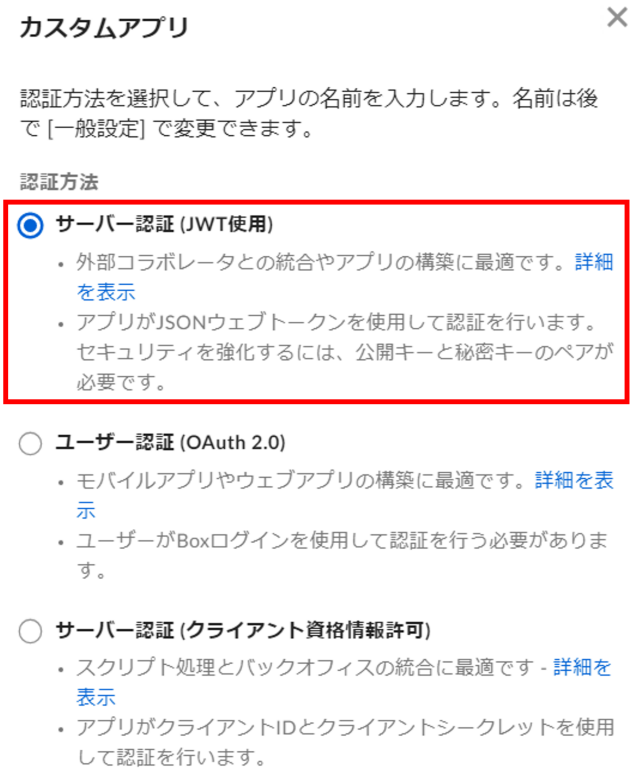
パラメータ名	パラメータ概要
head_kid	BoxアプリでRSA公開鍵を登録後に表示される公開キーID
privkey_filepath	Boxアプリで登録したRSA公開鍵と対になるRSA秘密鍵のファイルパス
client_id	BoxアプリのクライアントID
client_secret	Boxアプリのクライアントシークレット
payload_sub	BoxのエンタープライズIDまたはユーザーID

■Boxアプリケーションの作成

1. Boxの管理者アカウントでログインを行い、開発者コンソールを開きます。（赤枠を押下で開きます）

2. 開発者コンソールのマイアプリでアプリの新規作成を選び、カスタムアプリを新規作成します。（赤枠を押下で新規作成画面に遷移します）

3. 認証方式として「サーバー認証(JWT使用)」を選択してください。



■公開キーの追加について

アプリケーションの追加後、「構成」から公開キーの追加が出来ます。JobCenterからJWT認証を行う場合にはRSAキーペアを生成して、公開鍵をBoxに登録する必要があります。

また、生成した秘密鍵を格納したファイルパスは、Box連携部品共通で設定できるパラメータとして、privkey_filepathに指定することができます。

1.「公開キーを追加」のボタンを押下してください。公開キーを入力するウインドウが表示されます。

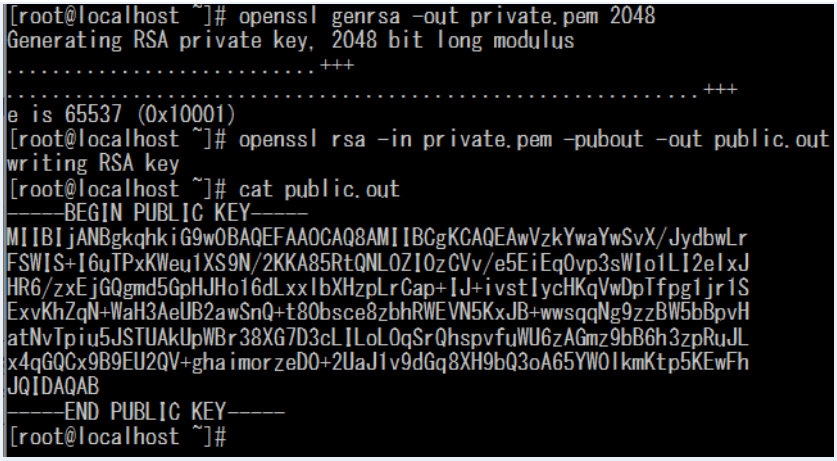
2. 表示された「新しい公開キーを追加」のウインドウに作成したRSAキーペアの公開鍵を登録します。

   登録後に公開キーIDが発行されます。

   Box連携部品共通で設定できるパラメータとして、この公開キーIDをhead_kidに指定することができます。

（ご参考）RSAキーペアを作るコマンドの一例

RSAキーペアはopensslコマンド等で生成可能です。作成した実際のコンソール画面の一例を記載します。



opensslコマンドで作成したpublic.out（公開鍵）の出力内容をBox側に登録してください。また、Box連携部品にはprivate.pem（秘密鍵）のファイルパスが必要となります。紛失しないように注意してください。

Box側でRSA鍵を作成することも可能ですが、その際にダウンロードする秘密鍵にはパスフレーズが設定されています。Box連携部品はパスフレーズの秘密鍵を利用することはできませんので、別途opensslコマンドで「パスフレーズを解除」を行ってからご利用ください。

■作成したBoxアプリの設定

作成したアプリの「構成」から必要な設定を行います。以下の表に従い、必要な設定を行ってください。

項目	設定内容	設定詳細
アプリケーションアクセス	Enterpriseを指定	既存ユーザーへのアクセスが必要となるため、Enterpriseを指定してください。
アプリケーションスコープ	利用したい機能にチェックボックス有効化	利用するAPIの対象となるスコープを選択してください。（アップロードには書き込み権限が必要となります）
高度な機能	以下を有効化 ▪ as-userヘッダーを使用してAPI呼び出しを行う ▪ ユーザーアクセストークン生成する	ユーザーとしてAPIを実行できるよう、両方とも有効化を行ってください。

■アプリの承認

1. 管理コンソールから作成したカスタムアプリケーションの承認を行います。管理コンソールの「アプリ」を開きます。

2. 「アプリ」の上のバーメニューから「カスタムアプリマネージャ」を押下し、表示された項目の「アプリの追加」を押下します。

3. アプリの追加というダイアログが出てきたら、作成したアプリのAPIキーが求められます。このAPIキーは作成したBoxアプリのクライアントIDを指します。アプリ詳細画面のOAuth2.0資格情報からクライアントIDの文字列をコピーしてアプリの認証を行ってください。

   このクライアントIDは、Box連携部品共通で設定できるパラメータとして、client_idに指定することができます。

   また、アプリ詳細画面のOAuth2.0資格情報にクライアントIDと合わせて表示されているクライアントシークレットは、Box連携部品共通で設定できるパラメータとして、client_secretに指定することができます。

■エンタープライズIDまたはユーザーIDの確認

エンタープライズIDまたはユーザーIDについては、以下の方法で確認することができます。

- エンタープライズID

  1. 管理コンソールの「アカウントと請求」を開きます。

  2. アカウント情報に表示されているEnterprise IDを確認してください。

- ユーザーID

  1. Boxへログイン後のページにおいて、右上にあるアカウントのアイコンを押下し、「アカウント設定」を開きます。

  2. アカウントの詳細に表示されているアカウントIDを確認してください。

7.3. Box連携部品共通の仕様

Box連携部品で共通となる仕様について説明します。

7.3.1. Box連携部品共通で設定可能なパラメータ

以下に記載するカスタムパラメータは、全部品で共通的に設定可能です。

一部のパラメータは事前準備にて取得した値を設定します。

APIトークンを取得する部品を利用する方法におけるBOX_APIトークン取得部品、または、Boxの操作を行う部品を単体で利用する方法におけるBoxの操作を行う部品では、これらのパラメータの設定が必須となります。

必須	変数	データ型	説明	設定例
△	privkey_filepath	文字列	Boxアプリで登録したRSA公開鍵と対になるRSA秘密鍵のファイルパスを指定してください。	C:\tmp\boxpem.key
△	head_kid	文字列	BoxアプリでRSA公開鍵を登録後に表示される公開キーIDを指定してください。	abcde……(文字列を設定)
△	payload_iss	文字列	サービスのクライアントIDを指定してください。	abcde……(文字列を設定)
△	payload_sub	文字列	対象となるエンタープライズIDかユーザーIDを指定してください。	12345678……(数値のIDを設定)
△	payload_box_sub_type	リスト	Boxのユーザー種別(enterprise or user)を指定してください。	enterprise
△	client_id	文字列	Boxアプリのクライアントを指定してください。	abcde……(文字列を設定)
△	client_secret	文字列	Boxアプリのクライアントシークレットを指定してください。	abcde……(文字列を設定)
	proxy	文字列	Boxへの接続にプロキシサーバを利用する場合に指定します。 "<プロトコル名>://<サーバ名>:<ポート番号>" の形式で指定してください。	http://example.com:8080

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

7.3.2. APIトークンを取得する部品を利用する方法でのパラメータ設定

以下に記載するカスタムパラメータは、BOX_APIトークン取得部品以外の部品で共通的に設定可能です。

APIトークンを取得する部品を利用する方法では、BOX_APIトークン取得部品に対してBox連携部品共通のパラメータとdbkeyパラメータの設定を行います。

この設定により、dbkeyで指定した暗号化DBのキーに取得したAPIトークンが設定されます。

後続のBoxの操作を行う部品では、以下に記載したカスタムパラメータに対して暗号化DBを参照することによってAPIトークンを設定し、Boxの操作を行うことができます。

パラメータdbkeyの詳細やdbkeyで指定した暗号化DBの参照例については、「7.4　BOX_APIトークン取得部品」を参照してください。

APIトークンを取得する部品を利用する方法におけるBoxの操作を行う部品では、このパラメータの設定が必須となります。

必須	変数	データ型	説明	設定例
△	box_apitoken	文字列	APIトークンの文字列を直接指定してください。 本指定がある場合、proxy以外のBox連携部品共通で設定可能なパラメータは利用されません。	abcde...(文字列を直接指定)

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

7.3.3. Boxの操作を行う部品を単体で利用する方法でのパラメータ設定

Boxの操作を行う部品を単体で利用する方法では、Boxの操作を行う部品に対して、Box連携部品共通で設定可能なパラメータを設定します。

7.3.4. ワイルドカード設定

BOXアップロード部品およびBOXダウンロード部品の一部パラメータではワイルドカードを利用できます。

ワイルドカードとして利用できる文字は以下になります。

利用可能文字	説明
*	0文字以上の任意の文字列と一致するパターン
?	任意の1文字と一致するパターン

7.4. BOX_APIトークン取得部品

7.4.1. 部品情報

■格納パス
/Root/System/Tools/Box

■定義名
box_gettoken

■表示アイコン

7.4.2. 部品説明

Box APIを実行するためのAPIトークンを取得します。

取得したAPIトークンをカスタムパラメータ「dbkey」で指定した暗号化DBのキーに格納します。

後続のジョブは暗号化DBからAPIトークンを参照して利用できます。たとえば、後続のジョブとしてHTTP接続部品の「HTTP-DELETE部品」を配置し、「ファイルの削除」を行うなど様々なBox APIの利用が可能となります。

本部品は、APIトークンを取得する部品を利用する方法におけるAPIトークンを取得する部品として利用します。

7.4.3. スクリプト

本部品にはスクリプトはありません。

7.4.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	dbkey	文字列	APIトークンを暗号化して格納する暗号化DBのキーを指定してください。	testkey

上記のようにパラメータが指定されている場合、後続のジョブは環境変数やカスタムパラメータに暗号化パラメータ(キー名：testkey)を指定することで、APIトークンを取り出すことが出来ます。

カスタムパラメータに暗号化パラメータを入力する方法は「1.2.2.6.2 暗号化パラメータの利用」を参照してください。

7.4.5. 利用例

APIトークンを取得する部品を利用する方法における本部品の利用例を説明します。

7.4.5.1. BOX_APIトークン取得部品とHTTP接続部品の組み合わせ

Box連携部品で提供されている機能以外のBox APIは、BOX_APIトークン取得部品とHTTP接続部品を組み合わせることで利用することができます。

本利用例では、BOX_APIトークン取得部品とHTTP-DELETE部品を利用し、Box上の「ファイルA」を削除する例について説明します。

BOX_APIトークン取得部品、HTTP-DELETE部品の順で、部品を以下のように配置します。

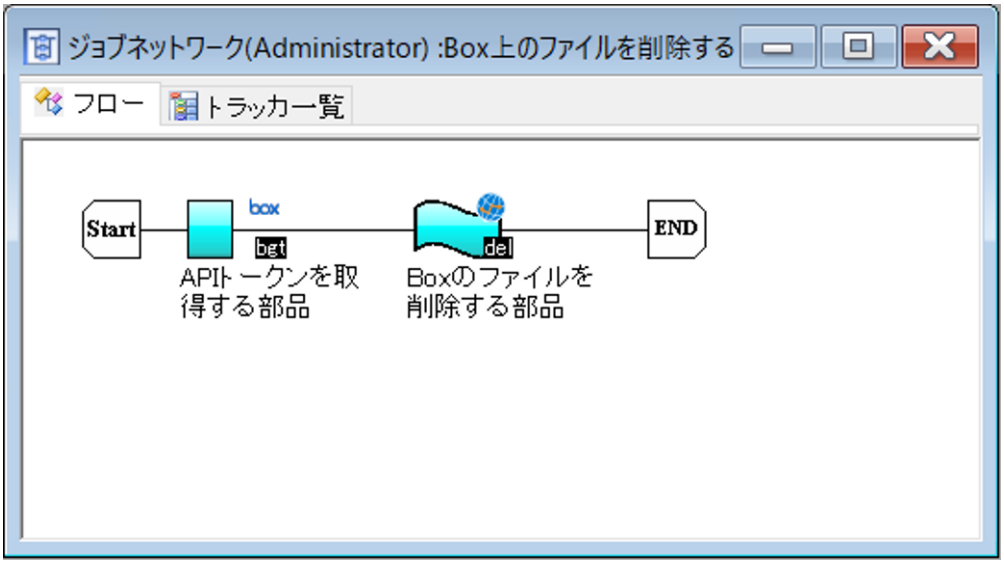


図7.4 Box上のファイルの削除を行うジョブネットワーク

BOX_APIトークン取得部品のカスタムパラメータについて、設定を行います。

本利用例では、dbkeyに「testkey」と指定し、取得したAPIトークンを暗号化DBのtestkeyに設定します。その他のBoxにアクセスするために必要なパラメータの詳細は「7.3.2　APIトークンを取得する部品を利用する方法でのパラメータ設定」を参照してください。

HTTP-DELETE部品のカスタムパラメータを以下のように指定します。

変数	設定値
authenticate	bearer
auth_token	$testkey$
url	https://api.box.com/2.0/files/<ファイルAのID>

このような設定を行うことで、暗号化DBのtestkeyに格納されたAPIトークンがauth_tokenに設定されます。

7.4.6. 実行結果

■APIトークンの取得に成功した場合

APIトークンの取得に成功した場合は終了値0となり、暗号化DBのdbkeyで指定したキーにAPIトークンを暗号化して格納します。

■必須パラメータが指定されていない場合

必須パラメータが指定されていない場合は終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗した場合は終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■暗号化DBの書き込みに失敗した場合

暗号化DBへの更新に失敗した場合は終了値3となり、エラーメッセージが標準エラーに出力されます。

7.4.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
get_APItoken request error.	APIトークン取得のAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
set Access_token error.	暗号化DBへの更新が失敗した場合に出力されます。

7.4.8. 注意事項

■暗号化DBに格納した情報は同一マシンからしか参照できません。後続部品で暗号化DBに格納されたAPIトークンを利用する場合には同一マシンのカスタムキューに投入して参照してください。

■発行されたAPIトークンには有効期限があります。有効期限が切れた場合に再度APIトークンの取得を行うなどの実装を行い、APIトークンの有効期限切れに注意してジョブネットワークを作成してください。

■カスタムパラメータdbkeyのキーに格納されるAPIトークンは重複が無いよう注意してください。同一ユーザーで同じキー名にAPIトークンを格納した場合、後から更新したトークンに上書きされます。（ジョブネットワーク毎、ユーザー毎でキー名を分けてお使いください）

■ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できません。

■Boxへ接続ができない環境から本部品を利用したAPIトークンの取得はできません。Boxへ接続ができることを確認した環境で利用してください。

■「privkey_filepath」で指定したRSA秘密鍵はパスフレーズが設定されていると利用できません。パスフレーズ解除を行ったRSA秘密鍵を指定してください。

7.5. BOXアップロード部品

7.5.1. 部品情報

■格納パス
/Root/System/Tools/Box

■定義名
box_upload

■表示アイコン

7.5.2. 部品説明

Boxへファイルをアップロードします。アップロードの方法として以下の2種類を提供しています。

■単一ファイルのアップロード

指定したひとつのローカルファイルをBoxへアップロードします。

■複数ファイルのアップロード

ローカルフォルダを指定して、複数のファイルをBoxへアップロードします。アップロードするファイル名の条件やサブフォルダを含めてアップロードするかを指定できます。

アップロード時に必要なBox上のサブフォルダは自動的に作成されます。

どちらの方法でアップロードを行うかは後述のカスタムパラメータ設定により決定されます。

また、アップロード先のBoxフォルダを指定する方法は以下の3種類を提供しています。

■フォルダID指定

box_parent_folderidでアップロード先フォルダのIDを指定します。IDがすでにわかっている場合などに利用可能です。

■フォルダID+サブフォルダ名指定

親フォルダと配下のサブフォルダ名を指定します。親フォルダのIDをbox_parent_folderid、サブフォルダ名をbox_sub_foldernameで指定します。

■フォルダパス指定

box_full_folderpathで指定したBoxのフォルダパスに対してファイルのアップロードを行います。

7.5.3. スクリプト

本部品にはスクリプトはありません。

7.5.4. カスタムパラメータ設定

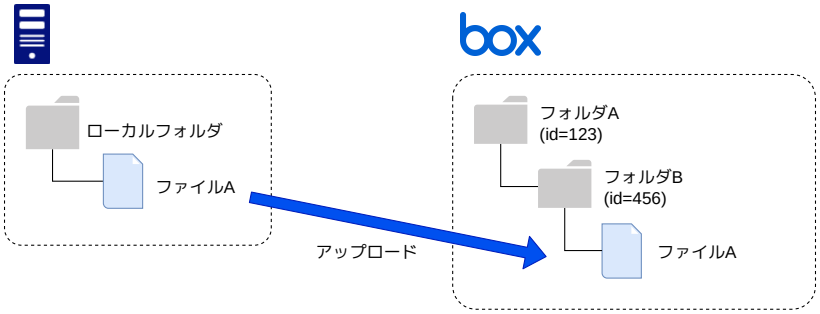
必須	変数	データ型	説明	設定例
	box_uploadfilename	文字列	アップロード後のファイル名を指定してください。ローカルファイル名と異なっても問題ありません。	sample.pdf

必須	変数	データ型	説明	設定例
			wild_cardまたはrecursiveをtrueに設定して複数ファイルをアップロードする場合は指定できません。	
○	local_folderpath	文字列	アップロードをするファイルがあるフォルダのパスを指定してください。	C:\tmp \samplefolder
	local_filename	文字列	local_folderpathフォルダ内に存在するアップロードしたいファイル名を指定してください。recursiveがfalseの場合は必須となります。	uploadfile.pdf *.pdf uploadfile.*
	wild_card	リスト	local_filenameに指定するファイル名にワイルドカードを利用する場合はtrueを指定してください。	true false
	recursive	リスト	local_folderpath配下のサブフォルダを再帰的にアップロード対象とする場合はtrueを指定してください。	true false
	exclude_file	文字列	アップロード対象から除外したいファイルを指定します。wild_cardにtrueを指定した場合はワイルドカードが利用可能です。また、複数指定する場合はカンマ区切りで指定してください。	exclude_file.txt file1.txt,file2.txt *.zip *.png,*.jpg
	box_parent_folderid	文字列	ファイルをアップロードする先のBoxのフォルダIDを指定してください。	0
	box_sub_foldername	文字列	box_parent_folderidで指定した配下のフォルダにファイルを置きたい場合にフォルダ名を指定してください。	sample_folder
	box_full_folderpath	文字列	ファイルをアップロードする先のBoxのフォルダパスを指定してください。box_folderidの指定より、本パラメータでの指定が優先されます。	/sample_folder/ folder_A
	quiet_mode	リスト	複数ファイルのアップロード時にアップロード元とアップロード先のファイルパスを標準出力へ出力しないようにする場合trueを指定してください。	true false

7.5.5. 利用例

7.5.5.1. 単一ファイルのアップロード

ローカルの「ファイルA」をBoxのフォルダ「/フォルダA/フォルダB」配下にアップロードする例です。



アップロード先フォルダの指定方法は3種類あります。それぞれの場合におけるカスタムパラメータの指定方法を説明します。

アップロード先のフォルダIDを利用する場合はカスタムパラメータを以下のように指定します。

変数	設定値
local_folderpath	(ローカルフォルダの絶対パス)
local_filename	ファイルA
box_parent_folderid	456
box_filename	ファイルA

アップロード先フォルダの名前と親フォルダのIDを利用する場合はカスタムパラメータを以下のように指定します。

変数	設定値
local_folderpath	(ローカルフォルダの絶対パス)
local_filename	ファイルA
box_parent_folderid	123
box_sub_foldername	フォルダB
box_filename	ファイルA

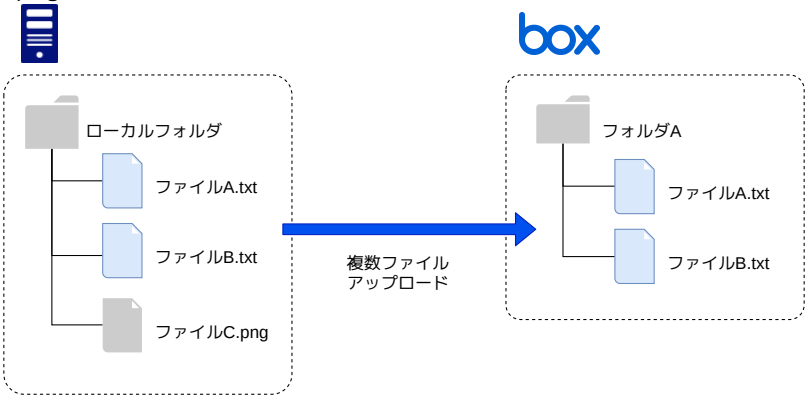
アップロード先フォルダのパスを利用する場合はカスタムパラメータを以下のように指定します。

変数	設定値
local_folderpath	(ローカルフォルダの絶対パス)
local_filename	ファイルA
box_full_folderpath	/フォルダA/フォルダB
box_filename	ファイルA

7.5.5.2. 複数ファイルのアップロード

ローカルフォルダ配下の複数のファイルをBoxのフォルダ「/フォルダA」配下にアップロードする例です。

この例では拡張子が「.png」であるファイルをアップロード対象から除外するものとしています。



この場合のカスタムパラメータは以下のように指定します。

変数	設定値
wild_card	true
local_folderpath	(ローカルフォルダの絶対パス)
local_filename	*

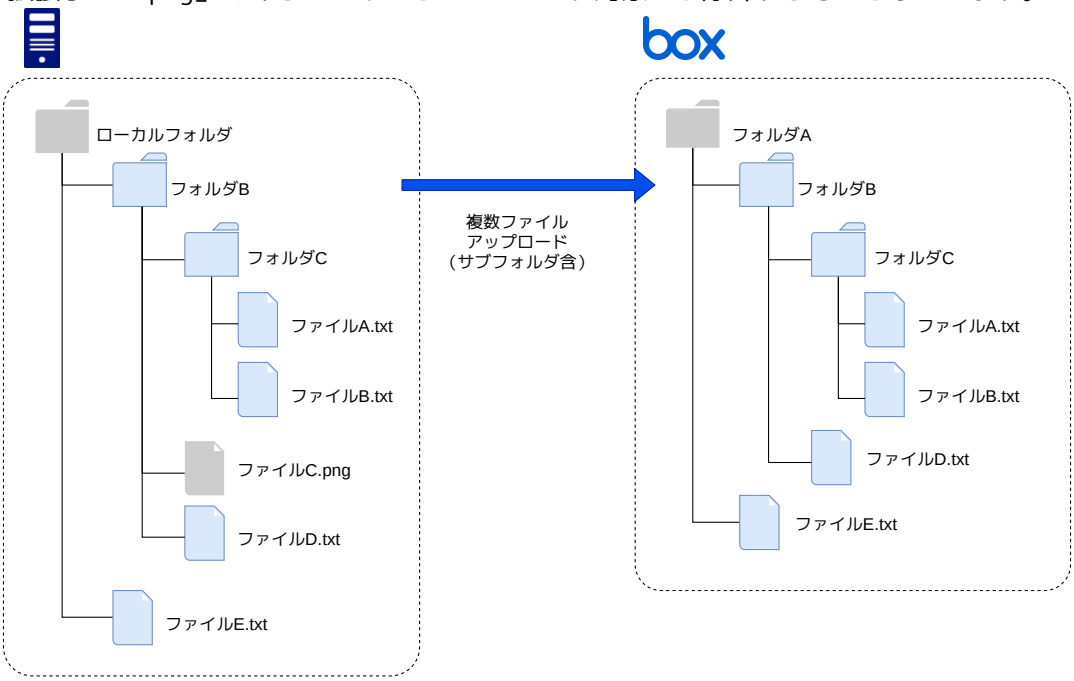
変数	設定値
exclude_file	*.png
box_full_folderpath	/フォルダA

アップロード先フォルダの指定にフォルダIDを利用することもできます。単一ファイルアップロードの場合と同様のため設定例は省略します。

7.5.5.3. サブフォルダを含む複数ファイルのアップロード

ローカルフォルダ配下の複数のファイルおよびサブフォルダをBoxのフォルダ「/フォルダA」配下にアップロードする例です。

この例では拡張子が「.png」であるファイルをアップロード対象から除外するものとしています。



この場合のカスタムパラメータは以下のように指定します。

変数	設定値
recursive	true
wild_card	true
local_folderpath	(ローカルフォルダの絶対パス)
local_filename	*
exclude_file	*.png
box_full_folderpath	/フォルダA

アップロード先フォルダの指定にフォルダIDを利用することもできます。単一ファイルアップロードの場合と同様のため設定例は省略します。

7.5.6. 実行結果

■アップロードに成功した場合

アップロードに成功した場合は終了値0となります。

単一ファイルのアップロードの場合は成功を表すメッセージが標準出力に出力されます。

複数ファイルのアップロードの場合はファイルごとにアップロード元とアップロード先のファイルパスが標準出力に出力されます。

■必須パラメータが指定されていない場合

必須パラメータが指定されていない場合は終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗した場合は終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■アップロード処理に失敗した場合

アップロード処理に失敗した場合は終了値3となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

7.5.7. 主要メッセージ

メッセージ	内容
upload success.	単一ファイルのアップロードに成功した場合に出力されます。
from: [ローカルファイルパス] to: [Boxファイルパス]	複数ファイルのアップロード時にファイルごとにアップロード元のファイルパスとアップロード先のファイルパスを出力します。 出力が不要な場合、カスタムパラメータquiet_modeをtrueに設定することでメッセージを抑止できます。
Info: No files were uploaded.	複数ファイルのアップロード時にアップロード対象のファイルが一件も存在しなかった場合に出力します。
file is already exists, or the account has run out of disk space. file: [ファイルパス]	複数ファイルのアップロード時に、Boxフォルダ上に同名のファイルが存在していた場合や容量超過によりアップロード処理が失敗した場合に出力します。 このメッセージが出力された場合であってもアップロード処理自体は継続され、エラーとはなりません。
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
get_APItoken request error.	APIトークン取得のAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
get_Folders request error.	フォルダ情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
Error:Can't Find folderid from [box_full_folderpath].	box_full_folderpathで指定した名前のフォルダが存在していない場合に出力されます。 box_full_folderpathで指定したパスに誤りが無いか、確認してください。
Error:Can't Find folderid from [box_sub_foldername].	box_sub_foldernameで指定した名前のフォルダが存在していない場合に出力されます。 box_sub_foldernameで指定した名前に誤りが無いか、確認してください。またbox_parent_folderidで指定した親フォルダの配下にあるかも確認してください。

メッセージ	内容
Access token may be expired. Refresh access token and try again.	box_apitokenが未指定の場合かつアップロード処理中にアクセストークンの有効期限が切れた場合に出力されます。 このメッセージが表示された場合はアクセストークンの再取得を行いアップロード処理を継続します。
[XXX]. Try again.	Box　API実行時のサーバエラーが原因で処理をリトライする場合に出力されます。 [XXX]はAPI実行時に発生したエラー内容によって変化します。
The upload process retry limit has been exceeded.	アップロード処理のリトライ回数が上限に達してエラーとなった場合に出力されます。
upload request error.	アップロードAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。

7.5.8. 注意事項

■ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できません。

■単一ファイルアップロード時にすでに指定されたパスに同一ファイルが存在する場合エラーとなります。

■複数ファイルのアップロード時に、Boxフォルダ上に同名ファイルが存在している場合やBoxアカウントの容量超過によりアップロードに失敗した場合、当該ファイルのアップロードをスキップし部品自体はエラーとはしません。

アップロードに失敗したファイルごとに以下のエラーメッセージを出力して残りアップロード処理を継続します。

```
file is already exists, or the account has run out of disk space.
```

7.6. BOXダウンロード部品

7.6.1. 部品情報

■格納パス
/Root/System/Tools/Box

■定義名
box_download

■表示アイコン

7.6.2. 部品説明

指定したファイルをBoxからダウンロードします。ダウンロードの方法として以下の2種類を提供しています。

■単一ファイルのダウンロード

Boxから指定したひとつのファイルをダウンロードします。

■複数ファイルのダウンロード

Box上のフォルダを指定して、複数のファイルをダウンロードします。ダウンロードするファイル名の条件やサブフォルダを含めてダウンロードするかを指定できます。

ダウンロード時に必要なサブフォルダは自動的に作成されます。

どちらの方法でダウンロードを行うかは後述のカスタムパラメータ設定により決定されます。

また、ダウンロード対象のファイルを指定する方法としては以下の2種類を提供しています。

■ファイルID指定

download_fileidで指定したファイルにたいしてダウンロードを行います。単一ファイルのダウンロードの場合のみ利用可能です。

■Boxパス指定

box_download_folderpathで指定したパスの中にあるbox_download_filenameで指定したファイルに対してダウンロードを行います。

7.6.3. スクリプト

本部品にはスクリプトはありません。

7.6.4. カスタムパラメータ設定

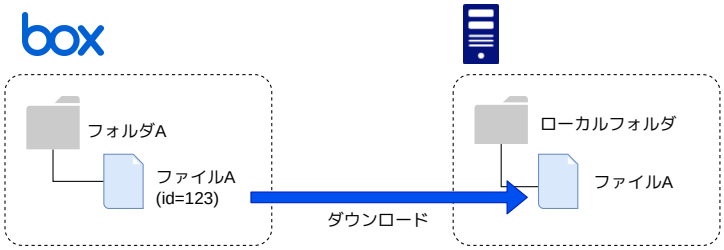
必須	変数	データ型	説明	設定例
○	local_folderpath	文字列	ダウンロード後に格納するローカルのフォルダパスを指定してください。	C:\tmp
	local_filename	文字列	ダウンロードしたファイルにつけるファイル名を指定してください。未指定の場合にはBox側で設定されている名前を取得して設定します。	download_output.pdf

必須	変数	データ型	説明	設定例
			wild_cardまたはrecursiveにtrueを設定して複数ファイルをダウンロードする場合は指定できません。	
	box_download_fileid	文字列	ダウンロードするBoxのファイルIDを指定してくださ い。box_download_folderpathの指定より本パラメータが優先されます。 ファイルID指定の場合は必須です。このパラメータまたは(box_download_folderpathとbox_download_filename)のどちらかが必須です。	0123…（数値）
	box_download_folderpath	文字列	ダウンロードするファイルが配置されているフォルダのパスを記載してください。 box_download_fileidを利用する場合は指定できません。	/sample_folder/folder_A
	box_download_filename	文字列	box_download_folderpathで指定したフォルダ内に存在するダウンロードしたいファイル名を指定してください。 wild_cardをtrueに設定した場合はワイルドカードを利用可能です。 box_download_fileidを利用する場合は指定できません。	download.pdf download.*
	wild_card	リスト	box_download_filenameに指定するファイル名にワイルドカードを利用する場合はtrueを設定してください。	true false
	recursive	リスト	box_download_folderpath配下のサブフォルダを再帰的にダウンロード対象とする場合はtrueを指定してくださ い。	true false
	exclude_file	文字列	ダウンロード対象から除外したいファイルを指定します。wild_cardにtrueを指定した場合はワイルドカードが利用可能です。また、複数指定する場合はカンマ区切りで指定してください。	exclude_file.txt file1.txt,file2.txt *.zip *.png,*.jpg
	quiet_mode	リスト	複数ファイルのダウンロード時にダウンロード元とダウンロード先のファイルパスを標準出力へ出力しないようにする場合trueを指定してください。	true false

7.6.5. 利用例

7.6.5.1. 単一ファイルのダウンロード

Boxのフォルダ「/フォルダA」配下のファイル「ファイルA」をローカルフォルダにダウンロードする例です。



ダウンロード対象のファイルをファイルIDで指定する場合は以下のようにカスタムパラメータを指定します。

変数	設定値
box_download_fileid	123
local_folderpath	(ローカルフォルダの絶対パス)
local_filename	ファイルA

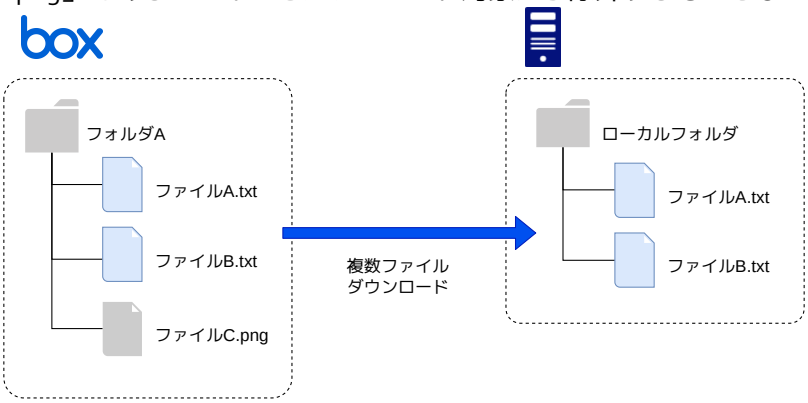
ダウンロード対象のファイルを親フォルダのフォルダパスとファイルで指定する場合は以下のようにカスタムパラメータを指定します。

変数	設定値
box_download_folderpath	/フォルダA
box_download_filename	ファイルA
local_folderpath	(ローカルフォルダの絶対パス)
local_filename	ファイルA

7.6.5.2. 複数ファイルのダウンロード

Boxのフォルダ「/フォルダA」配下のファイルをローカルフォルダ配下にダウンロードする例です。

この例では拡張子が「.png」であるファイルをダウンロード対象から除外するものとしています。



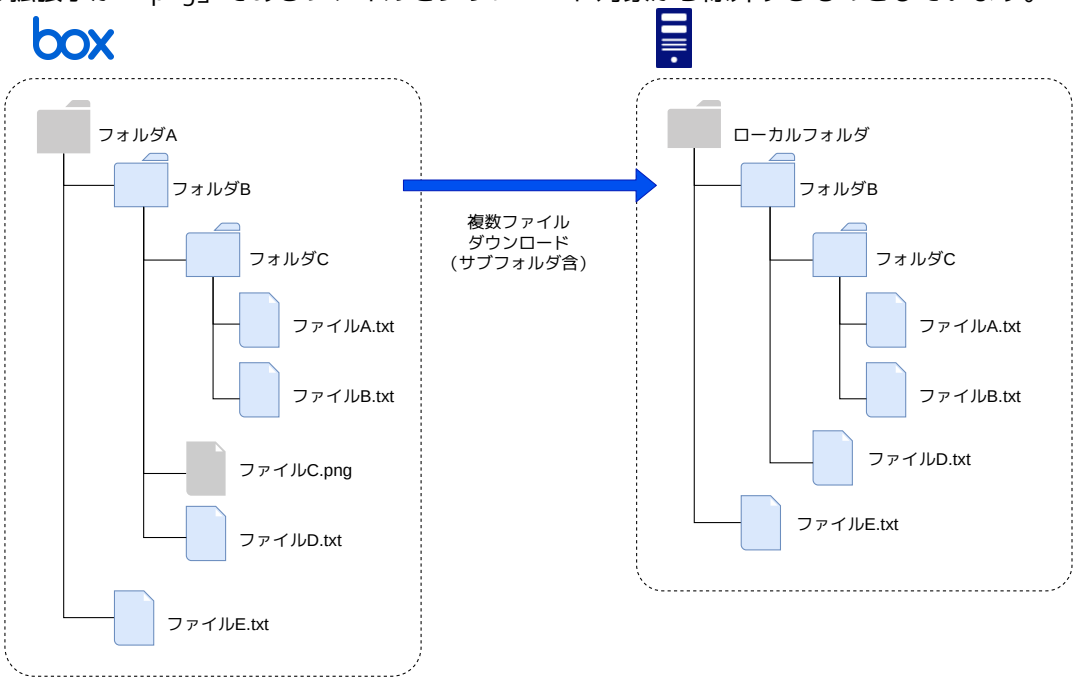
この場合のカスタムパラメータは以下のように指定します。

変数	設定値
wild_card	true
box_download_folderpath	/フォルダA
box_download_filename	*
exclude_file	*.png
local_folderpath	(ローカルフォルダの絶対パス)

7.6.5.3. サブフォルダを含む複数ファイルのダウンロード

Boxフォルダ「/フォルダA」配下のファイルおよびサブフォルダをローカルフォルダ配下にダウンロードする例です。

この例では拡張子が「.png」であるファイルをダウンロード対象から除外するものとしています。



この場合のカスタムパラメータは以下のように指定します。

変数	設定値
recursive	true
wild_card	true
box_download_folderpath	/フォルダA
box_download_filename	*
exclude_file	*.png
local_folderpath	(ローカルフォルダの絶対パス)

7.6.6. 実行結果

■ダウンロードに成功した場合

ダウンロードに成功した場合は終了値0となります。

単一ファイルのダウンロードの場合は成功を表すメッセージが標準出力に出力されます。

複数ファイルのダウンロードの場合はファイルごとにダウンロード元とダウンロード先のファイルパスが標準出力に出力されます。

■必須パラメータが指定されていない場合

必須パラメータが指定されていない場合は終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗した場合は終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■ダウンロード処理に失敗した場合

ダウンロード処理に失敗した場合は終了値3となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

7.6.7. 主要メッセージ

メッセージ	内容
box_download success.	単一ファイルのダウンロードに成功した場合に出力されます。
from: [Boxファイルパス] to: [ローカルファイルパス]	複数ファイルのダウンロード時にファイルごとにダウンロード元のファイルパスとダウンロード先のファイルパスを出力します。 出力が不要な場合、カスタムパラメータquiet_modeをtrueに設定することでメッセージを抑止できます。
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
get_APItoken request error.	APIトークン取得のAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
get_Folders request error.	フォルダ情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
Error:Can't Find folderid from [box_download_folderpath].	box_download_folderpathで指定した名前のフォルダが存在していない場合に出力されます。 box_download_folderpathで指定したパスに誤りが無いか、確認してください。
Error:Can't Find folderid from [box_download_filename].	box_download_filenameで指定した名前のフォルダが存在していない場合に出力されます。 box_download_filenameで指定した名前に誤りが無いか、確認してください。
get_FileName request error.	ファイル情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
Access token may be expired. Refresh access token and try again.	box_apitokenが未指定の場合かつダウンロード処理中にアクセストークンの有効期限が切れた場合に出力されます。 このメッセージが表示された場合はアクセストークンの再取得を行いダウンロード処理を継続します。
[XXX]. Try again.	Box　API実行時のサーバエラーが原因で処理をリトライする場合に出力されます。 [XXX]はAPI実行時に発生したエラー内容によって変化します。
The download process retry limit has been exceeded.	ダウンロード処理のリトライ回数が上限に達してエラーとなった場合に出力されます。
box_download request error.	ダウンロードAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
fileopen error. check download_filename.	ダウンロードしたファイルの保存に失敗した場合に出力されます。後続に原因のメッセージが出力されます。 ローカルストレージの空き容量が残っているかなど、ローカルストレージの確認を行ってください。

メッセージ	内容
redirect_download request error.	リダイレクトアドレスからファイルのデータを読みに失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。

7.6.8. 注意事項

■ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できません。

■BoxのファイルIDは随時付与されますので、同一ファイル名、同一フォルダにアップロードしてもBoxのIDは異なりますので注意してください。

7.7. コンテンツID取得部品

7.7.1. 部品情報

■格納パス
/Root/System/Tools/Box

■定義名
box_getContentID

■表示アイコン

7.7.2. 部品説明

指定したBoxのパスからコンテンツIDを標準出力にフォーマット出力します。Box APIを実行する際にはフォルダIDかファイルIDが必要になりますが、Boxのフォルダパス、ファイルパスからIDを取得するAPIは存在しません。

本部品はその課題を解決する部品として、フォルダIDやファイルIDをスクリプトで出力し、他のアプリケーションや環境変数にいれて後続のジョブで利用できます。

2種類の方法でコンテンツIDを取得できます。

■フォルダパス指定

box_folderpathで指定したフォルダのIDを出力します。

■フォルダパス指定＋ファイル名指定

box_folderpathで指定したパスの中にあるbox_filenameで指定したファイルのIDを出力します。

7.7.3. スクリプト

標準出力へ実行結果を出力するためのテンプレートをMustache言語形式で記述します。

Boxのコンテンツを参照するには{{{boxContentID}}}を記述します。この{{{boxContentID}}}の部分は指定したコンテンツIDに置き換わり表示されます。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

7.7.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	box_folderpath	文字列	IDを取得したいフォルダパスを指定してください。box_filename未指定時にはフォルダIDが、指定時にはbox_filenameで指定したファイルIDがフォーマット出力されます。	/sample_folder/folder_A
	box_filename	文字列	box_folderpathで指定したフォルダ内に存在するファイル名を指定してください。	download.pdf

7.7.5. 実行結果

■ID取得に成功した場合

ID取得に成功した場合は終了値0となり、スクリプトに記述したテンプレートを適用した内容が標準出力に出力されます。

■必須パラメータが指定されていない場合

必須パラメータが指定されていない場合は終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗した場合は終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■フォルダID取得APIに失敗した場合

アップロード先のフォルダIDを取得するAPIの実行に失敗した場合は終了値3となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■指定したBoxコンテンツが存在しなかった場合

box_folderpathで指定したフォルダ名やbox_filenameで指定したファイル名が存在しなかった場合、終了値4となりエラーメッセージが標準エラーに出力されます。

7.7.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
get_APItoken request error.	APIトークン取得のAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
get_Folders request error.	フォルダ情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
Error:Can't Find folderid from [box_folderpath].	box_folderpathで指定した名前のフォルダが存在していない場合に出力されます。 box_folderpathで指定したパスに誤りが無いか、確認してください。
Error:Can't Find fileid from [box_filename].	box_filenameで指定した名前のファイル名が存在していない場合に出力されます。 box_filenameで指定した名前に誤りが無いか、確認してください。

7.7.7. 注意事項

■ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できません。

■参照するBoxフォルダパスの階層の数だけBox APIを実行しますので、Boxの階層が深い場合にはAPIの実行回数が多くなります。

Box APIの実行回数は契約されているライセンスの種類により制限がある場合があります。ライセンスの詳細などはご契約されているBox社のサポートまでお問い合わせください。

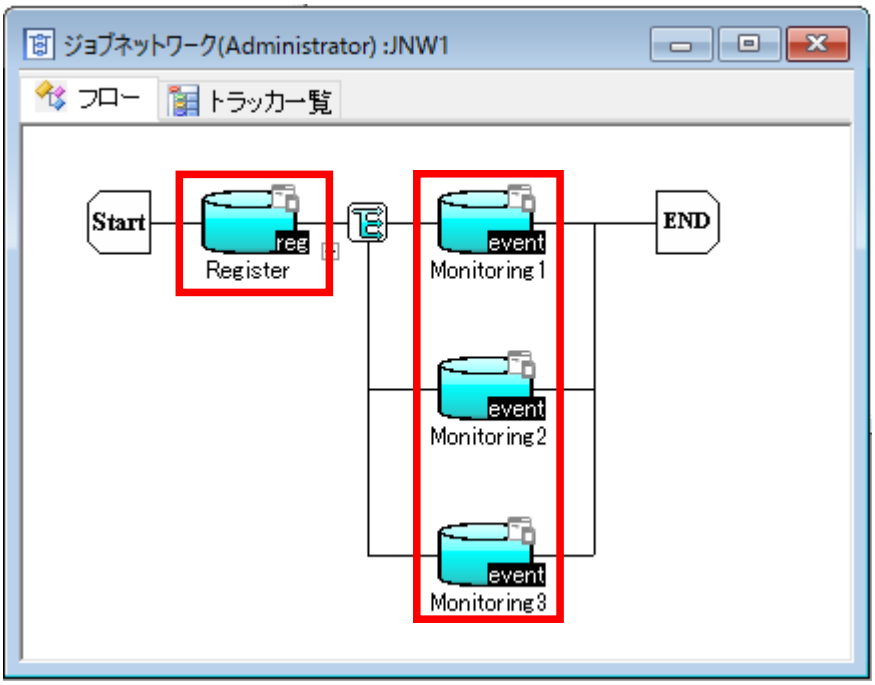
8. SystemManager G連携部品

本章ではSystemManager G連携部品について説明します。

SystemManager G連携部品は、NECのWebSAM SystemManager G(以下、SystemManager Gと記します)と連携する機能を提供しています。

8.1. メッセージ監視機能概要

メッセージ監視機能は、SystemManager Gに登録されるメッセージを監視し、条件に一致するメッセージを検出すると、JobCenterに通知する機能です。



ジョブネットワークでは、上記例の、左側のメッセージ監視登録部品と、右側のメッセージ待ち合わせ部品の2種類を配置します。

アイコン	名前	説明
	メッセージ監視登録部品	SystemManager Gに対して、メッセージ監視の条件登録と開始をリクエストします。 監視条件登録では、メッセージ待ち合わせ部品から、「filter.」で始まる名前のカスタムパラメータを収集して登録します。 詳細は、「8.3 メッセージ監視登録部品」を参照してください。
	メッセージ待ち合わせ部品	SystemManager Gから、条件に一致するメッセージが通知されるのを待ち合わせます。 詳細は、「8.4 メッセージ待ち合わせ部品」を参照してください。

8.1.1. 事前準備

SystemManager G連携部品による、メッセージ監視機能を利用するためには、JobCenter、SystemManager G、ともに事前準備が必要です。

8.1.1.1. Jobcenterの準備

SystemManager Gからのメッセージ通知を受信するため、JobCenter MG/SV専用のHTTPデーモンのjcresコマンドを起動します。

■SSLサーバ証明書の設定

httpsプロトコルによる通知を受信するため、jcresコマンドの設定ファイルに、SSL証明書の設定を記載します。

- SystemManager Gとの通信では、httpsプロトコルによる通信を必ず有効にしてください。
- jcresコマンドで利用するための、SSL証明書を用意してください。

設定ファイルの詳細は　スタンダードモードの場合は　<スタンダードモード用コマンドリファレンス>の「3.27.3　設定ファイル」、　クラシックモードの場合は　<クラシックモード用コマンドリファレンス>の「3.30.3 設定ファイル」を参照してください。

設定ファイルのうち、httpsの通信に関連する設定は以下のとおりです。

パラメータ	説明	範囲	初期値
SSL	httpsプロトコルによる通信の有効/無効を設定します。 onに変更してください。	on / off	off
SSL.PORT	https受信ポートを設定します。	1～65535	50443
SSL.VERSION	https通信に使用するTLSバージョンの下限値を設定します。 TLSv1_2を設定した場合、TLS 1.2、および、TLS 1.3での通信が可能です。	TLSv1_2 / TLSv1_3	TLSv1_2
SSL.CERTIFICATE	SSLサーバ証明書のファイルパスを絶対パスで設定します。	絶対パス	-
SSL.CERTIFICATE_KEY	秘密鍵のファイルパスを絶対パスで設定します。	絶対パス	-
SSL.CA_CERTIFICATE	認証局証明書のファイルパスを絶対パスで設定します。	絶対パス	-
SSL.VERIFY_CLIENT	TLSのクライアント認証の有効/無効を設定します。 onを設定した場合、TLSの通信で、クライアント証明書が必須となります。	on / off	off

設定ファイルのサンプルは以下のとおりです。証明書のパスは例です。任意のファイルパスに配置してください。

```
#LOGDIR=/usr/spool/nqs/log
#PORT=50080
#IPADDRESS=0.0.0.0
SSL=on
#SSL.PORT=50443
#SSL.VERSION=TLSv1_2
SSL.CERTIFICATE=/usr/local/ssl/certs/server.crt
SSL.CERTIFICATE_KEY=/usr/local/ssl/certs/server.key
#SSL.CA_CERTIFICATE
#SSL.VERIFY_CLIENT
```

■jcresコマンドの起動

jcresコマンドを起動します。

SystemManager　　G側の設定ファイルで、JobCenterアドレスに${hostname}を使用する場合、jcresの設定ファイルのIPADDRESSには0.0.0.0を設定してください。

${hostname}については、「8.1.1.2 SystemManager Gの準備」を参照してください。

jcresコマンドの使用方法の詳細は スタンダードモードの場合は <スタンダードモード用環境構築ガイド>の17章 「jcresの使用方法」 、 クラシックモードの場合は <クラシックモード用環境構築ガイド>の18章「jcresの使用方法」 を参照してください。

8.1.1.2. SystemManager Gの準備

JobCenterにメッセージを通知するため、SystemManager GのManagerにて、通知先の情報を設定します。

メッセージ監視機能については、JobCenterと連携可能なSystemManager Gのバージョンは、11.0以上です。

手順の中のファイルパスはManagerのインストール先がデフォルトの場合です。実際のインストール先に読み替えてください。

SystemManager GのManagerが、Linuxの場合はrootユーザで、Windowsの場合は管理者ユーザで実行してください。

設定内容の詳細は、SystemManager G リファレンスガイドを参照してください。

8.1.1.2.1. ManagerがLinuxの場合

■メッセージストアの設定ファイル

以下の設定ファイルを更新します。

```
/opt/nec/pf/opm/manager/conf/msc_messagestore.properties
```

設定ファイルが存在しない場合、sampleディレクトリから設定ファイルをコピーして編集してください。

```
コピー元：/opt/nec/pf/opm/manager/conf/sample/msc_messagestore.properties
```

設定ファイルの最後に以下の設定を追記してください。

※1行目は、表記の都合により折り返しています。「・・・/messages」までが1行目です。

```
event.result_notification.access_point=https://(JobCenterアドレス):(JobCenterポート)/sysmgr/
events/messages
event.result_notification.retry_interval=1
event.retention_period=1440
```

項目	説明	例
JobCenterアドレス	以下の2通りの指定が可能です。 ■ 固定のIPアドレスやホスト名 ・ JobCenterのjcresコマンドのIPアドレスや、それに紐づいたホスト名を指定してください。 ・ ホスト名を指定する場合、DNSやhostsファイルで名前解決できる必要があります。	192.168.0.10 Server-A ${hostname}

項目	説明	例
	■ 置換文字列「${hostname}」 ・ 連携するJobCenterが複数ある場合など、通知先が固定でない場合に指定してください。 ・ この置換文字列を指定すると、メッセージ通知時、通知先JobCenterのホスト名(クラスタサイトの場合はサイト名)に自動的に置換されます。 ・ JobCenterのホスト名やサイト名は、DNSやhostsファイルによって名前解決できる必要があります。 ・ この置換文字列は、SystemManager Gのバージョンが12.0以上の場合に指定可能です。 ・ JobCenter側では、ホストや各サイト宛のメッセージ通知を1つのjcresコマンドで受信します。そのため、jcresコマンドの設定のIPアドレスには、0.0.0.0を指定してください。	
JobCenterポート	jcresコマンドのhttps受信ポートを指定してください。	50443

※jcresコマンドの設定の詳細は、「8.1.1.1 Jobcenterの準備」を参照してください。

■メッセージストアの再起動

RHEL 6 の場合

```
# service msc_messagestore stop
# service msc_messagestore start
```

RHEL 7/8 の場合

```
# systemctl stop msc_messagestore
# systemctl start msc_messagestore
```

8.1.1.2.2. ManagerがWindowsの場合

■メッセージストアの設定ファイル

以下の設定ファイルを更新します。

```
C:\Program Files\NEC\pf\opm\manager\conf\msc_messagestore.properties
```

設定ファイルが存在しない場合、sampleフォルダから設定ファイルをコピーして編集してください。

```
コピー元：C:\Program Files\NEC\pf\opm\manager\conf\sample\msc_messagestore.properties
```

設定ファイルの最後に以下の設定を追記してください。

※1行目は、表記の都合により折り返しています。「・・・/messages」までが1行目です。

```
event.result_notification.access_point=https://(JobCenterアドレス):(JobCenterポート)/sysmgr/
events/messages
event.result_notification.retry_interval=1
event.retention_period=1440
```

項目	説明	例
JobCenterアドレス	以下の2通りの指定が可能です。 ■ 固定のIPアドレスやホスト名 ・ JobCenterのjcresコマンドのIPアドレスや、それに紐づいたホスト名を指定してください。 ・ ホスト名を指定する場合、DNSやhostsファイルで名前解決できる必要があります。 ■ 置換文字列「${hostname}」 ・ 連携するJobCenterが複数ある場合など、通知先が固定でない場合に指定してください。 ・ この置換文字列を指定すると、メッセージ通知時、通知先JobCenterのホスト名(クラスタサイトの場合はサイト名)に自動的に置換されます。 ・ JobCenterのホスト名やサイト名は、DNSやhostsファイルによって名前解決できる必要があります。 ・ この置換文字列は、SystemManager Gのバージョンが12.0以上の場合に指定可能です。 ・ JobCenter側では、ホストや各サイト宛のメッセージ通知を1つのjcresコマンドで受信します。そのため、jcresコマンドの設定のIPアドレスには、0.0.0.0を指定してください。	192.168.0.10 Server-A ${hostname}
JobCenterポート	jcresコマンドのhttps受信ポートを指定してください。	50443

※jcresコマンドの設定の詳細は、「8.1.1.1 Jobcenterの準備」を参照してください。

■メッセージストアの再起動

[管理ツール]-[サービス]を開き、[SystemManager G MessageStore Service]を再起動してください。

8.2. ACOS監視機能概要

ACOS監視機能は、SystemManager　Gを通じてNEC　ACOS(以下、ACOSと記します)に対してコマンドを実行し、実行結果をJobCenterで監視する機能です。

SystemManager GのACOS監視機能を使用して、ACOSへのコマンド実行と実行結果メッセージの通知を行います。

メッセージ監視登録部品、ACOSジョブ実行部品、メッセージ待ち合わせ部品を使用します。

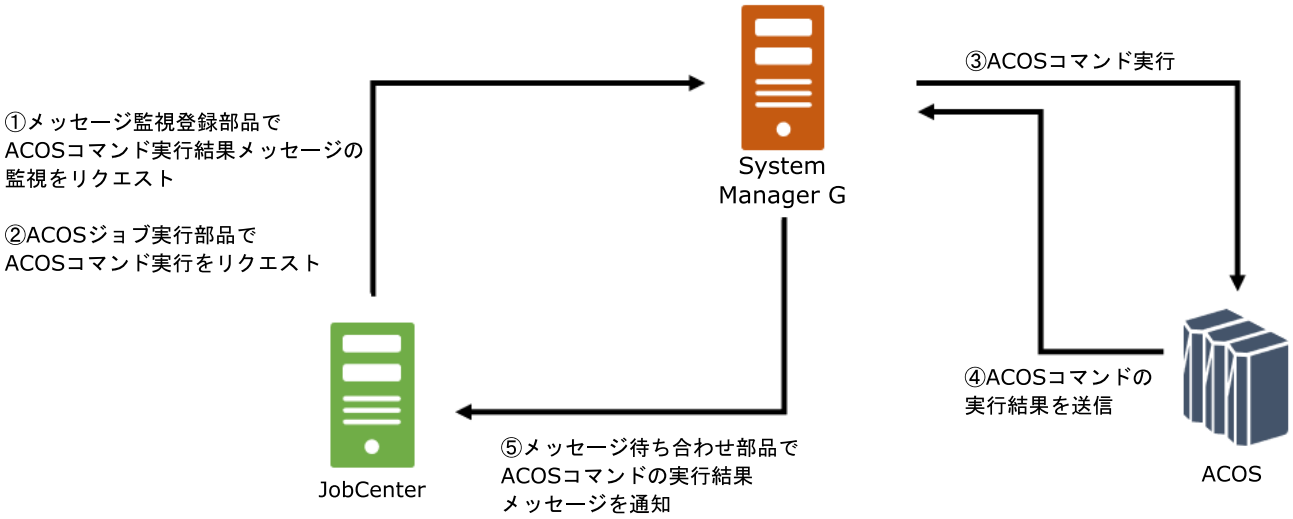


図8.1 ACOS監視機能イメージ

8.2.1. 事前準備

ACOS監視機能を利用するためには、JobCenter、SystemManager Gともに事前準備が必要です。

ACOS監視機能については、JobCenterと連携可能なSystemManager　Gのバージョンは、12.0.1以上です。

■JobCenter - SystemManager G間の連携の準備

ACOS監視機能では、ACOSコマンド実行結果の取得にメッセージ監視機能を使用します。

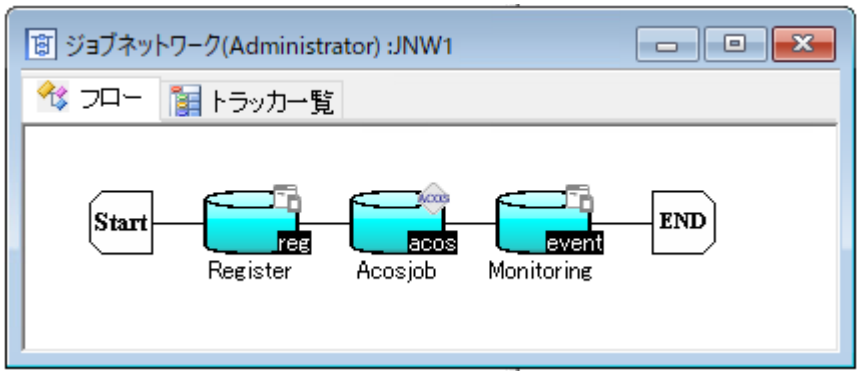
メッセージ監視機能の設定については、「8.1.1.1 Jobcenterの準備」および「8.1.1.2 SystemManager Gの準備」を参照してください。

■SystemManager G - ACOS間の連携の準備

SystemManager GのACOS監視機能で、ACOSのエージェントに対してコマンド実行後に出力されるメッセージを監視する設定が必要になります。

SystemManager GとACOSの連携の準備については、SystemManager Gのマニュアルを参照してください。

8.2.2. ジョブネットワークの設定



ジョブネットワークでは、上記例のようにメッセージ監視登録部品、ACOSジョブ実行部品、メッセージ待ち合わせ部品の順になるように配置します。

■1つのコマンドで複数のメッセージが出力される場合、ACOSでのコマンド実行結果をJobCenterで監視するには、メッセージ毎にメッセージ待ち合わせ部品が必要です。

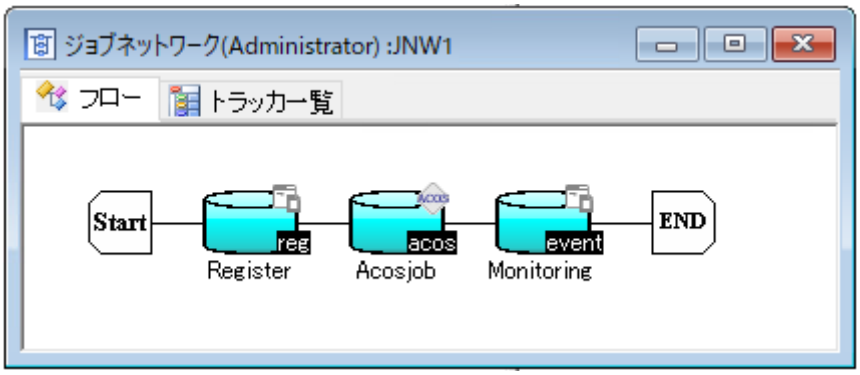
■ACOSジョブ実行部品のカスタムパラメータ「target」に複数の宛先を設定した場合、ACOSでのコマンド実行結果をJobCenterで監視するには、宛先毎にメッセージ待ち合わせ部品が必要です。

アイコン	名前	説明
	メッセージ監視登録部品 (Register)	図8.1「ACOS監視機能イメージ」の①の処理に該当します。 SystemManager Gに対して、ACOSコマンド実行結果メッセージの監視条件登録と監視開始をリクエストします。 監視条件登録では、メッセージ待ち合わせ部品から、「filter.」で始まる名前のカスタムパラメータを収集して登録します。 詳細は、「8.3　メッセージ監視登録部品」を参照してください。
	ACOSジョブ実行部品 (Acosjob)	図8.1「ACOS監視機能イメージ」の②の処理に該当します。 SystemManager　GのACOS監視機能を利用し、ACOSにコマンド実行をリクエストします。 ACOSコマンド実行時の照会画面の内容を、出力結果に表示します。詳細は、「8.5　ACOSジョブ実行部品」を参照してください。
	メッセージ待ち合わせ部品 (Monitoring)	図8.1「ACOS監視機能イメージ」の⑤の処理に該当します。 SystemManager Gから、条件に一致するACOSコマンド実行結果メッセージが通知されるのを待ち合わせます。 ACOSからSystemManager　Gに通知されたメッセージを受け取って出力します。詳細は、「8.4　メッセージ待ち合わせ部品」を参照してください。

8.2.2.1. カスタムパラメータ設定例

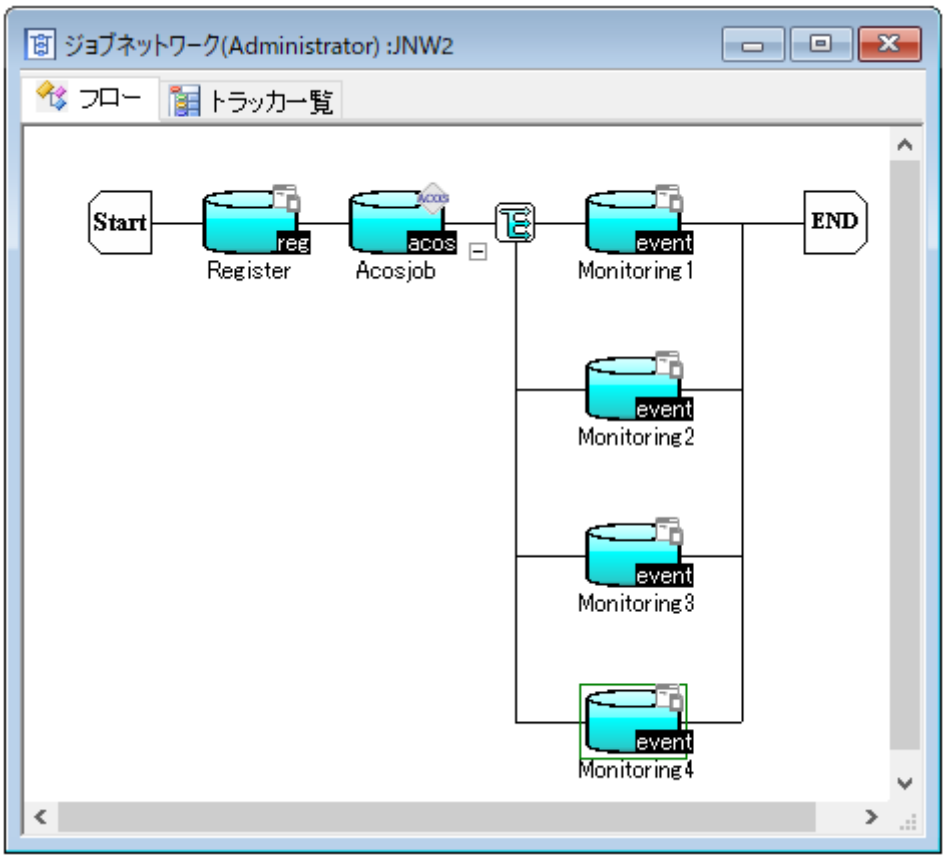
以下に、各部品の主なカスタムパラメータの設定例を示します。

■パターン1：1つの宛先に1つのメッセージが出力される場合



部品	カスタムパラメータ	値	説明
ACOSジョブ実行部品 (Acosjob)	command	RJ SYSMGRG_MSGSEND_ TEST01 IL=DIOS.OPM.MYSL	ACOSで実行するコマンドを設定します。
	target	REGION1@LCMP1:HOST1	コマンドを実行するACOSの物理ホスト名/論理システム名を設定します。
メッセージ待ち合わせ部品 (Monitoring)	filter.message_id	TEST01	コマンド実行時に出力され、SystemManager　Gで監視されるメッセージのIDを指定します。
	filter.node_name	HOST1	ACOSのホスト名を設定します。
	filter.message_text	TEST MESSAGE 1	ACOSコマンドの1つ目の実行結果メッセージに含まれる文字列を設定します。

■パターン2：2つの宛先に2つのメッセージが出力される場合



部品	カスタムパラメータ	値	説明
ACOSジョブ実行部品 (Acosjob)	command	DIOS OP DM LSNUM	ACOSで実行するコマンドを設定します。
	target	REGION1@LCMP1:HOST1, REGION1@LCMP1:HOST2	コマンドを実行するACOSの物理ホスト名/論理システム名を設定します。
メッセージ待ち合わせ部品1 (Monitoring1)	filter.message_id	DIOO12	コマンド実行時に出力され、SystemManager Gで監視されるメッセージのIDを指定します。
	filter.node_name	HOST1	1つ目のACOSのホスト名を設定します。
	filter.message_text	COMMAND ACCEPTED C=DM	ACOSコマンドの1つ目の実行結果メッセージに含まれる文字列を設定します。
メッセージ待ち合わせ部品2 (Monitoring2)	filter.message_id	DIOOL1	コマンド実行時に出力され、SystemManager Gで監視されるメッセージのIDを指定します。
	filter.node_name	HOST1	1つ目のACOSのホスト名を設定します。
	filter.message_text	LOGICAL SYSTEM IS NOT EXIST	ACOSコマンドの2つ目の実行結果メッセージに含まれる文字列を設定します。

部品	カスタムパラメータ	値	説明
メッセージ待ち合わせ部品3 (Monitoring3)	filter.message_id	DIOO12	コマンド実行時に出力され、SystemManager　Gで監視されるメッセージのIDを指定します。
	filter.node_name	HOST2	2つ目のACOSのホスト名を設定します。
	filter.message_text	COMMAND ACCEPTED C=DM	ACOSコマンドの1つ目の実行結果メッセージに含まれる文字列を設定します。
メッセージ待ち合わせ部品4 (Monitoring4)	filter.message_id	DIOOL2	コマンド実行時に出力され、SystemManager　Gで監視されるメッセージのIDを指定します。
	filter.node_name	HOST2	2つ目のACOSのホスト名を設定します。
	filter.message_text	LOGICAL SYSTEM IS EXIST	ACOSコマンドの2つ目の実行結果メッセージに含まれる文字列を設定します。

8.2.2.2. 実行結果

ACOSジョブ実行部品の出力結果に、ACOSコマンド実行時の照会画面を、メッセージ待ち合わせ部品の出力結果に、ACOSコマンド実行結果(監視設定したメッセージの内容)を表示します。

ACOSジョブ実行部品の出力結果の詳細については「8.5　ACOSジョブ実行部品」の出力結果を、メッセージ待ち合わせ部品の出力結果の詳細については「8.4　メッセージ待ち合わせ部品」の出力結果を参照してください。

以下に実行結果の例を記載します。設定値は8.2.2.1. カスタムパラメータ設定例と同一です。

■パターン1：1つの宛先に1つのメッセージが出力される場合

- ACOSコマンド実行結果
  - コマンド

    RJ SYSMGRG_MSGSEND_TEST01 IL=DIOS.OPM.MYSL
  - 照会画面の内容

    下記、ACOSジョブ実行部品の[screen]に出力(この例は、照会画面なしのコマンドのパターンのため出力なし)
  - コマンド実行結果

    下記、メッセージ待ち合わせ部品のmessage_textに出力
- ACOSジョブ実行部品

```
SystemManager Address: sysmgrg.local
Target[1]: REGION1@LCMP1:HOST1
Command: RJ SYSMGRG_MSGSEND_TEST01 IL=DIOS.OPM.MYSL

Command request success.
```

```
=== Target results ======================

[target]
REGION1@LCMP1:HOST1

[result]
Success

[screen]
```

- メッセージ待ち合わせ部品

```
～省略～
node_name : HOST1
message_text : 060116 TEST01 000545.001 TEST MESSAGE 1
～省略～
message_id : TEST01
～省略～
```

■パターン2：2つの宛先に2つのメッセージが出力される場合

- ACOSコマンド実行結果
  - コマンド

    DIOS OP DM LSNUM

  - 照会画面の内容

    下記、ACOSジョブ実行部品の各targetの[screen]に出力

  - コマンド実行結果

    下記、メッセージ待ち合わせ部品1～4のmessage_textに出力

- ACOSジョブ実行部品

```
SystemManager Address: sysmgrg.local
Target[1]: REGION1@LCMP1:HOST1
Target[2]: REGION1@LCMP1:HOST2
Command: DIOS OP DM LSNUM

Command request success.

=== Target results ======================

[target]
REGION1@LCMP1:HOST1

[result]
Success

[screen]
   060116 DIOO12     I 000424.001 COMMAND ACCEPTED C=DM
   060116 DIOOL1     I 000424.001 LOGICAL SYSTEM IS NOT EXIST

----------------------------------------
[target]
```

```
REGION1@LCMP1:HOST2

[result]
Success

[screen]
   060116 DIOO12     I 000424.001 COMMAND ACCEPTED C=DM
   060116 DIOOL2     I 000424.001 LOGICAL SYSTEM IS  EXIST , ACTIVE=01 STANDBY=00
```

- メッセージ待ち合わせ部品1

```
～省略～
node_name : HOST1
message_text : 060116 DIOO12     I 000424.001 COMMAND ACCEPTED C=DM
～省略～
message_id : DIOO12
～省略～
```

- メッセージ待ち合わせ部品2

```
～省略～
node_name : HOST1
message_text : 060116 DIOOL1     I 000424.001 LOGICAL SYSTEM IS NOT EXIST
～省略～
message_id : DIOOL1
～省略～
```

- メッセージ待ち合わせ部品3

```
～省略～
node_name : HOST2
message_text : 060116 DIOO12     I 000424.001 COMMAND ACCEPTED C=DM
～省略～
message_id : DIOO12
～省略～
```

- メッセージ待ち合わせ部品4

```
～省略～
node_name : HOST2
message_text : 060116 DIOOL2     I 000424.001 LOGICAL SYSTEM IS  EXIST , ACTIVE=01 STANDBY=00
～省略～
message_id : DIOOL2
～省略～
```

8.3. メッセージ監視登録部品

8.3.1. 部品情報

■格納パス
/Root/System/WebSAM/SystemManagerG

■定義名
register

■表示アイコン

8.3.2. 部品説明

SystemManager Gに対して、メッセージ監視フィルタの登録をリクエストします。

登録に成功すると、SystemMangerGはメッセージの監視を開始します。

メッセージ監視の条件は、メッセージ待ち合わせ部品のカスタムパラメータに設定します。

SystemMangerGで、条件に一致するメッセージが見つかると、JobCenterに通知され、条件に一致するメッセージ待ち合わせ部品が終了します。

本部品の実行より前のメッセージは検出できません。「8.1　メッセージ監視機能概要」の図のように、本部品を、メッセージ待ち合わせ部品よりも先に実行してください。

8.3.3. スクリプト

本部品にはスクリプトはありません。

8.3.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	message_monitoring_group	文字列	メッセージ監視グループ名。 値は任意の文字列です。 メッセージ待ち合わせ部品にも設定することで、同じ値の部品をグルーピングすることができます。 同じグループの、すべてのメッセージ待ち合わせ部品の監視フィルタを、本部品実行時にSystemManager Gに登録します。 メッセージ監視のタイミングを分けたい場合や、複数のSystemManager G対してメッセージを監視したい場合、グループを分けます。	group1
	sysmgr_port	数値	SystemManager GサーバのAPI受信ポート。 メッセージストアのhttpsポートを指定します。 パラメータ省略時は、デフォルト値として42524が設定されます。	42524

必須	変数	データ型	説明	設定例
○	sysmgr_server	文字列	SystemManager Gサーバのアドレス。	sysmgrg.local, 127.0.0.1
	tls_version	リスト	HTTPS接続時のTLSバージョン。 指定可能な値は 1.3 / 1.2 です。 パラメータ省略時は　1.3　での通信を行います。	1.3

8.3.5. 実行結果

■終了値

終了値	説明
0	SystemManager Gへの登録が成功した場合
1	エラー（メッセージ待ち合わせ部品の監視フィルタが未設定）の場合
2	エラー（SystemManager Gへの登録に失敗）の場合

■出力結果

登録成功時、SystemManager Gのアドレスと、登録した監視フィルタの数を出力します。

成功時の出力例は以下の通りです。

```
SystemManager Address: sysmgrg.local
Number of Registration filters: 2
Registration success.
```

■エラー出力結果

エラー時、エラーメッセージを出力します。

メッセージの内容は、後述の主要メッセージを参照してください。

8.3.6. 主要メッセージ

メッセージ	内容
Request parameter is empty.	メッセージ待ち合わせ部品の監視フィルタが未設定の場合に出力されます。
Registration failure. (XXX)	SystemManager　Gへのリクエストに失敗した場合に出力されます。

8.3.7. 注意事項

特になし

8.4. メッセージ待ち合わせ部品

8.4.1. 部品情報

■格納パス
/Root/System/WebSAM/SystemManagerG

■定義名
monitoring

■表示アイコン

8.4.2. 部品説明

条件に一致するメッセージが、SystemManager Gから通知されるのを待ち合わせます。

通知の対象となるメッセージは、メッセージ監視登録部品実行後に、SystemManager Gで検出されたメッセージです。

「8.1　メッセージ監視機能概要」の図のように、本部品は、メッセージ監視登録部品よりも後に実行してください。

メッセージ待ち合わせ部品は投入キューの指定は不要です。（特殊な部品であり、Managerで実行される部品です）

8.4.3. スクリプト

本部品にはスクリプトはありません。

8.4.4. カスタムパラメータ設定

8.4.4.1. カスタムパラメーター一覧

必須	変数	データ型	説明	設定例
	filter. agent_group_id	文字列	監視フィルタ：エージェントグループID。 指定したグループのエージェントのメッセージを監視します。 filter.agent_group_idを指定する場合、filter.agent_group_tenant_idを必ず指定する必要があります。 正規表現では指定できません。 指定した条件に一致しないメッセージを監視する場合、NOTを付与します。（例「NOT/文字列」） NOTについては、「8.4.4.5　NOTについて」を参照してください。 filter.agent_group_idで監視できるメッセージは、node_idが指定されていて、node_typeがAGENTのメッセージのみです。	msc_extlink_ 0000000002

必須	変数	データ型	説明	設定例
	filter.agent_group_recursive	リスト	監視フィルタ：通知対象のエージェントグループの範囲。 true： 指定したエージェントグループと配下のグループが対象です。 false： 指定したエージェントグループのみが対象です。 デフォルト値はfalseです。	true false
	filter.agent_group_tenant_id	文字列	監視フィルタ：filter.agent_group_idに指定したエージェントグループが所属するテナントのID。 filter.agent_group_idをフィルタパラメータに指定する場合、filter.agent_group_tenant_idを必ず指定する必要があります。	DefaultTenant
	filter.application	文字列	監視フィルタ：アプリケーション。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5　NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	Unified Management Framework
	filter.category	文字列	監視フィルタ：カテゴリ。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5　NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	SYSTEM ^APPLICATION$
	filter.definition_code	文字列	監視フィルタ：メッセージ定義ID。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」）	2000000000

必須	変数	データ型	説明	設定例
			NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	
	filter. eventcategory	文字列	監視フィルタ：イベントカテゴリ。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	^Unified Management Framework$
	filter.message_id	文字列	監視フィルタ：メッセージID。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	^14000.*
	filter. message_summary	文字列	監視フィルタ：メッセージ概要。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	Error
	filter.message_text	文字列	監視フィルタ：メッセージ本文。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」）	error_code=.*

必須	変数	データ型	説明	設定例
			NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	
	filter.name	文字列	監視フィルタ名。 この監視フィルタ(filter)の名前です。 128文字以内で設定してください。	filter1
	filter.node_id	文字列	監視フィルタ：ノードID。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	msc_extlink_MANAGER~AGENT
	filter.node_name	文字列	監視フィルタ：ノード名。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	nodeA
	filter.node_type	文字列	監視フィルタ：ノード種別。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5 NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	AGENT
	filter.object	文字列	監視フィルタ：オブジェクト。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」）	ProcessMonitor

必須	変数	データ型	説明	設定例
			指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5　NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	
	filter.selkey[0].compare_type	リスト	監視フィルタ：メッセージ本文のキー指定フィルタの比較条件。 メッセージ本文の、filter.selkey[0].keyで指定されるキーが持つ値 と、filter.selkey[0].valueの値を比較する際の条件です。 =： メッセージの指定キーの値 が、filter.selkey[0].valueの正規表現[注1]に一致することが条件です。 <>： メッセージの指定キーの値 が、filter.selkey[0].valueの値と一致しないことが条件です。 >： メッセージの指定キーの値 が、filter.selkey[0].valueの値より大きいことが条件です。 <： メッセージの指定キーの値 が、filter.selkey[0].valueの値より小さいことが条件です。 >=： メッセージの指定キーの値 が、filter.selkey[0].valueの値以上であることが条件です。 <=： メッセージの指定キーの値 が、filter.selkey[0].valueの値以下であることが条件です。 NOT_XXX： XXXの条件に一致しないことが条件です。 デフォルト値は=です。 selkeyについては、「8.4.4.4　selkeyについて」を参照してください。 NOTについては、「8.4.4.5　NOTについて」を参照してください。	= NOT_<=
	filter.selkey[0].key	文字列	監視フィルタ：メッセージ本文に対するキー指定フィルタのキー。 64文字以内で設定してください。	STATUS

必須	変数	データ型	説明	設定例
			selkeyについては、「8.4.4.4　selkeyについて」を参照してください。	
	filter.selkey[0].value	文字列	監視フィルタ：メッセージ本文に対するキー指定フィルタの比較値。 この値と、メッセージ本文の、キーで指定された値を比較します。 filter.selkey[0].compare_typeに=を指定した場合、正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」）。 1～64文字の範囲で設定してください。 selkeyについては、「8.4.4.4　selkeyについて」を参照してください。	SUCCESS
	filter.selkey[0].value_type	リスト	監視フィルタ：メッセージ本文に対するキー指定フィルタの比較値形式。 filter.selkey[0].valueの比較形式を指定します。 STRING： 文字列（バイナリ値）として比較します。 NUMBER： 数値として比較します。 デフォルト値はSTRINGです。 selkeyについては、「8.4.4.4　selkeyについて」を参照してください。	STRING
	filter.selkey[1]～[7]は、filter.selkey[0]と同様の内容です。 filter.selkey[0]～[7]で複数のselkeyを設定した場合、すべての条件を満たすメッセージが監視対象となります。			
	filter.selpos[0].compare_type	リスト	監視フィルタ：メッセージ本文の位置指定フィルタの比較条件。 メッセージ本文 の、filter.selpos[0].positionの位置の値 と、filter.selpos[0].valueの値を比較する際の条件です。 =： メッセージの指定位置の値 が、filter.selpos[0].valueの値と完全一致することが条件です。 <>： メッセージの指定位置の値 が、filter.selpos[0].valueの値と一致しないことが条件です。 >： メッセージの指定位置の値 が、filter.selpos[0].valueの値より大きいことが条件です。	= NOT_<

必須	変数	データ型	説明	設定例
			<： メッセージの指定位置の値 が、filter.selpos[0].valueの値より小さいことが条件です。 >=： メッセージの指定位置の値 が、filter.selpos[0].valueの値以上であることが条件です。 <=： メッセージの指定位置の値 が、filter.selpos[0].valueの値以下であることが条件です。 NOT_XXX： XXXの条件に一致しないことが条件です。 デフォルト値は=です。 selposについては、「8.4.4.3　selposについて」を参照してください。 NOTについては、「8.4.4.5　　NOTについて」を参照してください。	
	filter.selpos[0]. position	数値	監視フィルタ：メッセージ本文に対する位置指定フィルタの位置。 1～8192の整数を設定してください。 selposについては、「8.4.4.3　selposについて」を参照してください。	5
	filter.selpos[0]. value	文字列	監視フィルタ：メッセージ本文に対する位置指定フィルタの比較値。 この値と、メッセージ本文 の、filter.selpos[0].positionで指定された値を比較します。 正規表現は指定できません。 1～64文字の範囲で設定してください。 selposについては、「8.4.4.3　selposについて」を参照してください。	ERROR
	filter.selpos[0]. value_type	リスト	監視フィルタ：メッセージ本文に対する位置指定フィルタの比較値形式。 filter.selpos[0].valueの比較形式を指定します。 STRING： 文字列（バイナリ）として比較します。 NUMBER： 数値として比較します。 デフォルト値はSTRINGです。	STRING

必須	変数	データ型	説明	設定例
			selposについては、「8.4.4.3　selposについて」を参照してください。	
	filter.selpos[1]～[7]は、filter.selpos[0]と同様の内容です。 filter.selpos[0]～[7]で複数のselposを設定した場合、すべての条件を満たすメッセージが監視対象となります。			
	filter.severity	数値	監視フィルタ：重要度。 指定が無い場合は、フィルタ条件として無視されます。 0～255文字の整数を設定してください。	50
	filter. severity_range	リスト	監視フィルタ：重要度範囲。 指定が無い場合は、フィルタ条件として無視されます。 filter.severity_rangeを設定する場合、必ずfilter.severityを設定する必要があります。 >=： filter.severityの値以上の重要度のメッセージを監視します。 =： filter.severityの値と等しい重要度のメッセージを監視します。 <=： filter.severityの値以下の重要度のメッセージを監視します。 NOT_XXX： XXXの条件に一致しないメッセージを監視します。 デフォルト値は=です。 NOTについては、「8.4.4.5　NOTについて」を参照してください。	= NOT_>=
	filter.system_name	文字列	監視フィルタ：システム名。 正規表現[注1]で指定します。（完全一致の場合の例「^文字列$」） 指定した条件に一致しないメッセージを監視する場合、「NOT」を付与します。（例「NOT/^文字列$」） NOTについては、「8.4.4.5　NOTについて」を参照してください。 指定が無い場合は、フィルタ条件として無視されます。	System1
	filter.tags[0]. compare_type	リスト	監視フィルタ：メッセージのタグ情報の比較方法。	INCLUDES NOT_START_WITH

必須	変数	データ型	説明	設定例
			メッセージのタグ情報 と、filter.tags[0].nameの値を比較する際の条件です。 START_WITH： メッセージのタグ情報 が、filter.tags[0].nameの文字列で始まることが条件です。 END_WITH： メッセージのタグ情報 が、filter.tags[0].nameの文字列で終わることが条件です。 INCLUDES： メッセージのタグ情報 に、filter.tags[0].nameの文字列が含まれることが条件です。 EQUALS： メッセージのタグ情報 が、filter.tags[0].nameの文字列と同一であることが条件です。 REGEX： メッセージのタグ情報 が、filter.tags[0].nameで指定した正規表現[注1]に一致することが条件です。 NOT_XXX： XXXの条件に一致しないことが条件です。 デフォルト値はINCLUDESです。 NOTについては、「8.4.4.5　NOTについて」を参照してください。	
	filter.tags[0].name	文字列	監視フィルタ：メッセージに付与されたタグ名。 メッセージのタグ情報 と、filter.tags[0].compare_typeで指定した方法で比較します。	tag1
	filter.tags[1]～[2]は、filter.tags[0]と同様の内容です。			
	filter. tags_condition	リスト	監視フィルタ：メッセージのタグ情報の比較条件。 AND： filter.tags[0]～[2]の全ての条件を満たしたメッセージを通知します。 OR： filter.tags[0]～[2]のいずれかの条件を満たしたメッセージを通知します。 デフォルト値はORです。	AND
○	message_monitoring _group	文字列	メッセージ監視グループ名。 値は任意の文字列です。	group1

必須	変数	データ型	説明	設定例
			メッセージ監視登録部品にも設定することで、同じ値の部品をグルーピングすることができます。 本部品に設定した監視フィルタは、同じグループのメッセージ監視登録部品によってSystemManager Gに登録されます。 メッセージ監視のタイミングを分けたい場合や、複数のSystemManager G対してメッセージを監視したい場合、グループを分けます。	
	repexp[0].end_key	文字列	キー指定メッセージ継承の終了キー。 メッセージ情報のうち、設定した終了キーの直前までの文字列を後続ジョブに引き継ぎます。 正規表現で指定します。 半角スペースのみの場合、[ ]と入力してください。 省略した場合、メッセージ情報の末尾まで引き継ぎます。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引き継ぎ(変数継承)について」を参照してください。	END
	repexp[0].env	文字列	キー指定メッセージ継承の環境変数名。 後続ジョブに引き継ぐ際の環境変数名を設定します。 最大51byteの文字列を設定可能です。 省略した場合、引き継ぎません。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引き継ぎ(変数継承)について」を参照してください。	env_name
	repexp[0].refer	リスト	キー指定メッセージ継承の参照情報。 メッセージ情報のうち、後続ジョブに引き継ぐパラメータを設定します。 省略した場合、引き継ぎません。 デフォルト値はmessage_textです。 選択肢については、「8.4.4.1.1　メッセージのパラメーター覧」を参照してください。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引	message_text

必須	変数	データ型	説明	設定例
			き継ぎ(変数継承)について」を参照してください。	
	repexp[0].start_key	文字列	キー指定メッセージ継承の開始キー。 メッセージ情報の文字列のうち、設定した開始キーの後に続く文字列を後続ジョブに引き継ぎます。 正規表現で指定します。 半角スペースのみの場合、[ ]と入力してください。 省略した場合、メッセージ情報の先頭から引き継ぎます。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引き継ぎ(変数継承)について」を参照してください。	START
	repexp[1]～[7]は、repexp[0]と同様の内容です。 repexp[0]～[7]で同名の環境変数名を設定した場合、数字の大きいrepexpで抽出した値が継承されます。 値が空のrepexpは無視されます。			
	reppos[0].env	文字列	位置指定メッセージ継承の環境変数名。 後続ジョブに引き継ぐ際の環境変数名を設定します。 最大51byteの文字列を設定可能です。 省略した場合、引き継ぎません。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引き継ぎ(変数継承)について」を参照してください。	env_name
	reppos[0].range	数値	位置指定メッセージ継承の文字数。 メッセージ情報の文字列のうち、後続ジョブに引き継ぐ部分の文字数を設定します。 設定範囲は1～8192です。マルチバイト文字も1文字とカウントします。 省略した場合、メッセージ情報の末尾まで引き継ぎます。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引き継ぎ(変数継承)について」を参照してください。	10
	reppos[0].refer	リスト	位置指定メッセージ継承の参照情報。	message_text

必須	変数	データ型	説明	設定例
			メッセージ情報のうち、後続ジョブに引き継ぐパラメータを設定します。 省略した場合、引き継ぎません。 デフォルト値はmessage_textです。 選択肢については、「8.4.4.1.1　メッセージのパラメーター一覧」を参照してください。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引き継ぎ(変数継承)について」を参照してください。	
	reppos[0]. start_position	数値	位置指定メッセージ継承の開始位置。 メッセージ情報の文字列のうち、後続ジョブに引き継ぐ部分文字列の開始位置を設定します。 設定範囲は1～8192です。マルチバイト文字も1文字とカウントします。 省略した場合、メッセージ情報の先頭から引き継ぎます。 後続ジョブへの引き継ぎについては、「8.4.4.2　後続ジョブへのパラメータ引き継ぎ(変数継承)について」を参照してください。	5
	reppos[1]～[7]は、reppos[0]と同様の内容です。 reppos[0]～[7]で同名の環境変数名を設定した場合、数字の大きいreppposで抽出した値が継承されます。 値が空のrepposは無視されます。			

注1 入力値はそのままSystemManager Gに渡します。正規表現の細則はSystemManager Gに従います。

8.4.4.1.1. メッセージのパラメーター一覧

メッセージのパラメータは、SystemManager Gの定義に準拠します。

パラメータ名	説明
message_id	メッセージID
definition_code	メッセージ定義
tenant_id	テナントID
component_type	コンポーネントタイプ
severity	重要度
create_time	発生日時
system_name	システム名
node_id	ノードID
node_type	ノード種別
node_name	ノード名

パラメータ名	説明
application	アプリケーション
object	オブジェクト
category	カテゴリ
eventcategory	イベントカテゴリ
summary	メッセージ概要
receive_time	受信日時
message_no	メッセージ番号
message_text	メッセージ本文
original_message_no	送信元メッセージ番号

8.4.4.2. 後続ジョブへのパラメータ引き継ぎ(変数継承)について

単位ジョブでは、同一ルートジョブネットワーク内であれば、フロー上の任意の単位ジョブ、任意のサブジョブネットワーク内の単位ジョブの間で、変数継承機能により環境変数の値を引き継ぐことができます。

メッセージ待ち合わせ部品も、この機能によって、SystemManager Gから受信したメッセージの内容を、後続の単位ジョブに、環境変数として引き継ぐことができます。

Windows環境では、値が""(空文字)の環境変数は継承できません。

1. メッセージ待ち合わせ部品の［カスタムジョブパラメータ設定］ダイアログの［結果］タブで、変数継承を［STDOUT］に設定します。(デフォルトで［STDOUT］に設定されています。)

2. ［カスタムパラメータ設定］タブを開き、キー指定の場合は［repexp］、位置指定の場合は［reppos］で始まるカスタムパラメータを設定します。

   repexpについては「8.4.4.2.1 repexpについて」を、repposについては「8.4.4.2.2 repposについて」を参照してください。

3. ジョブネットワークを投入し、メッセージ待ち合わせ部品が終了した際、出力結果の最後に、以下のフォーマットで環境変数と値が出力されます。

   ここで出力された環境変数が、後続の単位ジョブに引き継がれます。

```
EXPORTVAR
(環境変数名)=(メッセージの内容)
(環境変数名)=(メッセージの内容)
・・・
EXPORTVAR
```

■EXPORTVAR内の環境変数は、以下のカスタムパラメータの順番で上から出力されます。

repexp[0]、repexp[1]、…、repexp[7]、reppos[0]、reppos[1]、…、reppos[7]

■EXPORTVARの出力の中に、同じ環境変数名が複数ある場合、最も下の値が後続の単位ジョブに引き継がれます。

8.4.4.2.1. repexpについて

repexpとは、SystemManager Gから通知されたメッセージの文字列から、指定した2つのキー(開始キーと終了キー)で囲まれた文字列を抽出するためのパラメータです。

開始キーと終了キーは、正規表現で指定します。

例：通知されたメッセージの本文から、ログIDだけを後続の単位ジョブに引き継ぐ

メッセージ待ち合わせ部品の設定

カスタムパラメータ	値	説明
repexp[0].refer	message_text	メッセージ本文を引き継ぐため、message_textを設定。
repexp[0].env	ENV_MESSAGE_TEXT	メッセージ本文を引き継ぐ環境変数名として、ENV_MESSAGE_TEXTを設定。
repexp[0].start_key	LOGID=	メッセージ本文の「LOGID=」以降を引き継ぐため、LOGID=を設定。
repexp[0].end_key	[ ]	メッセージ本文の区切り文字である[ ](半角スペース)を設定。

SystemManager Gから通知されたメッセージ

通知されたメッセージの項目	値
message_text(メッセージ本文)	MASSAGE LOGID=123 DATE=2022/01/01.

このとき、メッセージ待ち合わせ部品の出力結果の最後に以下の内容が出力され、環境変数「ENV_MESSAGE_TEXT」と、その値「123」が、後続の単位ジョブに引き継がれます。

```
EXPORTVAR
ENV_MESSAGE_TEXT=123
EXPORTVAR
```

8.4.4.2.2. repposについて

repposとは、SystemManager Gから通知されたメッセージの文字列の指定した位置(開始位置)から、指定した文字数(抽出文字数)の部分文字列を抽出して後続ジョブに環境変数として引き継ぐためのパラメータです。

開始位置の値は1文字目を1とします。

例：通知されたメッセージのシステム名から、連番部分だけを後続の単位ジョブに引き継ぐ

メッセージ待ち合わせ部品の設定

カスタムパラメータ	値	説明
reppos[0].refer	system_name	メッセージのシステム名を引き継ぐため、system_nameを設定。
reppos[0].env	ENV_SYSTEM_NAME	メッセージのシステム名を引き継ぐ環境変数名として、ENV_MESSAGE_TEXTを設定。
reppos[0].start_position	7	システム名の「SYSTEM」以降を引き継ぐため、7を設定。
repexp[0].range	3	システム名の連番部分を引き継ぐため、3を設定。

SystemManager Gから通知されたメッセージ

通知されたメッセージの項目	値
system_name(システム名)	System001.

このとき、メッセージ待ち合わせ部品の出力結果の最後に以下の内容が出力され、環境変数「ENV_SYSTEM_NAME」と、その値「001」が、後続の単位ジョブに引き継がれます。

```
EXPORTVAR
ENV_SYSTEM_NAME=001
```

```
EXPORTVAR
```

8.4.4.3. selposについて

selposとは、メッセージ本文上の特定の位置の値を監視フィルタにしたものです。

メッセージ本文の固定文字数の位置にある文字列を監視し、条件に一致する値であればSystemManager Gから通知されます。

例：特定の位置に[ERROR]と書かれたメッセージを監視したい

```
2021/01/01 12:00:00 [ERROR] file not found.
```

上記のように、メッセージ本文の21文字目(日時の後)の文字列が、[ERROR]であるメッセージを監視したい場合、以下のように、selposを使って監視フィルタを設定することができます。

パラメータ	値	説明
filter.selpos[0].position	21	メッセージ本文の21文字目の値。
filter.selpos[0].compare_type	=	値が一致すること。
filter.selpos[0].value	[ERROR]	メッセージ本文の値と文字列[ERROR]と比較。
filter.selpos[0].value_type	STRING	文字列として比較。

8.4.4.4. selkeyについて

selkeyとは、メッセージ本文上の特定のキーの値を監視フィルタにしたものです。

メッセージ本文の特定のキーで指定される文字列を監視し、条件に一致する値であればSystemManager Gから通知されます。

キーと値は、「キー:値」または「キー=値」の形式である必要があります。

例：キーCPUの値が50以上のメッセージを監視したい

```
Result paramer(HOST=jobmanager CPU=52 MEMORY=1234567890)
```

上記のように、CPUの値が50以上となっているメッセージを監視したい場合、以下のように、selkeyを使って、メッセージ上のキーを指定した監視フィルタを設定することができます。

パラメータ	値	説明
filter.selkey[0].key	CPU	キーCPUを監視。
filter.selkey[0].compare_type	>=	比較値以上であること。
filter.selkey[0].value	50	メッセージ本文の値と、50と比較。
filter.selpos[0].value_type	NUMBER	数値として比較。

8.4.4.5. NOTについて

一部の監視フィルタでは、例えば「システム名がsystemA以外のメッセージ」というように、指定した条件に一致しないメッセージを監視することができます。

カスタムパラメーター覧のうち、説明にNOTについて記載があるものが対象です。

NOTに対応する監視フィルタには文字列とリスト2種類のデータ型があります。

データ型	説明	監視フィルタ例
文字列	任意の文字列を設定する監視フィルタ。 書式	filter.message_text filter.system_name

データ型	説明	監視フィルタ例

	`<value>`
	<value>に一致するメッセージを監視
	`NOT/<value>`
	<value>に一致しないメッセージを監視
	`/<value>`
	<value>に一致するメッセージを監視
	(「NOT/」や「/」で始まる値に一致するメッセージを監視する場合などに使用)
	
	例：filter.system_name
	
	filter.system_name：NOT/systemA
	・・・システム名に「systemA」を含まないメッセージを監視
	filter.system_name：/NOT/systemA
	・・・システム名に「NOT/systemA」を含むメッセージを監視
	filter.system_name：//systemA
	・・・システム名に「/systemA」を含むメッセージを監視
		など

リスト	選択肢から選んで値を設定する監視フィルタ。
	
	書式
	
	`<value>`
	<value>に一致するメッセージを監視
	`NOT_<value>`
	<value>に一致しないメッセージを監視
	
	例：filter.tags[0].compare_type
	
	filter.tags[0].name：tag1
	filter.tags[0].compare_type：NOT_START_WITH
	・・・「tag1」で始まらないタグのメッセージを監視
		filter.selkey[0].compare_type
		
		filter.severity_range
		
		など

NOTに対応した文字列型の監視フィルタでは、値の先頭が「/」の場合、「/<value>」の書式に従い、「/」はないものとして動作します。「/」で始まる文字列を監視したい場合、先頭を「//」としてください。（例：filter.message_textに「//abc」を設定した場合、本文に「/abc」を含むメッセージを監視）

8.4.5. 実行結果

■終了値

終了値	説明
0	メッセージが通知され、待ち合わせが正常終了した場合
1	エラー（通知されたメッセージが不正）の場合

■出力結果

正常終了時、SystemManager Gから通知されたメッセージの内容を出力します。

また、カスタムパラメータrepexp[n].referおよびrepexp[n].env(またはreppos[n].referおよびreppos[n].env)を設定した場合、メッセージ内容の後に変数継承の内容も出力します。

成功時の出力例は以下の通りです。

この例では、カスタムパラメータrepexp[0].referにmessage_text、repexp[0].envに
ENV_MESSAGE_TEXTを設定しています。

```
comment :
tenant_id : DefaultTenant
node_name : node1
message_text : NEC System Manager G Agent is normal.
component_type : component1
category : APPLICATION
eventcategory : Unified Management Framework
node_id : msc_extlink_MANAGER~AGENT
create_time : 2021-02-04 01:23:45
confirm : UNCHECKED
severity : 50
message_id : 12345
system_name : System1
node_type : AGENT
application : Unified Management Framework
receive_time : 2021-02-04 01:23:45
summary : message - normal -
message_no : 10
mark :
original_message_no : 1
definition_code : 1400000
object : ProcessMonitor

EXPORTVAR
ENV_MESSAGE_TEXT=NEC System Manager G Agent is normal.
EXPORTVAR
```

■エラー出力結果

エラー時、エラーメッセージを出力します。

メッセージの内容は、後述の主要メッセージを参照してください。

8.4.6. 主要メッセージ

メッセージ	内容
Received invalid messages.(XXX)	SystemManager　Gから不正なメッセージが通知された場合に出力されます。

8.4.7. 注意事項

■特になし

8.5. ACOSジョブ実行部品

8.5.1. 部品情報

■格納パス
/Root/System/WebSAM/SystemManagerG

■定義名
acosjob

■表示アイコン

8.5.2. 部品説明

NEC ACOS(以下、ACOSと記します)でコマンドを実行します。

SystemManager GのACOS監視機能を利用するため、JobCenterから、SystemManager Gを通してACOSにコマンド実行をリクエストします。

ACOSジョブ実行部品を使ってコマンドを実行すると、以下の2種類の結果が得られます。

■ACOSの照会画面の内容

ACOSのコマンド実行時の照会画面の文字列です。

ACOSジョブ実行部品の出力結果に表示されます。

■実行結果

コマンドの実行結果の文字列です。

SystemManager Gのメッセージとして通知されます。

SystemManager GのACOS監視機能の監視設定によっては、通知されない場合もあります。

取得するには、メッセージ監視登録部品とメッセージ待ち合わせ部品が必要となります。

上記の実行結果を取得する場合、「8.2 ACOS監視機能概要」の図のように、本部品を、メッセージ監視登録部品よりも後かつメッセージ待ち合わせ部品よりも先に実行してください。

8.5.3. スクリプト

本部品にはスクリプトはありません。

8.5.4. カスタムパラメータ設定

必須	変数	データ型	説明	設定例
○	command	文字列	ACOSで実行するコマンドを指定します。	DIOS OP DM LSNUM
	command_timeout	数値	1つのACOSにコマンドを投入してから完了を待機する時間(秒)を指定します。 5から3600までの整数で入力してください。 パラメータ省略時は、デフォルト値として60秒が設定されます。	120
	sysmgr_port	数値	SystemManager GサーバのAPI受信ポート。	42539

必須	変数	データ型	説明	設定例
			ACOS Monitorの受信ポートを指定します。 パラメータ省略時は、デフォルト値として42539が設定されます。	
○	sysmgr_server	文字列	SystemManager Gサーバのアドレス。	sysmgrg.local, 127.0.0.1
○	target	文字列	コマンドを実行するACOSの物理ホスト名/論理システム名を指定します。 複数設定する場合、[,]区切りで指定します。 フォーマットについては、「8.5.4.1　targetのフォーマット」を参照してください。	HOST1,@LCMP2 LS2
	tls_version	リスト	HTTPS接続時のTLSバージョン。 指定可能な値は 1.3 / 1.2 です。 パラメータ省略時は　1.3　での通信を行います。	1.3

8.5.4.1. targetのフォーマット

宛先種別	フォーマット	例
ホスト	ホスト名	HOST1
	:ホスト名	:HOST1
	@LCMP:ホスト名	@LCMP1:HOST1
	CENTER@LCMP:ホスト名	TOKYO@LCMP1:HOST1
論理システム	@LCMP 論理システム名	@LCMP2 LS2
	@LCMP 論理システム名 投入先タイプ	@LCMP2 LS2 A
	CENTER@LCMP 論理システム名	TOKYO@LCMP2 LS2
	CENTER@LCMP 論理システム名 投入先タイプ	TOKYO@LCMP2 LS2 A
APM	@LCMP 論理システム名.APM名	@LCMP3 LS3.APM3
	@LCMP 論理システム名.APM名 投入先タイプ	@LCMP3 LS3.APM3 S
	CENTER@LCMP 論理システム名.APM名	TOKYO@LCMP3 LS3.APM3
	CENTER@LCMP 論理システム名.APM名 投入先タイプ	TOKYO@LCMP3 LS3.APM3 A
FIPS	@LCMP 論理システム名.FIPS名	@LCMP4 LS4.FIP4
	CENTER@LCMP 論理システム名.FIPS名	TOKYO@LCMP4 LS4.FIP4

■CENTER

CENTER名(リージョン名)を設定します。

省略可能です。

■LCMP

LCMP名(クラスタ名)を設定します。

宛先種別がホストの場合、省略可能です。

■投入先タイプ

A(稼動系)、S(待機系)のいずれか1文字を設定します。

省略可能です。省略時の値はAが設定されます。

FIPSは、稼働系と待機系の概念がない(厳密には稼働系のみである)ため、投入先タイプは設定しません。

8.5.5. 実行結果

■終了値

終了値	説明
0	SystemManager GでのACOSコマンド投入が成功した場合
1～240	SystemManager GでのACOSコマンド投入が失敗した場合 終了値は、SystemManager Gでのコマンド投入の結果コード
241	ACOSジョブ実行部品のカスタムパラメータが不足している場合
242	SystemManager GへのAPIリクエストが失敗した場合
	SystemManager GでのACOSコマンド投入が失敗した場合(SystemManager Gの結果コードが不正な値の場合)

■出力結果

以下の情報を出力します。

項目	説明	表示例
SystemManager Address	リクエスト先 SystemManager G エンドポイント(サーバ名またはIPアドレス)。 カスタムパラメータのsysmgr_serverの値を表示します。	sysmgrg.local
Target[n]	宛先のACOS。 カスタムパラメータのtargetの値を表示します。 「Target[n]: <宛先>」の形式で、カンマ区切りの先頭から連番で表示します。	acos1 acos2
Command	ACOSで実行するコマンド。 カスタムパラメータのcommandの値を表示します。	DIOS OP DM LSNUM
リクエスト結果	ACOSへコマンドをリクエストした結果。	▪ 成功時 Command request success. ▪ 失敗時 出力しません。エラー出力結果にエラーメッセージを出力します。
以下は宛先のACOSが複数ある場合、宛先のACOS毎にTarget[n]の順番で出力します。		

項目	説明	表示例
リクエストが失敗した場合、以下は出力結果には出力されず、エラー出力結果にエラーメッセージが出力されます。		
[target]	宛先のACOS。	acos1
[result]	コマンドの投入結果。 失敗時の結果コードとエラーメッセージについては、SystemManager　Gの窓口に問い合わせてください。	■ 成功時 Success ■ 失敗時 Failure (<結果コード> : <エラーメッセージ>)
[screen]	ACOSコマンドの参照結果。 ACOSの照会画面の内容を表示します。	00:34 DIOA20 I 000012.001 COMMAND ACCEPTED C=DI

リクエスト成功時の出力例は以下の通りです。

この例では、カスタムパラメータtargetに「acos1,acos2」、commandに「DIOS OP DM LSNUM」を設定しています。

```
SystemManager Address: sysmgrg.local
Target[1]: acos1
Target[2]: acos2
Command: DIOS OP DM LSNUM

Command request success.

=== Target results ======================

[target]
acos1

[result]
Success

[screen]
00:34  DIOA20     I 000012.001 COMMAND ACCEPTED C=DI
DIOS :ANY OP DM LSNUM
   060116 DIOO12     I 000424.001 COMMAND ACCEPTED C=DM
   060116 DIOOL1     I 000424.001 LOGICAL SYSTEM IS NOT EXIST

----------------------------------------
[target]
acos2

[result]
Failure (100 : command timeout.)

[screen]
```

■エラー出力結果

エラー時、エラーメッセージを出力します。

メッセージの内容は、後述の主要メッセージを参照してください。

8.5.6. 主要メッセージ

メッセージ	内容
Required parameter is empty.(<カスタムパラメータ名>)	ACOSジョブ実行部品のカスタムパラメータが不足している場合に出力されます。 必須となっているカスタムパラメータの設定を見直してください。 終了値は241です。
Failed to request to SystemManager G.(<エラー内容>)	SystemManager　Gへのリクエストに失敗した場合に出力されます。 終了値は242です。 <エラー内容>のフォーマットは以下のとおりです。 `http : <HTTPのエラー情報>, error : <SystemManager Gのエラーメッセージ>[<SystemManager Gのエラーコード>]`
Failed to execute ACOS command.	SystemManager　GでのACOSコマンド投入に失敗した場合に出力されます。 終了値は1～240または242です。

8.5.7. 注意事項

特になし

8.6. SystemManager G連携部品の注意事項

■MG/SVが、HP-UX、または、AIXである環境では、サポートしていません。

発行年月 May 2024
NEC Corporation　2024