スタンダードモード用ジョブ実行エージェント構築ガイド

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、JobCenter のジョブ実行を行う JobCenter AG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

→ <セットアップガイド>を参照してください。

■JobCenterを初めて利用される場合

→ <セットアップガイド>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用基本操作ガイド>を参照してください。

■JobCenterの動作を制御する設定やネットワーク関連の設定を理解したい場合

→　ジョブ管理マネージャ(MG)機能の設定については<スタンダードモード用環境構築ガイド>を参照してください。

→　ジョブ実行エージェント(AG)機能の設定については<スタンダードモード用ジョブ実行エージェント構築ガイド>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用コマンドリファレンス>を参照してください。

■JobCenterのイベントログ出力方法など監視に関連した機能を理解したい場合

→ <スタンダードモード用環境構築ガイド>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用移行ガイド>を参照してください。

■JobCenterのクラスタ環境を構築したい場合

→ <スタンダードモード用クラスタ機能利用の手引き>を参照してください。

■ブラウザを用いたJobCenterの監視やWebAPIでの制御などWebに関連した機能を理解したい場合

→ <スタンダードモード用Web機能利用の手引き>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. コマンドの表記方法

(例)

コマンド中の | は、「または」を意味します。

各オプションは、「-英文字 オプション名($xxx)」または「-数字(-$xxx)」となります。

引数は、「引数名($xxx)」となります。

[ ]付きのオプションは、省略可能です。

・・・ は直前の記述が繰り返し可能なことを意味します。

3. 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。

4. 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。

5. 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. コマンドの表記方法 vi
3. 凡例 vii
4. 関連マニュアル viii
5. 改版履歴 x
1. 概要 1
2. エージェントの環境構築 3
2.1. 言語設定 5
2.1.1. 初期設定 5
2.1.2. 言語の変更 6
2.2. インスタンス作成 7
2.3. インスタンス更新 10
2.4. インスタンス削除 13
2.5. サービス登録 14
2.6. サービスの起動と停止 15
2.6.1. サービスの起動 15
2.6.2. サービスの停止 15
2.6.3. サービスの状態確認 15
2.6.4. OSのシャットダウンに関する注意事項 16
2.7. OSユーザ登録 17
2.8. OSユーザ登録解除 19
3. エージェントの運用 20
3.1. エージェントの動作設定変更 21
3.1.1. エージェントの設定一覧 21
3.1.2. マネージャへの接続のリトライ間隔を変更する 27
3.1.3. ログサイズ、世代数を変更する 28
3.1.4. 再起動時の動作を変更する 28
3.1.5. リクエスト実行時の環境変数の設定を変更する 29
3.1.6. リクエストの実行ユーザの設定変更を即時に反映できるように変更する(Windows) 31
3.1.7. 巨大な出力結果の退避によるディスク使用量を制限する 31
3.1.8. 実行中リクエストの出力結果の更新頻度を変更する 32
3.1.9. ファイル待ち合わせの動作を変更する 32
3.2. エージェントの構成情報のバックアップ・復元 34
3.2.1. エージェントの構成情報のバックアップ 34
3.2.2. エージェントの構成情報の復元 34
4. OS要塞化 36
5. トラブルシューティング 37
5.1. JobCenter Agent の障害発生時、原因究明に必要な情報を採取する 38
5.1.1. jc_getinfoコマンドで採取する場合 38

表の一覧

2.1. エージェントを利用するために必要な作業 3
2.2. エージェントの設定内容の変更や削除に必要な作業一覧 4
2.3. エージェントに設定可能な言語一覧 6
2.4. jcagctrl createコマンドの引数 7
2.5. jcagctrl updateコマンドで変更可能なパラメータに対応する引数 10
3.1. エージェントの設定ファイルのパラメーター一覧 21
3.2. エージェントの構成情報のバックアップファイル 34

1. 概要

JobCenter AGはJobCenter MGからの指示を受けてバッチ処理を実行するアプリケーションです。JobCenter AGは単体では動作しませんので、必ずJobCenter MGとの連携設定が必要となります。

JobCenter AGとJobCenter MGの連携設定の概要は下図の通りとなります。

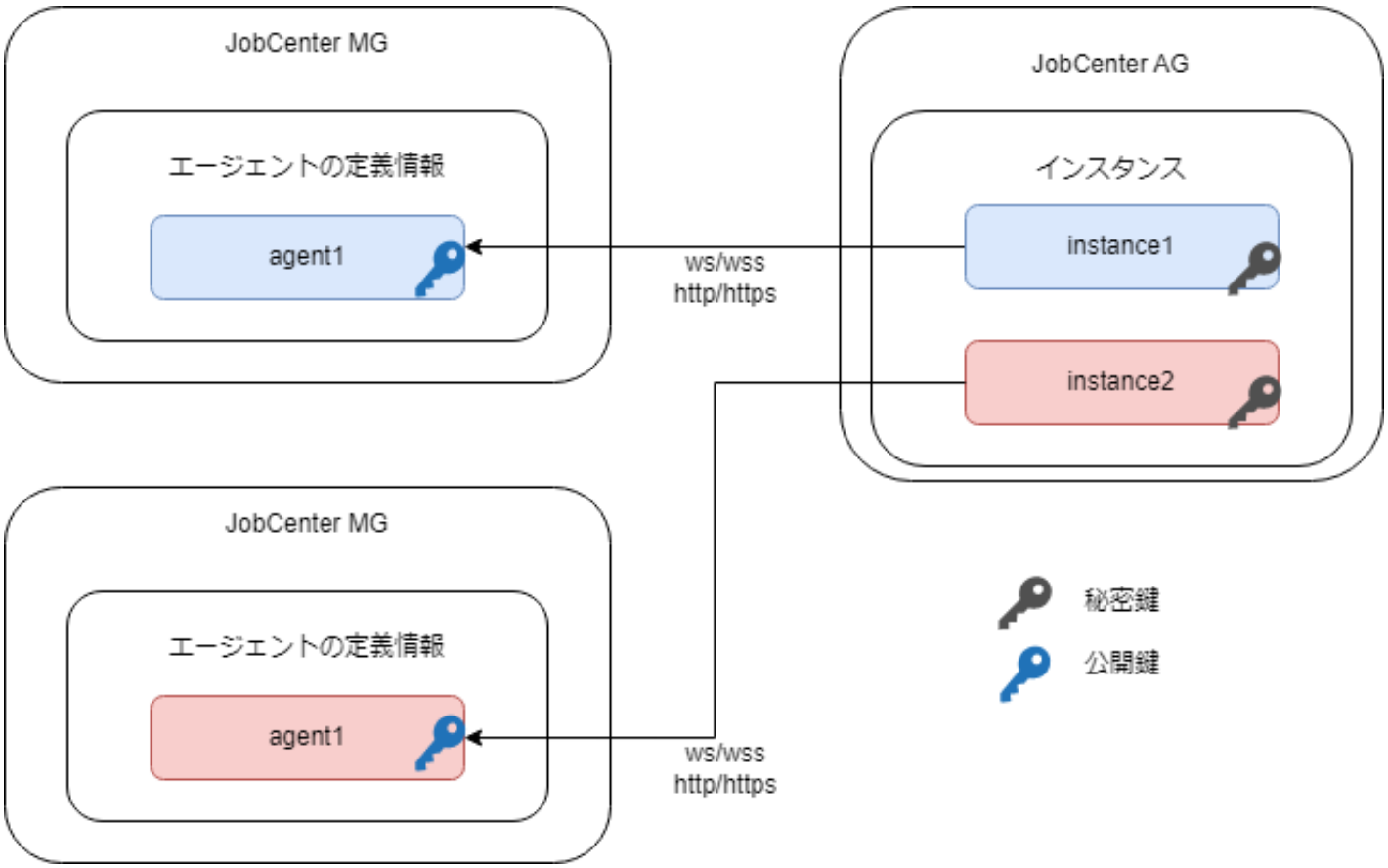

図1.1 JobCenter MGとJobCenter AGの連携の概要

JobCenter AGではインスタンスという実行単位毎にアプリケーション(プロセス)を起動し、JobCenter MGとの連携やバッチ処理の実行を行います。

インスタンス毎に管理する情報として接続先のJobCenter MGのエンドポイント、連携時の認証に使用する秘密鍵の格納場所、インスタンスが使用するspoolディレクトリなどがあります。

spoolディレクトリとは、エージェントのインスタンスのデータを格納するディレクトリとなります。

JobCenter AGの各インスタンスとJobCenter MGの通信はws/wss(WebSocket)およびhttp/https通信で行われます。

連携設定時に使用されるのはws/wss(WebScoket)通信であり、http/https通信については連携設定後に通信量が大きくなる可能性がある一部の通信(バッチ処理の結果返却など)で使用されます。

またコネクションの確立を行うのはJobCenter AGからJobCenter MGに対してのみとなります。このため、JobCenter AGからJobCenter MGに対して通信ができる環境であれば連携が可能です。

連携設定時、JobCenter MGではJobCenter AGからの接続が不正な接続ではないことを保証するために、秘密鍵と公開鍵を用いた認証を行います。

具体的にはJobCenter　MG上では公開鍵を保持しJobCenter　AG側で公開鍵に対応する秘密鍵を指定することで、接続してきたJobCenter AGが信用できるか(秘密鍵が公開鍵に対応しているか)を検証します。

JobCenterでは実行するバッチ処理をリクエストという単位で管理します。

ジョブネットワークの中の単位ジョブやカスタムジョブ、拡張カスタムジョブなどジョブの実行毎に1つのリクエストを作成し、リクエスト単位でキューイング、およびJobCenter　AG側での実行を行います。

ジョブの状態は対応するリクエストの状態にひもづきます。このため、内部エラーなどバッチ処理以外の部分でリクエストが異常終了した場合は対応するジョブも異常終了します。

JobCenter MG側に事前に公開鍵を設定する必要があるため、事前にJobCenter MG側でエージェントの定義情報の登録が必要となります。

JobCenter　MG側でのエージェントの定義情報の登録の詳細については<スタンダードモード用環境構築ガイド>の「4.1 エージェントの管理」を参照してください。

JobCenter AG上に配置する秘密鍵の格納場所については任意の場所で問題ありませんが、重要な認証情報となりますので一般ユーザではアクセスできない場所に配置いただき、秘密鍵自体も一般ユーザでアクセスできないよう適切に権限を設定いただくことを推奨いたします。

2. エージェントの環境構築

エージェントの環境構築手順について説明します。

エージェントの環境構築は、JobCenter　AGのjcagctrlコマンドを使用します。jcagctrlコマンドの詳細については、<スタンダードモード用コマンドリファレンス>の「4.1 jcagctrl エージェントの情報を管理」を参照してください。

■本マニュアルは、エージェント側で設定する環境構築手順について記載しています。エージェントを利用するにはマネージャ側の設定が必要な項目があります。詳細は表2.1「エージェントを利用するために必要な作業」の手順列に記載しているマニュアルを参照してください。

■jcagctrlコマンドは、UNIX版ではrootユーザ、Windows版ではAdministratorsグループのユーザで実行してください。

■コマンド中の %InstallDirectory% は、Windows版JobCenter AGのインストールディレクトリに読み替えてください。

エージェントを利用するために必要な作業は以下のとおりです。

表2.1 エージェントを利用するために必要な作業

対象	作業内容	手順
マネージャ	マネージャへのエージェント登録	<スタンダードモード用環境構築ガイド>の「4.1 エージェントの管理」
エージェント	言語設定(初期設定)	「2.1.1 初期設定」
エージェント	インスタンス作成	「2.2 インスタンス作成」
エージェント	サービス登録	「2.5 サービス登録」
エージェント	サービス起動	「2.6.1 サービスの起動」
エージェント	OSユーザ登録	「2.7 OSユーザ登録」
マネージャ	ユーザマッピング登録	<スタンダードモード用環境構築ガイド>の「4.2 ユーザマッピング」

以下の作業はエージェントのインスタンスの設定内容の更新や削除手順です。実施したい場合に参照してください。

表2.2 エージェントの設定内容の変更や削除に必要な作業一覧

対象	作業内容	手順
エージェント	言語設定(変更)	「2.1.2 言語の変更」
エージェント	インスタンス更新	「2.2 インスタンス作成」
エージェント	インスタンス削除	「2.4 インスタンス削除」
エージェント	OSユーザ登録解除	「2.8 OSユーザ登録解除」

2.1. 言語設定

エージェントの言語を設定する手順について説明します。

単位ジョブ等のバッチ処理を実行する際に使用する言語になります。エージェントの環境構築の最初に必ず設定する必要があります。

本設定はJobCenter AGをインストールしたマシン上で共通となります。インスタンスごとに異なる言語を設定することはできません。

2.1.1. 初期設定

エージェントの言語設定を初めて実施する場合の手順について説明します。

エージェントの言語設定はjcagctrl set-langコマンドで行います。以下のコマンドを実行してください。

■UNIX版

```
/usr/bin/jcagctrl set-lang --lang $lang
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl set-lang --lang $lang [--unicode-mode]
```

$langには、設定したい言語に対応した値を指定します。JobCenter AGをインストールしたOSによって選択可能な言語が異なります。

日本語環境のWindows場合、UNICODEモードと非UNICODEモードを選択できます。--unicode-modeオプションを指定した場合はUNICODEモード、指定しない場合は非UNICODEモードとなります。

■非UNICODEモードを指定する場合

```
%InstallDirectory%\bin\jcagctrl set-lang --lang sjis
```

■UNICODEモードを指定する場合

```
%InstallDirectory%\bin\jcagctrl set-lang --lang sjis --unicode-mode
```

■UNICODEモードについて

単位ジョブのバッチ処理をUTF-16で実行します。単位ジョブのスクリプト内でUTF-16で出力するプログラムを実行したい場合に選択してください。それ以外の場合は非UNICODEモードを選択することを推奨します。

■UNICODEモードは日本語環境のみサポートします。中国語環境または英語環境でUNICODEモード(--unicode-modeオプション指定)を設定した場合は動作保証しておりません。

■UNICODEモードの場合、単位ジョブの出力結果およびエラー出力結果が全てUTF-16となるようにしてください。UTF-16以外の文字が混在している場合、単位ジョブの実行結果が文字化けします。

指定できる言語および$langの値の対応表は以下のとおりです。

表2.3 エージェントに設定可能な言語一覧

OS	言語	$lang
UNIX	UTF-8	utf-8
	EUC-JP	euc
	Shift-JIS	sjis
	GB18030	gb18030
	English	english
Windows (日本語環境)	日本語 (非UNICODEモード)	sjis
	日本語 (UNICODEモード)	sjis (--unicode-mode指定)
	英語	english
Windows (中国語環境)	中国語	gb18030
	英語	english
Windows (英語環境)	英語	english

2.1.2. 言語の変更

エージェントの言語を変更する場合の手順について説明します。

変更手順は「2.1.1　初期設定」と同様です。ただし、jcagctrl　set-langコマンドを実行するには、全てのエージェントが停止している必要があります。事前に全てのエージェントを停止してから実施してください。エージェントの停止方法は「2.6 サービスの起動と停止」を参照してください。

jcagctrl set-langコマンドを実行すると、以下のメッセージが表示されます。

```
After change language settings, will need to reconfigure the agent.
Do you want to continue? [y/N]
```

「y」を入力すると、エージェントの言語が変更されます。「N」を入力すると、変更がキャンセルされます。

2.2. インスタンス作成

エージェントのインスタンスを作成する手順について説明します。

インスタンスの作成には、事前に以下の作業を実施する必要があります。

■エージェントの言語設定

■マネージャへのエージェント登録

エージェントの言語設定の手順は、「2.1 言語設定」を参照してください。

マネージャへのエージェント登録は、JobCenter CL/Winでマネージャに接続しマネージャフレームのエージェント登録画面から行います。手順は<スタンダードモード用環境構築ガイド>の「4.1 エージェントの管理」を参照してください。

1. 秘密鍵の格納

マネージャへのエージェント登録の際にダウンロードした秘密鍵を、JobCenter AGをインストールしたマシン上からアクセスできるディレクトリに格納します。

秘密鍵は一般ユーザからアクセスできないディレクトリに格納することを推奨します。

2. spoolディレクトリの格納先作成

spoolディレクトリの格納先(空ディレクトリ)を作成します。spoolディレクトリとは、エージェントのインスタンスのデータを格納するディレクトリとなります。

■UNIX版の場合、spoolディレクトリのパーミッションは755に設定してください。

■spoolディレクトリの上位ディレクトリは、一般ユーザで読み書き可能なパスを指定してください。

3. インスタンスの作成

以下のコマンドを実行してください。

■UNIX版

```
/usr/bin/jcagctrl create --instance-name $name --endpoint $url --agent-name $agent-name --
private-key $filepath --spool-dir $filepath
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl create --instance-name $name --endpoint $url --agent-name
 $agent-name --private-key $filepath --spool-dir $filepath
```

引数の説明は以下のとおりです。

表2.4 jcagctrl createコマンドの引数

引数	説明	設定例
--instance-name $name	エージェントのインスタンス名を指定します。	agent1

引数	説明	設定例
	JobCenter AGをインストールしたマシン上で一意の名前を指定してください。	
--endpoint $url	接続先マネージャのURLを指定します。 $urlの書式は以下のとおりです。 <httpまたはhttps>://<接続先マネージャのホスト名またはIPアドレス>:<JCEXECUTORのポート番号> プロトコルについて、接続先マネージャのjcexecutor_webserverのHTTPS設定を行っている場合はhttps、行っていない場合はhttpとなります。詳細は<スタンダードモード用環境構築ガイド>の「6.9 jcexecutor_webserverデーモンの動作設定について」を参照してください。 JCEXECUTORのポート番号のデフォルト値は23151です。ポート番号を変更している場合はそのポート番号を指定してください。	https://manager:23151
--agent-name $agent-name	エージェント名を指定します。 マネージャへのエージェント登録の際に入力したエージェント名を指定してください。	agent1
--private-key $filepath	秘密鍵のパスを指定します。 手順1で実施した秘密鍵の格納先のファイルパスを指定してください。	/usr/local/jcagent_private_key/rsa.key
--spool-dir $filepath	エージェントのspoolディレクトリのパスを指定します。 手順2で作成したspoolディレクトリのパスを指定してください。	/usr/spool/agent1

上記コマンドは必要最低限の引数のみ記載しています。その他の引数および引数の詳細については、<スタンダードモード用コマンドリファレンス>の「4.1 jcagctrl エージェントの情報を管理」のcreateサブコマンドを参照してください。

4. インスタンスの確認

手順3のjcagctrl　createコマンドが成功すると、エージェントにインスタンスが登録されます。登録されたインスタンスは以下のコマンドで確認可能です。

■UNIX版

```
/usr/bin/jcagctrl list
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl list
```

jcagctrl listコマンドの実行例は以下のとおりです。

```
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+------------+-----------------------+
| INSTANCE-NAME | INSTANCE-ID                          | ACTIVE | ENABLE | AGENT-ID                             | AGENT-NAME | USER-GROUP | ENDPOINT              |
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+------------+-----------------------+
| agent1        | b850b57d-f5ac-4a34-a757-f18e71f0e6c9 | false  | false  | a7a6d98a-9a50-4ec1-a085-0184aa71c7f9 | agent1     | jobcenterag | https://manager:23151 |
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+------------+-----------------------+
```

作成したエージェントのインスタンスを利用するには、サービス登録とサービスの起動が必要です。サービス登録は「2.5 サービス登録」、サービスの起動は「2.6.1 サービスの起動」を参照してください。

2.3. インスタンス更新

エージェントのインスタンスの設定を更新する手順について説明します。

1. インスタンスのサービス停止

   インスタンス更新は、インスタンスのサービスが停止している必要があります。

   インスタンスのサービス停止手順は「2.6.2 サービスの停止」を参照してください。

2. インスタンスの更新

   以下のコマンドを実行してください。

   ■UNIX版

   ```
   /usr/bin/jcagctrl update --instance-name $name <変更するパラメータ>
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl update --instance-name $name <変更するパラメータ>
   ```

   $nameには、更新対象のインスタンス名を指定します。

   <変更するパラメータ>には、変更するパラメータに対応する引数を複数指定します。変更しないパラメータについては省略可能です。

   --no-verifyオプションについて、指定時はルート証明書の検証を無効、省略時は有効となるため、無効のままにしたい場合は都度指定してください。

   変更可能なパラメータは以下のとおりです。パラメータの詳細については、<スタンダードモード用コマンドリファレンス>の「4.1　jcagctrl　エージェントの情報を管理」のupdateサブコマンドを参照してください。

表2.5 jcagctrl updateコマンドで変更可能なパラメータに対応する引数

引数	説明	設定例
--endpoint $url	接続先マネージャのURLを指定します。	https://manager:23151
--agent-id $agent-id	エージェントIDを指定します。	a7a6d98a-9a50-4ec1-a085-0184aa71c7f9
--agent-name $agent-name	エージェント名を指定します。	agent1
--private-key $filepath	秘密鍵のパスを指定します。	/usr/local/jcagent_private_key/rsa.key
--spool-dir $filepath	エージェントのspoolディレクトリのパスを指定します。	/usr/spool/agent1
--user-group $group	エージェントのOSユーザのユーザグループ名を指定します。 指定したユーザグループがOS上に存在しない場合は、自動でユーザグループを作成します。	jobcenterag
--root-ca $filepath	エージェントで使用するルート証明書のパスを指定します。	/usr/local/jcagent_root_ca/cacert.pem

引数	説明	設定例
--no-verify	エージェントで使用するルート証明書の検証を無効化したい場合に指定します。 有効化したい場合は、本オプションの指定をせずに実行してください。	-
--proxy $proxy	マネージャに接続する際に使用するプロキシのURLを指定します。	https://proxy.example.com

例えば、UNIX版でインスタンス名instance1の接続先マネージャのURLをhttps://manager-new:23151に変更する場合は、以下のように実行します。

```
/usr/bin/jcagctrl update --instance-name instance1 --endpoint https://manager-new:23151
```

■jcagctrl updateコマンドは、変更後のパラメータでマネージャへの接続確認を行います。接続できなかった場合はインスタンスの更新に失敗します。

■jcagctrl updateコマンドは、エージェントのインスタンスの情報を更新します。マネージャに登録されているエージェント情報は更新されない点にご注意ください。

例えば、インスタンス名instance1のエージェント名をagent1からagent2に変更する場合、マネージャに登録されたエージェントagent2のインスタンスがinstance1に変更されることになります。マネージャにagent2が登録されていない場合、エージェントからマネージャへの接続に失敗し、インスタンスの更新に失敗します。

3. インスタンスの設定の確認

インスタンスの設定が正しく更新されたか確認します。以下のコマンドで対象のパラメータが変更されていることを確認してください。

■UNIX版

```
/usr/bin/jcagctrl list --format=json --pretty-print
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl list --format=json --pretty-print
```

エージェントのインスタンスの全パラメータを確認するには、JSON形式(--format　json)で表示する必要があります。

jcagctrl listコマンドの実行例は以下のとおりです。

```
[
    {
        "instance-name": "agent1",
        "instance-id": "619bca8e-0a55-457c-8bd1-37444d53a00a",
        "agent-id": "a7a6d98a-9a50-4ec1-a085-0184aa71c7f9",
        "agent-name": "agent1",
        "endpoint": "http://manager:23151",
        "private-key": "/usr/local/jcagent_private_key/rsa.key",
        "spool-dir": "/usr/spool/agent1",
        "group": "jobcenterag",
        "root-ca": null,
        "no-verify": false,
```

```
        "proxy": null,
        "enable": true,
        "active": true
    }
]
```

4. インスタンスのサービス起動

   インスタンスの設定が正しく更新されたことを確認できたら、インスタンスのサービスを起動します。

   インスタンスのサービス起動手順は「2.6.1 サービスの起動」を参照してください。

2.4. インスタンス削除

エージェントのインスタンスを削除する手順について説明します。

1. インスタンスのサービス停止

   インスタンス削除は、インスタンスのサービスが停止している必要があります。

   インスタンスのサービス停止手順は「2.6.2 サービスの停止」を参照してください。

2. インスタンスのサービス解除

   以下のコマンドを実行してください。

   ■UNIX版

   ```
   /usr/bin/jcagctrl disable --instance-name $name
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl disable --instance-name $name
   ```

   $nameには、エージェントのインスタンス名を指定します。

3. インスタンスの削除

   以下のコマンドを実行してください。

   ■UNIX版

   ```
   /usr/bin/jcagctrl delete --instance-name $name
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl delete --instance-name $name
   ```

   $nameには、エージェントのインスタンス名を指定します。

4. インスタンスの削除確認

   インスタンスが削除されたことを確認します。以下のコマンドで確認可能です。

   ■UNIX版

   ```
   /usr/bin/jcagctrl list
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl list
   ```

   対象のインスタンスが表示されなければ、正常にインスタンスが削除されています。エージェントのspool
   ディレクトリは削除されませんので、不要であれば削除してください。

2.5. サービス登録

エージェントのインスタンスのサービスを登録する手順について説明します。

1. インスタンスのサービス登録

   以下のコマンドを実行してください。

   ■UNIX版

   ```
   /usr/bin/jcagctrl enable --instance-name $name
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl enable --instance-name $name
   ```

   $nameには、エージェントのインスタンス名を指定します。

2. インスタンスの確認

   手順1のjcagctrl　enableコマンドが成功すると、エージェントのインスタンスのサービスがOSに登録されます。以下のコマンドで確認可能です。

   ■UNIX版

   ```
   /usr/bin/jcagctrl list
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl list
   ```

   実行結果のENABLE列がtrueとなっていれば、正常にサービス登録されています。jcagctrl listコマンドの実行例は以下のとおりです。

```
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+-------------+-----------------------+
| INSTANCE-NAME | INSTANCE-ID                          | ACTIVE | ENABLE | AGENT-ID                             | AGENT-NAME | USER-GROUP  | ENDPOINT              |
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+-------------+-----------------------+
| agent1        | b850b57d-f5ac-4a34-a757-f18e71f0e6c9 | false  | true   | a7a6d98a-9a50-4ec1-a085-0184aa71c7f9 | agent1     | jobcenterag | https://manager:23151 |
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+-------------+-----------------------+
```

2.6. サービスの起動と停止

エージェントのサービスの起動と停止、およびサービスの状態を確認する手順について説明します。

サービスの起動と停止には、事前に以下の作業を実施する必要があります。

■エージェントのサービス登録

エージェントのサービス登録の手順は、「2.5 サービス登録」を参照してください。

2.6.1. サービスの起動

エージェントのサービス起動はjcagctrl startコマンドで行います。以下のコマンドを実行してください。

■UNIX版

```
/usr/bin/jcagctrl start --instance-name $name
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl start --instance-name $name
```

$nameには、エージェントのインスタンス名を指定します。

2.6.2. サービスの停止

エージェントのサービス停止はjcagctrl startコマンドで行います。以下のコマンドを実行してください。

■UNIX版

```
/usr/bin/jcagctrl stop --instance-name $name
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl stop --instance-name $name
```

$nameには、エージェントのインスタンス名を指定します。

2.6.3. サービスの状態確認

エージェントのサービスの状態はjcagctrl listコマンドで確認できます。以下のコマンドを実行してください。

■UNIX版

```
/usr/bin/jcagctrl list
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl list
```

実行結果のACTIVE列がtrueとなっていれば起動中、falseとなっていれば停止中となります。jcagctrl listコマンドの実行例は以下のとおりです。

```
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+------------+-----------------------+
| INSTANCE-NAME | INSTANCE-ID                          | ACTIVE | ENABLE | AGENT-ID                             | AGENT-NAME | USER-GROUP | ENDPOINT              |
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+------------+-----------------------+
| agent1        | b850b57d-f5ac-4a34-a757-f18e71f0e6c9 | true   | true   | a7a6d98a-9a50-4ec1-a085-0184aa71c7f9 | agent1     | jobcenterag | https://manager:23151 |
+---------------+--------------------------------------+--------+--------+--------------------------------------+------------+------------+-----------------------+
```

2.6.4. OSのシャットダウンに関する注意事項

Windowsをシャットダウンする際、JobCenter AGの終了処理が完了する前に、Windowsから強制的にJobCenter AGの各プロセスが終了されてしまうことがあります。その結果、JobCenter AGの正常動作を保証できなくなる可能性があります。

（例）エージェントに転送されているリクエスト情報が失われてしまう

これを回避するには、あらかじめJobCenter AGのサービスを停止してからWindowsをシャットダウンする必要があります。簡単で確実な方法は、Windowsのシャットダウンスクリプトで JobCenter AGのサービスの停止処理を行うことです。グループポリシーエディタ(gpedit.msc)を起動し、以下の操作でJobCenter AGのサービスの停止を行うスクリプトを追加してください。

コンピュータの構成→Windowsの設定→スクリプト

JobCenter AGのサービスの停止スクリプトは、通常のバッチファイルを作成し、サービス起動しているすべてのインスタンスに対して下記のコマンドを記述してください。

```
%InstallDirectory%\bin\jcagctrl stop --instance-name <インスタンス名>
```

%InstallDirectory% はJobCenter AGのインストールディレクトリのパスです。デフォルトのインストールディレクトリパスはC:\Program Files\JobCenter\Agentです。

2.7. OSユーザ登録

エージェントで使用するOSユーザの登録手順について説明します。

1. OSユーザ作成

   JobCenter AGをインストールしたマシン上に、エージェントで使用するOSユーザを作成します。

2. OSユーザ登録

   以下のコマンドを実行してください。

   ■UNIX版

   ```
   /usr/bin/jcagctrl add-user --instance-name $name --user-name $user
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl add-user --instance-name $name --user-name $user
   ```

   $nameには、エージェントのインスタンス名を指定します。

   $userには、登録するOSユーザ名を指定します。

   jcagctrl add-userコマンドに成功すると、OSユーザが所属するユーザグループに、エージェントのインスタンスに設定したユーザグループが登録されます。

   エージェントで使用するOSユーザは、インスタンスに設定したユーザグループに所属しているかで判定しています。そのため、複数のインスタンスでユーザグループの設定が重複している場合、jcagctrl add-userコマンドで指定したインスタンス以外にも、ユーザグループが同じインスタンス全てに登録されることになります。

   インスタンスごとに登録するOSユーザを区別したい場合は、「2.2　インスタンス作成」または「2.3　インスタンス更新」で--user-groupパラメータに異なるユーザグループを指定してください。

   Windows版の場合、ユーザグループにバッチ処理実行に必要な以下の権限が自動で付与されます。

権限	意味
SeInteractiveLogonRight	バッチジョブとしてログオン
SeBatchLogonRight	ローカルログオン

3. OSユーザの登録確認

   OSユーザが登録されたことを確認します。以下のコマンドで確認可能です。

   ■UNIX版

   ```
   /usr/bin/jcagctrl list-user --instance-name $name
   ```

   ■Windows版

   ```
   %InstallDirectory%\bin\jcagctrl list-user --instance-name $name
   ```

   $nameには、エージェントのインスタンス名を指定します。

UNIX版のjcagctrl list-userコマンドの実行例は以下のとおりです。

```
+----------+------+
| NAME     | UID  |
+----------+------+
| user1    | 1000 |
+----------+------+
```

Windows版のjcagctrl list-userコマンドの実行例は以下のとおりです。

```
+----------+------------+-------------------------------------------------+
| NAME     | DOMAIN     | SID                                             |
+----------+------------+-------------------------------------------------+
| user1    | domain1    | S-1-5-21-2644628043-2395542531-2746206096-1005  |
+----------+------------+-------------------------------------------------+
```

Windows版の場合、ドメインユーザはDOMAIN列にドメイン名が表示されます。ローカルユーザは空欄となります。

ここで登録したOSユーザが、単位ジョブ等のバッチ処理を実行するOSユーザの候補になります。本作業の実施後、マネージャ側でユーザマッピング設定を行う必要があります。ユーザマッピングの設定手順は<スタンダードモード用環境構築ガイド>の「4.2 ユーザマッピング」を参照してください。

2.8. OSユーザ登録解除

エージェントで使用するOSユーザの登録解除手順について説明します。

1. OSユーザ登録解除

以下のコマンドを実行してください。

■UNIX版

```
/usr/bin/jcagctrl del-user --instance-name $name --user-name $user
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl del-user --instance-name $name --user-name $user
```

$nameには、エージェントのインスタンス名を指定します。

$userには、登録解除するOSユーザ名を指定します。

jcagctrl　del-userコマンドに成功すると、エージェントのインスタンスに設定したユーザグループから指定したOSユーザが削除されます。

エージェントで使用するOSユーザは、インスタンスに設定したユーザグループに所属しているかで判定しています。そのため、複数のインスタンスでユーザグループの設定が重複している場合、jcagctrl　del-userコマンドで指定したインスタンス以外にも、ユーザグループが同じインスタンス全てが登録解除されることになります。

インスタンスごとに登録するOSユーザを区別したい場合は、「2.2　インスタンス作成」または「2.3　インスタンス更新」で--user-groupパラメータに異なるユーザグループを指定してください。

2. OSユーザの登録解除確認

OSユーザが登録解除されたことを確認します。以下のコマンドで確認可能です。

■UNIX版

```
/usr/bin/jcagctrl list-user --instance-name $name
```

■Windows版

```
%InstallDirectory%\bin\jcagctrl list-user --instance-name $name
```

$nameには、エージェントのインスタンス名を指定します。

対象のOSユーザが表示されなければ、正常にOSユーザの登録が解除されています。

3. エージェントの運用

本章では、エージェントの動作設定の変更やバックアップ・復元などのエージェントに関する運用について説明します。

3.1. エージェントの動作設定変更

エージェントの動作設定は設定ファイルによって行います。エージェントの設定ファイルはインスタンス作成時に指定したspool-dirのディレクトリ配下の以下の場所に格納されます。

```
jcexecutor/agent/jcexecutor_agent.yaml
```

jcagctrl　createコマンドによってインスタンスを作成したときに自動的に設定ファイルが作成されます。

設定ファイルの変更内容をエージェントに反映するためにはエージェントを再起動する必要があります。このため設定変更時は必ずエージェントを再起動して下さい。エージェントの停止・起動方法については2章 「エージェントの環境構築」 を参照してください。

3.1.1. エージェントの設定一覧

設定ファイルに設定できるパラメータは以下の通りです。

タイムアウトの設定やデバッグログの設定といった設定については通常デフォルト値のままとしていただいて問題ありません。

表3.1 エージェントの設定ファイルのパラメーター覧

パラメータ			**説明**
debug			デバックモードの有効/無効を設定します。 本パラメータは必ず設定する必要があります。 設定可能な値は以下です。 ■true 有効にします。 ■false 無効にします。 デフォルト値は false です。 有効の場合には、ログにデバッグログを出力します。
websocket	timeout	keepAlive	websocketのpingフレームを送る送信間隔を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 30 です。
		pongWait	websocketのpongフレームの待ち時間を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。

パラメータ			説明
			デフォルト値は 60 です。
		writeWait	メッセージ送信のタイムアウト時間を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 10 です。
		handshake	コネクション確立時のハンドシェイクのタイムアウト時間を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 45 です。
	connectRetryInterval		WebSocketの接続リトライの間隔を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 30 です。
api	timeout	keepAlive	HTTPキープライブで定期的に送信するパケットの間隔を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 30 です。
		idleConnTimeout	HTTPキープアライブで使用されていないコネクションのタイムアウト時間を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 60 です。
		handshake	コネクション確立時のハンドシェイクのタイムアウト時間を指定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 45 です。
log	errorLog	maxSize	エラーログファイルの 1 ファイルの最大サイズ(MB)を設定します。 本パラメータは必ず設定する必要があります。 最大サイズ(MB)は　4～100　の範囲で設定できます。

パラメータ				説明
				デフォルト値は 4 です。
		maxBackups		エラーログファイルのバックアップ数を設定します。 本パラメータは必ず設定する必要があります。 バックアップ数は 1～10 の範囲で設定できます。 デフォルト値は 10 です。 設定したバックアップ数よりも古いログファイルは自動的に削除されます。
request	onStartup	action		エージェントの停止時に実行中だったリクエストの再実行に関する動作を設定します。 設定可能な値は以下です。 ■default リクエストのリスタートの設定にしたがって動作します。 ■abort リクエストのリスタートの設定を無視して異常終了させます。 未設定の場合には default がデフォルト値になります。
	restart	resultAppend		エージェント再起動時のリクエスト再実行における出力結果の追記の有効/無効を指定します。 設定可能な値は以下です。 ■true 有効にする。 ■false 無効にする。 デフォルト値は true です。 無効の場合には出力結果を上書きします。この場合エージェントの停止前の出力結果を後から確認することはできません。
	environment	export	path	リクエスト実行時の環境変数 PATH の扱いを指定します。 設定可能な値は以下です。 ■default エージェントの環境に合わせて自動的に値を設定します。

パラメータ				説明
				■manager マネージャ側のリクエスト実行時の環境変数 PATH の値を使用します。 未設定の場合には default がデフォルト値になります。 エージェントの環境に合わせて自動的に設定する値については<スタンダードモード用環境構築ガイド>の14章　「ジョブ実行時の環境変数の取り扱い」 を参照してください。
		userProfile		エージェント全体の設定としてリクエスト実行時のユーザプロファイルの読み込みの有効/無効を指定します。 設定可能な値は以下です。 ■true 有効にする。 ■false 無効にする。 デフォルト値は true です。 本設定はWindowsでのみ有効です。Linuxの場合は動作に影響しません。
		userEnvironment		エージェント全体の設定としてリクエスト実行時のユーザ環境変数の読み込みの有効/無効を指定します。 設定可能な値は以下です。 ■true 有効にする。 ■false 無効にする。 デフォルト値は true です。 本設定はWindowsでのみ有効です。Linuxの場合は動作に影響しません。
		userSetting		リクエスト実行時のユーザプロファイルとユーザ環境変数の読み込みの有効/無効をユーザ毎に個別に指定します。 本設定はエージェント全体としての設定より優先されます。 本設定はWindowsでのみ有効です。Linuxの場合は動作に影響しません。

パラメータ				説明
			userName	対象のユーザ名を指定します。 ドメインユーザの場合は**ドメイン\ユーザ名**の形式で指定します。
			userProfile	リクエスト実行時のユーザプロファイルの読み込みの有効/無効を指定します。 設定可能な値は以下です。 ■true 有効にする。 ■false 無効にする。 デフォルト値は true です。
			user Environment	リクエスト実行時のユーザ環境変数の読み込みの有効/無効を指定します。 設定可能な値は以下です。 ■true 有効にする。 ■false 無効にする。 デフォルト値は true です。
	clearUserToken			リクエスト実行時に使用するユーザトークン情報のキャッシュの動作を設定します。 設定可能な値は以下です。 ■never 常にキャッシュを利用する。 ■always キャッシュを利用せず、常にユーザトークンの取得を行う。 未設定の場合には never がデフォルト値になります。 本設定はWindowsでのみ有効です。Linuxの場合は動作に影響しません。
	temporary ResultFile	maxSize		エージェント側で保持する出力結果ファイルの合計サイズの上限を指定します。 値の範囲は 0～1000000 で単位は(MB)です。 デフォルト値は 0 です。

パラメータ			説明
			1以上の値が指定された場合、保持している出力結果ファイルの合計サイズが指定値を超過した場合、古いファイルから削除されます。 0が指定された場合、上限はなしとなります。 エージェント側で保持する出力結果ファイルについては「3.1.7 巨大な出力結果の退避によるディスク使用量を制限する」を参照してください。
	sendResult	maxSendSizeAtOnce	リクエストの出力結果をマネージャに一度に送信できるデータサイズの上限を指定します。 値の範囲は 0～100 で単位は(MB)です。 デフォルト値は 10 です。 一度にマネージャに送信するデータサイズが上限を超えた場合は、複数回に分割して送信します。
		sendInterval	リクエストの出力結果をマネージャに送信する間隔を指定します。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 60 です。
timeout	boot		エージェントの起動のタイムアウトを設定します。 本パラメータは必ず設定する必要があります。 値の範囲は 1～3600 で単位は秒です。 デフォルト値は 300 です。
filecheck	confirmClose		ファイル待ち合わせ部品で通過条件が「存在する」または「更新」の場合の追加条件の有効/無効を指定します。 設定可能な値は以下です。 ■true 有効にします。 ■false 無効にします。 デフォルト値は false です。 有効とした場合、通過条件を満たした際に対象のファイルが他プロセスによってオープンされているかを確認します。 このとき対象ファイルが他プロセスによってオープンされていない場合に待ち合わせを終了します。

パラメータ		説明
	confirmNoUpdate	ファイル待ち合わせ部品で通過条件が「存在する」または「更新」の場合の追加条件の有効/無効を指定します。 設定可能な値は以下です。 ■true 有効にします。 ■false 無効にします。 デフォルト値は false です。 有効とした場合、通過条件を満たした際に一定時間の間に対象のファイルが更新されているかを確認します。 このとき noUpdateTime の設定値以上の時間更新されていない場合に待ち合わせを終了します。
	noUpdateTime	confirmNoUpdateが有効の場合に確認する時間を指定します。 confirmNoUpdateが無効の場合は本パラメータの設定は無視されます。 値の範囲は 1～300 で単位は秒です。 デフォルト値は 30 です。

3.1.2. マネージャへの接続のリトライ間隔を変更する

エージェントは起動時にマネージャに対してWebSocket接続を行い、この接続によってマネージャからのリクエスト実行指示を受け付けます。

このため、WebScoketの接続が切断されている間はマネージャからエージェントに対して新たにリクエストを実行することができません。

WebSocket接続は一度確立した後、基本的には切断されることはありませんが、マネージャの再起動が発生した場合などに一時的に切断される可能性があります。

WebSocket接続が切断された場合、エージェントは即座に再接続を試みます。このとき接続に失敗すると、以降は一定間隔でマネージャへの接続リトライを行い続けます。

このため、マネージャが再起動して接続が切断された後、エージェントとの接続が回復してマネージャからのリクエスト実行が再開されるまで、最大でリトライの間隔分の時間がかかる可能性があります。

この時間を短くしたい場合などに、エージェントの設定ファイル（jcexecutor_agent.yaml）で**connectRetryInterval**の値を変更することでリトライ間隔を変更できます。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）でリトライ間隔を20秒とする場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
websocket:
  timeout:
```

```
    keepAlive: 30
    pongWait: 60
    writeWait: 10
    handshake: 45
  connectRetryInterval: 20
```

エージェントの接続リトライの間隔を短くしすぎるとマネージャを長時間停止する場合などに不要な通信が多く発生することになります。またこの場合リトライ毎に接続失敗の旨のログが出続けることになりますので、必要なログが流れてしまうなどの可能性があります。

3.1.3. ログサイズ、世代数を変更する

エージェントのログはspool-dirのディレクトリ配下の以下の場所に格納されます。

```
log/jcexecutorAgent.log
```

ログのサイズや保持する世代数を変更したい場合、エージェントの設定ファイル（jcexecutor_agent.yaml）で**log.errorLog**の**maxSize**や**maxBackups**の指定値を変更します。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）でログのサイズを10(MB)、世代数を5とする場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
log:
  errorLog:
    maxSize: 10
    maxBackups: 5
```

3.1.4. 再起動時の動作を変更する

エージェントは停止時に実行中のすべてのリクエストを停止してから終了します。

また停止したリクエストはエージェントの再起動時に、リクエストのリスタート設定にしたがって動作します。

リクエストのリスタート設定は、リクエストに紐づく単位ジョブのパラメータのリスタートの設定値となります。単位ジョブのリスタートの設定については<スタンダードモード用基本操作ガイド>の「4.2.2.5 ［その他］タブ」を参照してください。

■ENABLE

リクエストを再実行します。この場合リクエストのバッチ処理を最初からやりなおします。

■DISABLE

リクエストを異常終了扱いとしてマネージャに通知します。この場合リクエストに紐づく単位ジョブは異常終了となります。

またエージェント停止前に出力されていたリクエストの結果は削除せずに保持します。このため、再実行後のリクエストの出力結果は以下のような区切り文字の後に追記されます。

```
======< JNW RERUN YYYYMMDDThhmmssZ    >==================== Request Restarted
```

実際の出力では YYYYMMDD は年月日、 hhmmss は時分秒に置きかわります。

3.1.4.1. 再起動時に全てのリクエストを終了する

エージェント再起動時に全てのリクエストを再実行したくない場合、エージェントの設定ファイル（jcexecutor_agent.yaml）で**request.onStartup**の**action**を abort と設定することで全てのリクエストを異常終了扱いとする動作となります。

この場合リクエストのリスタート設定は無視されます。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）で設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  onStartup:
    action: abort
```

3.1.4.2. リクエストの再実行時に以前の結果を残さず上書きする

リクエストの再実行時にエージェント停止前に出力されていたリクエストの結果を削除したい場合、エージェントの設定ファイル（jcexecutor_agent.yaml）で**request.restart**の**resultAppend**を false に設定することで出力結果の追記を行わず上書きする動作となります。

この場合区切り文字は出力されません。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）で設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  restart:
    resultAppend: false
```

3.1.5. リクエスト実行時の環境変数の設定を変更する

リクエスト実行時の環境変数は、エージェントの設定およびマネージャ側の設定によって決まります。

本節では、エージェント側で行う環境変数の設定変更について説明します。

マネージャ側の設定やデフォルトで設定される環境変数の詳細については<スタンダードモード用環境構築ガイド>の14章 「ジョブ実行時の環境変数の取り扱い」 を参照してください。

3.1.5.1. リクエスト実行時の環境変数 PATH をマネージャ側でまとめて設定する

リクエスト実行時の環境変数 PATH の値はリクエストを実行するエージェントの環境に合わせて自動的に設定します。

エージェントの環境に合わせて自動的に設定する値については<スタンダードモード用環境構築ガイド>の14章 「ジョブ実行時の環境変数の取り扱い」 を参照してください。

このため、例えば特定のアプリケーションの実行ファイルに追加でパスを通す必要がある場合、以下のような対応が必要となります。

■リクエストのバッチ処理の最初で環境変数 PATH を修正する

■自動的に設定された PATH に含まれる場所にシンボリックリンクなどを配置する

このため、エージェントの環境が多数ある場合、同様の対応が必要な環境が増えるほど設定の手間が大きくなっていくと考えられます。

このような場合に、エージェントの設定ファイル（jcexecutor_agent.yaml）で**request.environment.export**の**path**を manager とすることでマネージャ側の環境変数 PATH を引き継ぐ動作に変更できます。

例えばジョブネットワークの環境変数として　PATH　を指定した場合、本設定を行ったエージェントで実行されるリクエストの実行時の環境変数 PATH はジョブネットワークで指定した値となります。このため、マネージャ側で必要に応じてジョブネットワーク単位で環境変数 PATH を指定するといった対応ができます。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）で設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  environment:
    export:
      path: manager
```

3.1.5.2. リクエスト実行時にユーザプロファイルとユーザ環境変数を読み込む設定を変更する(Windows)

Windowsの場合、リクエストの実行時にユーザプロファイルとユーザ環境変数を読み込んでリクエストを実行します。

ユーザプロファイルの読み込み処理が他プロセスと競合することでリクエストの実行で異常が発生する可能性があります。特にウイルススキャンソフトがプロファイルデータをスキャンしていると発生する可能性が高いため、スキャン対象外にする等の設定をシステム側で行う必要があります。

詳細については、<セットアップガイド>の「2.2　インストールの準備をする」を参照してください。

この動作を変更したい場合、エージェントの設定ファイル（jcexecutor_agent.yaml）で**request.environment**の**userProfile**と**userEnvironment**をそれぞれ false とすることでユーザプロファイルとユーザ環境変数を読み込まないように変更できます。

ユーザプロファイルとユーザ環境変数の読み込みはリクエストの実行毎に行うため、読み込まないように設定することで処理性能が向上する場合があります。

バッチ処理の内容にユーザプロファイルの読み込みやユーザ環境変数が必要なコマンドなどが含まれる場合、バッチ処理が正常に終了しなくなる可能性があります。

またユーザプロファイルの読み込みのみを　false　とした場合、該当ユーザのユーザ環境変数を一部取得できません。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）でユーザプロファイルとユーザ環境変数の読み込みを無効にする場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  environment:
    userProfile: false
    userEnvironment: false
```

またユーザプロファイルとユーザ環境変数の設定はリクエストの実行ユーザ毎個別に設定することもできます。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）でユーザ毎に設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  environment:
    userSetting:
      - userName: Administrator
        userProfile: false
        userEnvironment: false
      - userName: user1
        userProfile: false
        userEnvironment: false
```

3.1.6. リクエストの実行ユーザの設定変更を即時に反映できるように変更する(Windows)

Windowsの場合、リクエストの実行ユーザ切り替えのために内部的に該当ユーザでのログオンがが必要となります。

デフォルトではこのログオンの回数を減らすために、一度ログオンしたときに取得したユーザのトークンをキャッシュして利用します。

またWindowsにおいて権限などの一部情報はトークンに含まれますので、ユーザ情報を変更した場合にエージェントを再起動して再ログオンするまで変更が反映されない可能性があります。

変更を即時に反映するようにしたい場合、エージェントの設定ファイル（jcexecutor_agent.yaml）で**request**の**clearUserToken**を always とすることでキャッシュを利用せず、リクエストの実行毎にログオンするように動作を変更できます。

この場合、リクエストの実行毎にログオンすることになりますので性能が下がる可能性があります。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）で設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  clearUserToken: always
```

3.1.7. 巨大な出力結果の退避によるディスク使用量を制限する

リクエストの出力結果をマネージャに転送する際、転送するサイズには制限があります。転送するサイズの制限はマネージャ側で設定された値に依存します。

リクエストの出力結果がマネージャ側で設定された転送するサイズを超過している場合、超過した分のデータがマネージャ側に転送されません。

この場合、リクエストの終了時にエージェント側でspool-dirのディレクトリ配下の以下の場所にリクエストの出力結果を退避します。

```
jcexecutor/agent/result/<リクエストID>/stdout
```

```
jcexecutor/agent/result/<リクエストID>/stderr
```

リクエストIDはUUIDの形式となります。CL/Win上では表示されないため、マネージャ側で**jcexectrl get**で確認いただく必要があります。

デフォルトの動作ではこの退避されたファイルは削除されません。このため、何らかの理由により巨大な出力結果が続くとエージェント側のディスク使用量を圧迫する可能性があります。

これを回避したい場合、エージェントの設定ファイル（jcexecutor_agent.yaml）
で**request.temporaryResultFile**の**maxSize**の値を　0　以上の数値とすることで退避ファイルの合計サイズ(MB)の上限を設定することができます。

出力結果ファイルの退避時に退避ファイルの合計サイズが設定値を超過した場合、作成日時が古い退避ファイルから順番に上限サイズに収まるように自動的に削除します。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）で設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  temporaryResultFile:
    maxSize: 1000
```

3.1.8. 実行中リクエストの出力結果の更新頻度を変更する

実行中のリクエストの出力結果は定期的にマネージャ側へ転送されます。デフォルトでは60秒ごとになっています。

このため、CL/Winの出力結果などは60秒毎に更新されることになります。

マネージャへの転送間隔を変更したい場合は、エージェントの設定ファイル（jcexecutor_agent.yaml）
の**request.sendResult**の**sendInterval**の値を変更します。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）で設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
request:
  sendResult:
    sendInterval: 1000
```

3.1.9. ファイル待ち合わせの動作を変更する

ファイル待ち合わせの通過条件が「存在する」または「更新」の場合、デフォルトではファイルの存在や更新を確認した時点で待ち合わせを終了します。

このため、ファイル転送によって新規に作成・更新されるファイルを対象にした場合など、対象のファイルの転送完了を待ち合わせることはできません。

こういったファイルの転送完了を待ち合わせたい場合などに、ファイル待ち合わせの通過条件が「存在する」または「更新」の場合、待ち合わせの終了条件として以下の判定を追加できます。

■通過条件を満たした際に対象のファイルが他プロセスによってオープンされていないことを確認します。

■通過条件を満たした際に一定時間の間に対象のファイルが更新されていないことを確認します。

通過条件に加えて、これらの条件を満たした場合にファイル待ち合わせを終了する動作となりますので、転送完了を待ち合わせることができます。

ファイル作成中、ファイルがオープンされたままとなる場合にはプロセスのオープン確認のみ行えば問題ありません。そうではなく、オープンとクローズが繰り返されるような場合は最終更新時刻から一定時間ファイルが更新されていないことを以てファイル作成が完了したと見なすようにしてください。これらの両方を指定することも可能です。

これらの設定は、エージェントの設定ファイル（jcexecutor_agent.yaml）
で**filecheck**の**confirmClose**、**confirmNoUpdate**を変更することで有効にできます。

また**filecheck**の**noUpdateTime**の値によって、確認する更新時間を変更できます。

以下は、エージェントの設定ファイル（jcexecutor_agent.yaml）で設定を行う場合のサンプルです。なお設定ファイルのうち一部のパラメータのみの抜粋となりますのでご注意ください。

```
filecheck:
  confirmClose: true
  confirmNoUpdate: true
    noUpdateTime: 20
```

3.2. エージェントの構成情報のバックアップ・復元

本章ではエージェントの設定ファイルなどの構成情報のバックアップと復元方法について説明します。

3.2.1. エージェントの構成情報のバックアップ

バックアップが必要となるファイルについて、手動でバックアップを行います。このときエージェントは起動したままで問題ありません。

バックアップが必要となるファイルは以下の通りです。

表3.2 エージェントの構成情報のバックアップファイル

ファイル	説明
■Linux <インストールディレクトリ>/data/agdata ■Windows <%ProgramDataのパス%>\JobCenter\Agent\agdata	エージェントのインスタンスデータ
■Linux <インストールディレクトリ>/data/aglang ■Windows <%ProgramDataのパス%>\JobCenter\Agent\aglang	エージェントの言語設定データ
<spool-dirのパス>/log/jcexecutor_agent.yaml	インスタンス毎の設定ファイル spool-dirのパスは**jcagctrl list --format json --pretty-print**の**spool-dir**の値から確認できます。
<インスタンスの秘密鍵>	インスタンス毎の秘密鍵ファイル **jcagctrl list --format json --pretty-print**の**private-key**の値から確認できます。
<インスタンスのルート証明書>	インスタンス毎のルート証明書 **jcagctrl list --format json --pretty-print**の**root-ca**の値から確認できます。

インスタンス毎のユーザグループに所属するユーザについては、OS側のユーザグループで管理されており、本構成情報のバックアップ・復元対象には含まれていません。

所属するユーザー覧のみであれば**jcagctrl list-user --instance-name <インスタンス名>**で取得できるため、この結果をとっておくことを推奨します。

3.2.2. エージェントの構成情報の復元

■事前に全てのインスタンスを停止する必要があります。

インスタンスの停止方法については2章 「エージェントの環境構築」 を参照してください。

■対応するエージェントキューにリクエストが残っている状態で復元しないでください。

リクエストが残っている状態でエージェントのみを復元した場合、リクエストに対応する単位ジョブが異常終了する可能性があります。

3.2.2.1. リクエストデータの削除

以下のディレクトリ配下の全てのファイル（リクエストのデータ）を削除します。

```
<spool-dirのパス>/jcexecutor/agent/request/
```

全てのインスタンスのspool-dirに対してリクエストのデータ削除を行ってください。

3.2.2.2. バックアップファイルの復元

バックアップしたファイルについて元の場所に配置（上書き）します。

エージェントの言語データとインスタンスデータのファイル配置後、各インスタンスのファイルを配置します。

別環境で復元する場合、各インスタンスデータの配置前に事前にspool-dirのディレクトリを作成する必要があります。

ディレクトリ作成後、秘密鍵、ルート証明書を配置し、**jcagctrl update**を一回実行いただくことでspool-dir配下が初期化されますので、その後インスタンス毎の残りのファイルを配置してください。

spool-dirで指定するディレクトリのパーミッションは755である必要があります。umaskを設定されている場合など、ディレクトリ作成後必要に応じてパーミッションを変更してください。

3.2.2.3. エージェントの起動

リクエストデータの削除、バックアップファイルの上書き完了後、各インスタンスを起動します。

インスタンスの起動方法については2章 「エージェントの環境構築」 を参照してください。

4. OS要塞化

この章ではSELinux設定およびユーザ環境をセキュアにしたサーバ環境におけるJobCenter　AGの運用に関して説明します。対象となるOSは以下となります。

■Red Hat Enterprise Linux 7

■Red Hat Enterprise Linux 8

■Red Hat Enterprise Linux 9

SELinux設定をセキュアにしたサーバ環境とは、/etc/selinux/configファイルにSELINUX=enforcingを設定した環境を指します。

■SELinux設定をセキュアにしたサーバ環境におけるJobCenter AGの運用に関する留意点を以下に示します。

- JobCenter　AG用に提供されるSELinuxポリシーファイル(jobcenter.pp)は、JobCenterの動作のベースとなるものとして提供しており、JobCenter運用環境において必要に応じてSELinuxポリシーをカスタマイズして頂くことを想定しています。そのため、JobCenter運用環境においてSELinuxに関する修正や設定を適切に行う体制が無い場合は、SELinux設定ファイルのSELINUX(/etc/selinux/configファイルのSELINUX=の設定)にpermissive(あるいはdisabled)を選択してください。
- SELinux設定ファイルのSELINUXTYPE(/etc/selinux/configファイルのSELINUXTYPE=の設定)は、既定値のtargetedのまま変更しないでください。
- JobCenterの動作に関連するOSのログインユーザのSELinuxコンテキストは、既定値の unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 のまま変更しないでください。

ユーザ環境をセキュアにしたサーバ環境とは、ユーザにおけるアクセス権が制限されている環境（umask設定を既定値から077等に設定変更）を指します。

■ユーザ環境をセキュアにしたサーバ環境におけるJobCenter AGの運用に関する留意点を以下に示します。

- （JobCenter自体の動作には問題ありませんが、）umask設定が077等に変更された環境では、JobCenterのコマンドを実行してファイル出力する場合、出力(保存)されるファイルの権限が変更後のumask設定の影響を受けます。

5. トラブルシューティング

JobCenterのトラブルシューティングに関する情報を記載しています。

5.1. JobCenter Agent の障害発生時、原因究明に必要な情報を採取する

JobCenter Agent で障害が発生しサポート窓口へお問い合わせいただく際に、お問い合わせの内容によっては原因究明に必要な情報を採取していただく必要があります。本章では情報の採取方法について説明します。

5.1.1. jc_getinfoコマンドで採取する場合

jc_getinfoコマンドで採取する手順は以下のとおりです。

1. 障害が発生したマシン上でjc_getinfoコマンドを実行し、情報を採取します。

   jc_getinfoコマンドの詳細は<スタンダードモード用コマンドリファレンス>の「8.3 jc_getinfo JobCenterの障害発生時、原因究明に必要なエージェントに関する1次情報を漏れなく採取」を参照してください。

発行年月 May 2024
NEC Corporation　2024