クラシックモード用運用・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　の運用・構築ガイドです。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。あらかじめご了承ください。

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

→ <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

→ <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

→ <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

→ <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

→ <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

→ <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

→ <クラシックモード用運用・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

→ <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

→ <クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド>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. 用語集

JobCenterを使用する上で大切な用語を次に説明します。

用語	説明
CL/Win	本書では、WebSAM JobCenter CL/Winを指します。
MG	本書では、WebSAM JobCenter MGを指します。
SV	本書では、WebSAM JobCenter SVを指します。
ジョブ	業務プロセスの単体または固まりを一単位として定義したものです。JobCenterでのジョブは、 ユーザから見た処理の単位で、単一もしくは連続的なプログラム群です。使用するプログラムは、シェルスクリプト（UNIX）もしくはバッチファイル（Windows）です。
ジョブネットワーク（JNW）	ジョブネットワークは、ジョブをグループ化したものでJobCenterの最も基本的な単位です。JobCenterにおけるジョブの実行順序、即時投入やスケジュール投入などのジョブの運用はすべてジョブネットワークを基本単位として行われます。ジョブネットワークの操作中でジョブの実行順序や実行条件を定義します。ジョブネットワークをJNWと表記することがあります。
スケジュール	スケジュールは、ジョブネットワークを自動実行するための定義です。スケジュールとジョブネットワークを関連付けて、ジョブを自動実行させます。
稼働日カレンダ	稼働日カレンダはサイトごとにジョブの運用を行う日を定義したカレンダです。 ジョブの運用を行うように定義された日を「稼働日」と呼び、ジョブの運用を行わないように定義された日を「休止日」と呼びます。稼働日カレンダはJobCenter管理者権限のあるユーザのみが作成できます。 各ユーザは稼働日カレンダを作成するスケジュールに適用することにより、作成しているジョブの自動実行スケジュールに稼働日相対や休日シフトを組み合わせることで、休止日を考慮したスケジュールを作成できます。
カスタムジョブ定義	カスタムジョブは、それぞれのジョブネットワークで共通に使用できるスクリプトをテンプレートとして定義したものです。作成したカスタムジョブは、ジョブネットワーク内に部品オブジェクトとして配置できます。 カスタムジョブを一度作成し、複数のジョブネットワークで使用すれば、各ジョブやネットワークごとに同じスクリプトを編集する必要がありません。また、特別なアイコンにすることができるので、監視時にジョブを見つけやすくできます。
トラッカ	ジョブネットワークの定義情報に対し、ジョブネットワークを実行した状態情報もしくは結果情報のことをトラッカと呼びます。定義情報が1つであるのに対して、トラッカは定義情報のインスタンスとしてその実行ごとに生成されます。そして、ログやステータスなどジョブネットワークの実行状況や結果情報を保持します。 また、単位ジョブの実行結果を単位ジョブトラッカ、ジョブネットワークの 実行結果をフローで表したものをトラッカフローと呼ぶこともあります。
NQS	Network Queuing Systemというジョブ管理を行うためのコンポーネントおよびプロトコルの名称です。ジョブ管理の分野において広く使われている標準的なプロトコルであり、キュー制御によるリクエストの管理を実施します。
リクエスト	NQSにおける基本処理単位です。ジョブネットワークフローで単位ジョブとして定義されているものがそのまま1つのリクエストとなります。
キュー	一時的にリクエストを溜めておく場所です。NQSではこのキューに溜まっているリクエストを順番に処理していきます。キューには後述するパイプキューとバッチキューの2種類があります。
パイプキュー	リクエストを別のキューに転送するためのキューです。
バッチキュー	リクエストをジョブとして実行するためのキューです。

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 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. 凡例 vi
3. 用語集 vii
4. 関連マニュアル viii
5. 改版履歴 x
1. 設計編 1
1.1. 製品構成について 2
1.1.1. 基本的な製品構成 2
1.2. マシン構成の設計 4
1.2.1. JobCenter MG、JobCenter SVの設計 4
1.2.2. 可用性の検討 4
1.2.3. ネットワーク構成 4
1.2.4. セットアップ言語 5
1.2.5. マシンID 5
1.3. 運用形態の設計 6
1.3.1. ユーザの設計 6
1.3.2. 監視設計 8
2. 構築編 10
2.1. インストール、初期セットアップ 11
2.2. JobCenter MG - JobCenter SV間の連携設定 12
2.2.1. マシン登録 12
2.2.2. ユーザIDマッピング 13
2.2.3. キューの設定 14
3. 開発編 17
3.1. ジョブネットワークの概要 18
3.2. ジョブネットワークの設計 19
3.2.1. ユーザ設計について 19
3.2.2. ジョブネットワークグループについて 19
3.2.3. ジョブネットワークの名前について 19
3.3. ジョブネットワークの作成 21
3.3.1. ジョブネットワークのパラメータを表示する 21
3.3.2. ジョブが異常終了したとき、フローを停止したい 21
3.3.3. 同じジョブネットワークの実行が重ならないようにしたい 23
3.3.4. 他のジョブネットワークと実行が重ならないようにしたい 25
3.3.5. 時間超過したものを自動検知したい（または打ち切ってしまいたい） 26
3.3.6. フロー内に配置した各単位ジョブに共通の環境変数を定義したい 28
3.3.7. ジョブの実行ユーザを切り替えたい 30
3.3.8. 単位ジョブの出力結果を後続の単位ジョブで参照したい 31
3.3.9. ジョブの正常終了・異常終了の条件を変更したい 32
3.3.10. ジョブ実行中にJobCenterを停止した場合の制御について 33
3.3.11. ジョブネットワークから他のジョブネットワークを実行したい 34
3.3.12. ユーザからの応答を待ち合わせてから、後続の処理を実行したい 36
3.3.13. ジョブが異常終了した場合に自動でリトライしたい 36
3.3.14. 同じ処理を何度も繰り返し実行し、特定の時刻になったら処理を終了したい 37
3.3.15. 複数の処理を同時に行わせたい 38
3.3.16. 並列分岐の数を変更したい・分岐そのものを消したい 39
3.3.17. メインの処理を実行中に一定時間経過すると次の処理に進めたい 40
3.3.18. ジョブ結果、またはサブジョブネットワーク結果により動作を変更したい 41
3.3.19. 日付によって動作を変更したい 42
3.3.20. 同期処理をするためのポイント 44
3.3.21. 他のマシン・他のユーザのトラッカと同期して処理したい 45
3.3.22. 同一ユーザのトラッカで異なるルートトラッカと同期して処理したい 46
3.3.23. 不要なイベントを残したくない 46

3.3.24. 連携するトラッカで作成したファイルの名前/パスを受信側で利用したい 47
3.3.25. 単位ジョブが終了するまで待ち合わせてから処理を継続したい 48
3.3.26. 他ジョブネットワークが終了するまで待ち合わせてから処理を継続したい 49
3.3.27. ファイルの作成や削除を待ち合わせてから処理を継続したい 50
3.3.28. ある時刻まで待ってから処理を継続したい 52
3.3.29. 大量のジョブ定義を効率的に作成したい 52
3.4. スケジューリング 53
3.4.1. スケジューリングのポイント 53
3.4.2. カレンダの作成 55
3.4.3. スケジュールの作成 55
3.5. 補足：ジョブ実行時の動作について 59
4. 運用編 62
4.1. 実行状況の監視 63
4.1.1. テキスト形式 / グラフィックモード形式での表示 63
4.1.2. トラッカの制御 65
4.2. キューの制御 68
4.2.1. キューの制御 68
4.2.2. JobCenter起動時の制御 69
4.2.3. 補足：JobCenter起動・停止の影響について 69
4.2.4. JobCenter SV停止・再起動 70
4.3. JobCenterサイトの状況の監視 72
4.3.1. JobCenterの自動復旧 72
4.4. 分析 74
4.4.1. トラッカ情報の分析 74
4.4.2. 実行状況の確認 74
4.5. メンテナンス 75
4.5.1. バックアップ・リストア 75
4.5.2. データの移行 76
5. 逆引きリファレンス 77

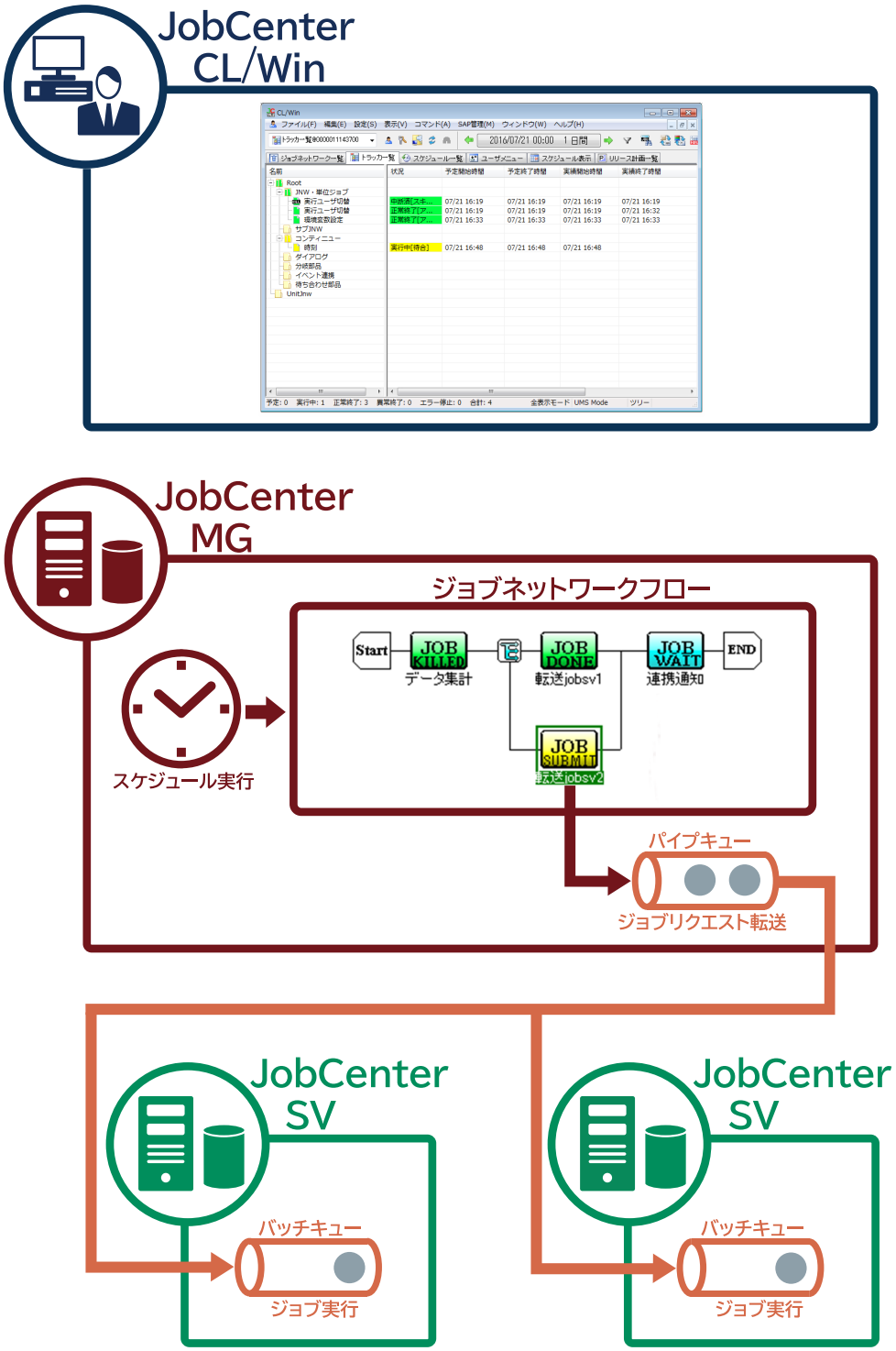
1. 設計編

ここでは、JobCenterを設計・構築する際のポイントについて説明します。

1.1. 製品構成について

1.1.1. 基本的な製品構成

JobCenterは次のような3層構造になっています。



■WebSAM JobCenter CL/Win

WebSAM JobCenter CL/Win（以降、CL/Winとします）は、運用者向けのビューア機能を提供します。また、次の操作をすることができます。

- ジョブネットワークの作成
- スケジューリング設定
- 実行監視

■WebSAM JobCenter MG

WebSAM JobCenter MG（以降、MGとします）は、管理マネージャ機能を提供します。

- スケジューラ機能
- ジョブネットワークのフロー制御、ジョブリクエスト転送
- 監視・通知機能

■WebSAM JobCenter SV

WebSAM JobCenter SV（以降、SVとします）は、NQSをベースにしたジョブ実行機能を提供します。

- ジョブ実行

1.2. マシン構成の設計

1.2.1. JobCenter MG、JobCenter SVの設計

業務要件に応じてシステム全体のマシン構成を決定してください。

■運用管理マネージャ

MGをインストールし、ジョブの集中管理を行うマシンです。

マシンのリソースについては、ピーク時間帯のジョブネットワーク実行数とジョブ実行数を大まかに想定し、 必要となるリソースを見積もる必要があります。

次のようなジョブを扱う大規模システムの場合、運用管理マネージャにはWindowsではなくUNIX/Linuxマシンを推奨します。※

- 1日あたり10万ジョブ以上、または
- 1時間あたり1万ジョブ以上

※： OSのプロセス生成の特性、およびそれに伴うJobCenterのアーキテクチャの違いにより、WindowsとUNIX/Linuxで単位時間あたりに処理できるジョブの数に差があるためです。

参照	<クラシックモード用環境構築ガイド> 23章 「システム利用資源」

■ジョブ実行マシン（業務処理マシン）

SVをインストールし、ジョブとして業務処理を行うマシンです。

SV自体のオーバヘッドは少なく、メモリやディスクはほとんど消費しません。プロセス数とファイル オープン数は増えるため、その部分については次を参照してください。

参照	<クラシックモード用環境構築ガイド> 23章 「システム利用資源」

参照	<クラシックモード用環境構築ガイド> 「23.1.1 nqsdaemon（リクエスト実行）」

マシンのリソースについては、対象マシンで行う業務処理の要件に応じて見積もってください。

1.2.2. 可用性の検討

マシンをクラスタ構成とすることで、システムの可用性を高めることができます。

MG、SVでそれぞれ役割が異なるため、メリットが異なります。

■JobCenter MG

スケジュール起動やジョブネットワークのフロー制御を担うため、フェールオーバー後もそれらの処理を継続的に実施できます。

■JobCenter SV

ジョブとして業務処理を実行することを担うため、フェールオーバー後もジョブを実行できます。

1.2.3. ネットワーク構成

ネットワーク構成は、次のことを考慮に入れて作成してください。

■JobCenterでは通信時にTCP/IPを利用する

■JobCenter MG - JobCenter SV間で名前解決の正引き/逆引きができている必要がある

複数NIC（複数セグメント）を持つマシンの場合、JobCenterが利用する方のネットワークにおいて名前解決できる必要があります。

■利用するポートは<クラシックモード用リリースメモ>の「3.4　使用するネットワークポート」を参照してください

名前解決さえできていればNAT越しの利用は可能です。

NAPT（IPマスカレード）は利用不可となります。

1.2.4. セットアップ言語

JobCenterインストール時に、セットアップ言語（文字コード）を指定する必要があります。ここで指定した文字コードが、対象マシンのJobCenterへの入出力の文字コードになります。

■入力:JobCenterの各種コマンドにおいて、JNW名などのマルチバイト文字を指定する場合の指定方法

■出力:ジョブの出力結果

SVの場合は、それぞれのジョブ（業務処理）を何のLANGで実行するのかで選択してください。必ずしも、対象マシンのシステムのLANGに一致させる必要はありません。また、システム全体でJobCenterのセットアップ言語を統一させる必要はありません。ただし、マシン間でセットアップ言語が異なる　場合はジョブ転送時に変換が必要となり、そのための設定を行う必要があります。

参照	<クラシックモード用環境構築ガイド> 9章 「日本語環境での文字コード変換」

1.2.5. マシンID

マシンIDは、MGおよびSVが持つユニークなIDで、お互いのマシンを識別するために利用されます（重複は不可）。システム設計時にあらかじめ決めておくことをお勧めします。

例) システムのマシン一覧表を作成し、インクリメントを使用して管理する

マシンID	マシン名	IPアドレス
1	jobmanager	192.168.10.1
2	jobserver01	192.168.10.2
3	jobserver02	192.168.10.3

「IPアドレスの下8bit（or　16bit）とマシンIDを一致させるようにする」などのルールにしておくと、管理表を見なくてもどのマシンなのかがわかりやすくなります。

1.3. 運用形態の設計

1.3.1. ユーザの設計

JobCenterではOS上に存在するユーザをそのまま利用します。利用ユーザは、大別すると次の2パターンになります（1ユーザでどちらの役割も兼ねることも可能）。

■ログインユーザ

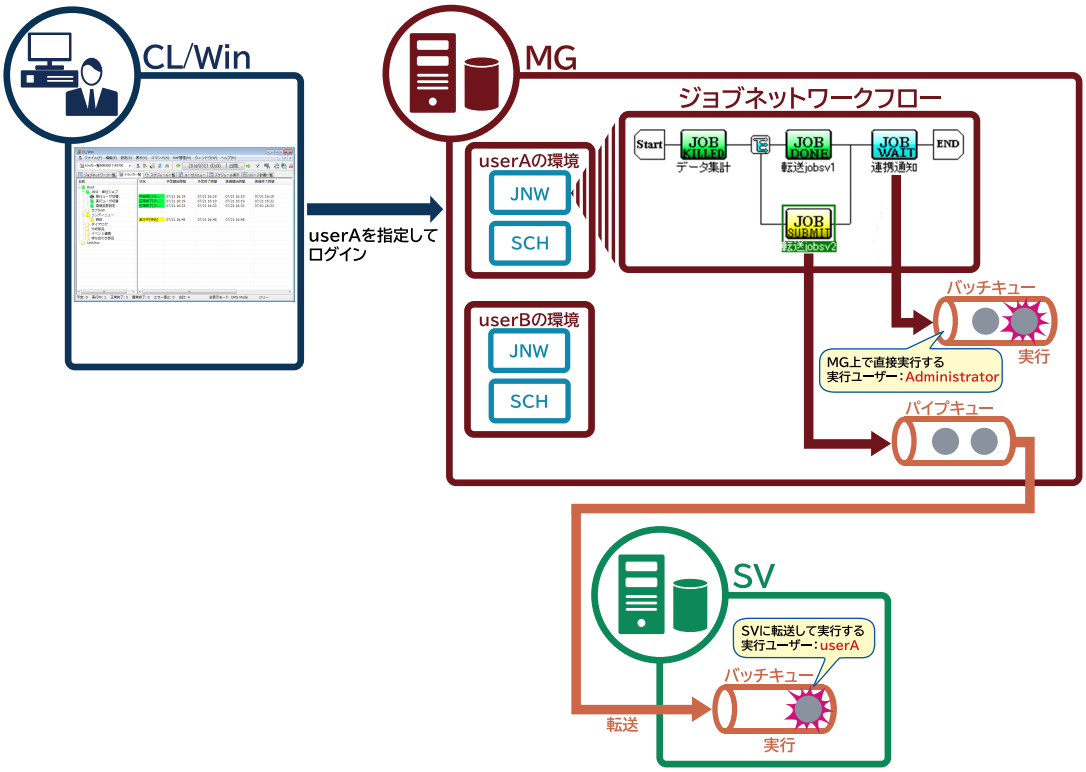
CL/WinからMGにログインし、JNWの作成やスケジュールの設定を行います。また、MGにログインしたユーザには、そのユーザ用のデータ格納領域（ユーザフレーム）が与えられます。

参照　「3.2.1 ユーザ設計について」

■ジョブ実行ユーザ

このユーザの権限でジョブプロセスを起動します。MGとSVのジョブ実行ユーザは、ユーザマッピングという概念で紐付けされます。

参照　「2.2.2 ユーザIDマッピング」



1.3.1.1. アクセス権限の設計

複数のログインユーザを利用する場合、以下を検討し、適切に分業できるようにしてください。

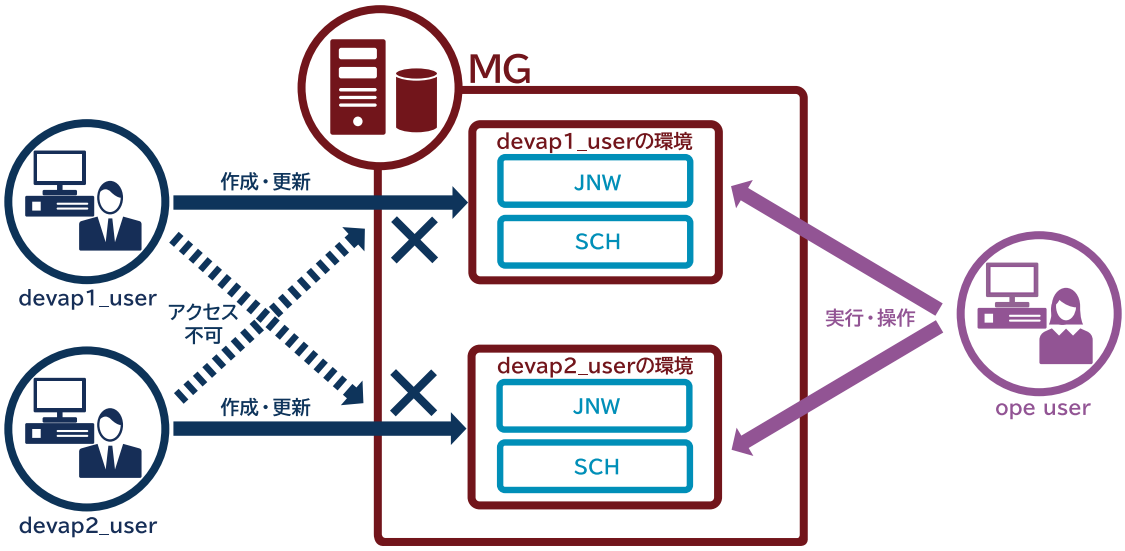
■定義情報（JNWやスケジュール）を保有するユーザ

■定義情報を作成・変更できるユーザ

■実行中のフローを参照・制御できるユーザ

兼任は可能です。システム規模が大きくなる場合に、分業を検討します。

参照	<クラシックモード用環境構築ガイド> 10章 「ユーザ権限（パーミッション設定）」



図解説

■AP1サブシステムのために、MG上にdevap1_userの環境を作成し、そこに定義情報（JNWなど）を作成します。AP2も同様にします。

■それぞれの定義情報を作成・更新できるのはそれぞれのユーザのみ（他ユーザの定義は更新できない）とします。

■上記とは別にオペレータ用アカウントとしてope_userというユーザをMGに作成しています。このユーザは定義の更新はできないが、実行中のフローを監視・制御する役割を与えるため、devap1_userとdevap2_userの両方の環境へのアクセスを許可します（定義の更新はできず、あくまで実行をされているフローの参照と、それらのジョブの制御のみ）。

1.3.1.2. 補足：ユーザの持つ定義情報について

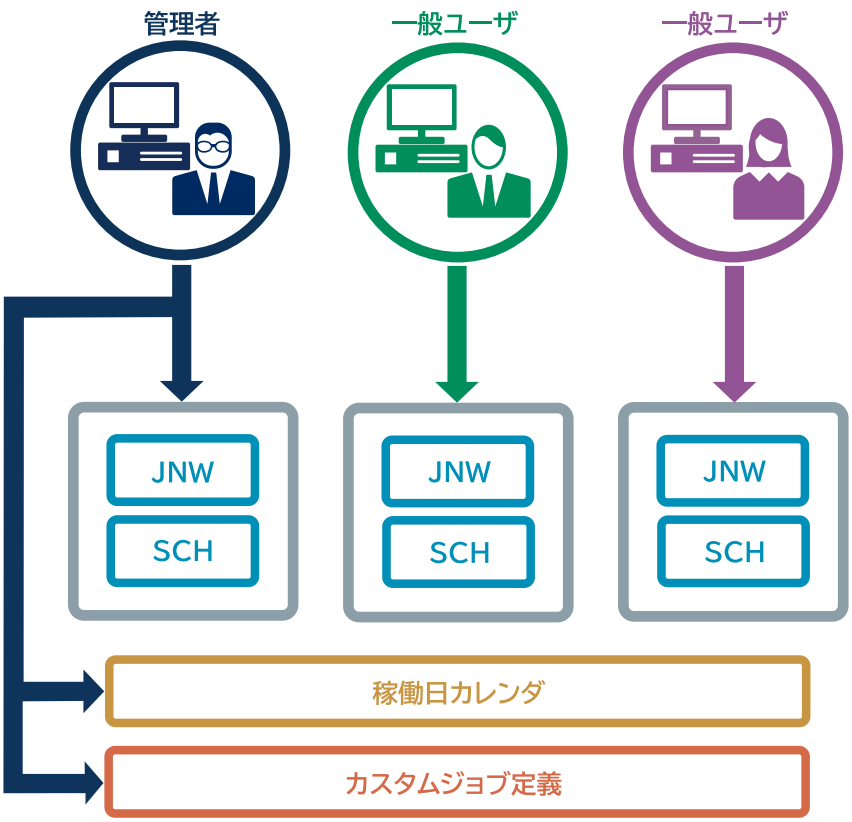
JobCenterのジョブネットワークを開発する上では、次の仕様が前提になっています。この点を考慮し、ジョブネットワークを開発する必要があります。

■JobCenterの利用ユーザはOSのユーザをそのまま利用します。

■定義データでは、ジョブネットワークとスケジュールはユーザの持ち物となり、稼働日カレンダとカスタムジョブ定義はシステムの持ち物となります。

■ユーザがCL/Winからログインすると、それらの定義データを作成できる場所が確保されます。

■管理者のみが、稼働日カレンダやカスタムジョブ定義を作成・更新できます。

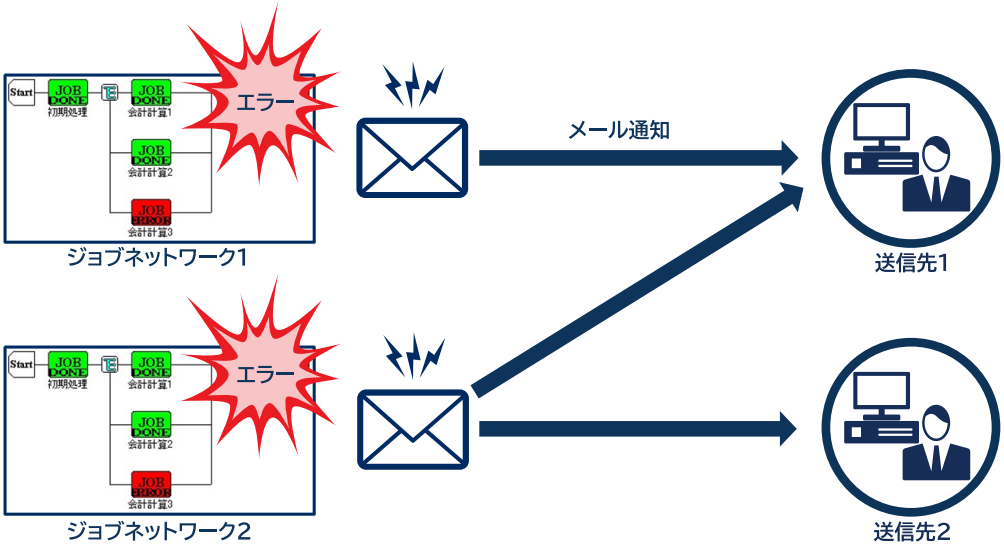


1.3.2. 監視設計

ジョブが異常終了した場合、それをいち早く察知するには、何らかの手段で担当者・管理に通知させる必要があります。この通知方法として次の2パターンがあります。

■メール通報

ジョブが異常終了した場合、指定したメールアドレスにメールを送信します。JobCenterの標準機能であり、JobCenter単独で実現可能です。また、統合監視系製品を導入しない比較的小規模なシステムに向いています。

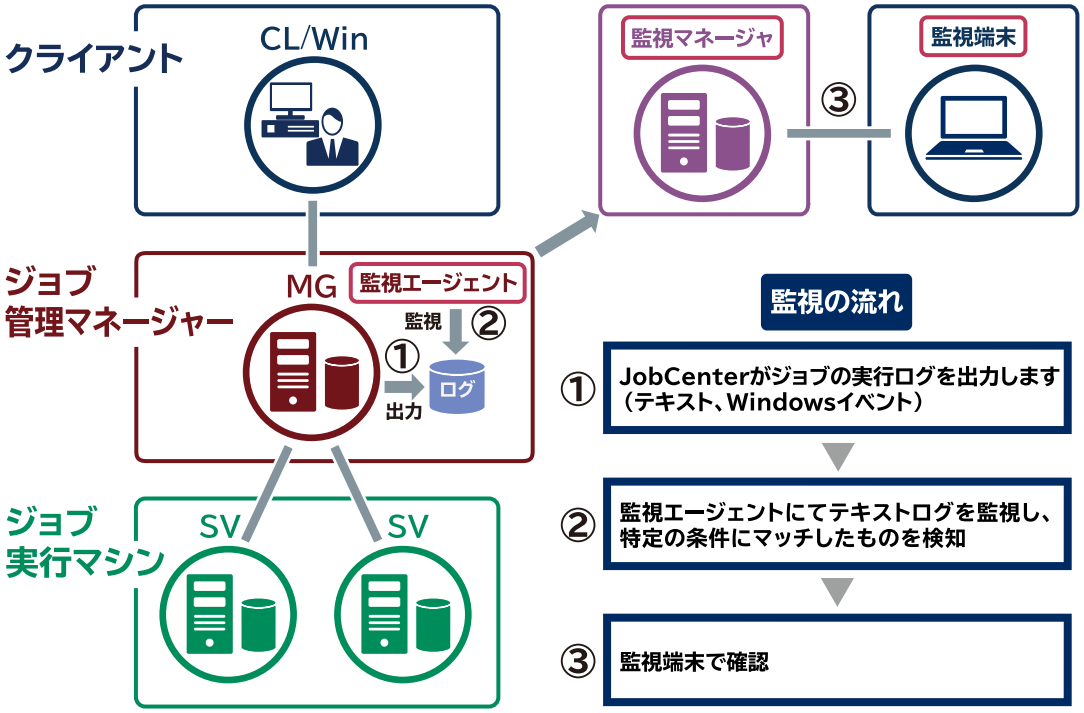


参照	<クラシックモード用基本操作ガイド> 15章 「エラー発生時のメール送信機能の設定方法」

■イベント監視連携

ジョブの開始/終了や時間超過など、様々なイベントを統合監視系製品（※）に通知することができ、オペレータは、他のAPやOS、HW等のアラートと一緒に JobCenterのイベントを監視できます。統合監視系製品を導入する比較的大規模なシステムに向いています。

例えば、監視製品と連携すると、Windowsではイベントログ、またはログファイル出力を、UNIX系のOSの場合には、ログファイル出力を利用し、ジョブの起動・終了（正常・エラー）の状態をメッセージとして集約して監視できます。



（※）統合監視系製品

WebSAM SystemManager G、Micro Focus Operations Manager software等

参照	<クラシックモード用環境構築ガイド> 12章 「イベント連携」

2. 構築編

ここでは、JobCenterのインストールからインストール後に設定すべきポイントについて説明します。

2.1. インストール、初期セットアップ

インストール、初期セットアップ方法については以下を参照してください。

参照	<クラシックモード用インストールガイド> 2章 「インストール」

参照	<クラシックモード用インストールガイド> 3章 「実行環境のセットアップ(UNIX版)」

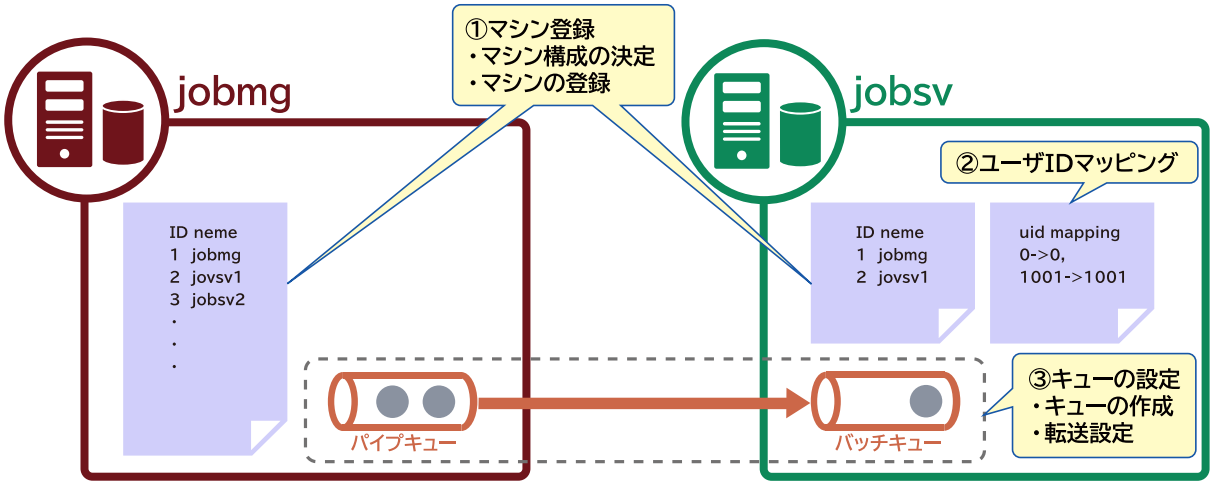
2.2. JobCenter MG - JobCenter SV間の連携設定

MG - SV間でジョブ連携を行うには、次の設定が必要です。

■マシン登録

■ユーザIDマッピング

■キューの設定



2.2.1. マシン登録

MG-SVのマシン登録（マシン構成）の方法には次の2パターンがあります。

・標準リモートマシン構成

・マシングループ構成

基本的には標準リモートマシン構成を選択します。マシングループ構成は特殊な構成のため、次の目的・要件がある場合に選択します。

■ジョブ実行状況に応じて負荷分散を行いたい

負荷分散についてはいくつかの方式がありますが、デマンドデリバリ方式のみマシングループ構成が必須です。

参照	<クラシックモード用NQS機能利用の手引き> 「6.7 負荷分散環境」

■複数あるシステムの各MGがもつトラッカを一元監視したい(1画面で管理したい)

マシングループ構成の場合のみ、マネージャフレームの[トラッカ一覧@全マシン]で各マシンが持っているトラッカを1画面で管理できます。

参照	<クラシックモード用基本操作ガイド> 「8.6 トラッカ一覧をマシンごとにソートして表示する」

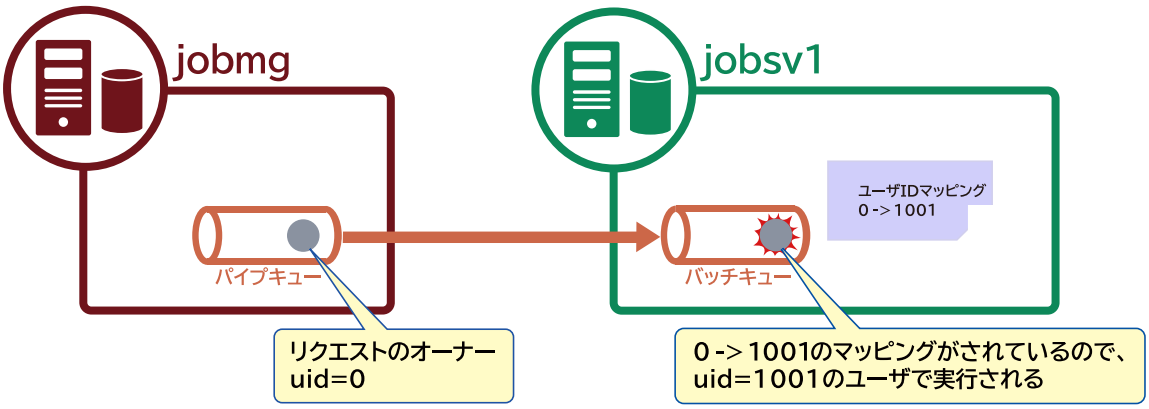
上記以外の場合は、標準リモートマシン構成を推奨します。それぞれの構成の違い・詳細については次のマニュアルを参照してください。マシンの登録方法についてもここに記載しています。

参照	<クラシックモード用環境構築ガイド> 「3.1 ネットワーク上にある他マシンのマシンIDを登録する」

2.2.2. ユーザIDマッピング

ジョブをどのユーザ権限で実行するかの設定は、各ジョブのパラメータで設定できますが、ジョブのパラメータで選択できるのはMG上のOSユーザになります。この場合、MGで直接実行するのであれば特に問題ありませんが、SVに転送して実行させたい場合、SVのどのユーザで実行するのかが問題になります。

JobCenterでは、MGからSVへリクエストを転送してジョブを実行する場合には、MGとSVのそれぞれのユーザIDを紐付けることによって、これを実現しています。（ユーザIDマッピング）

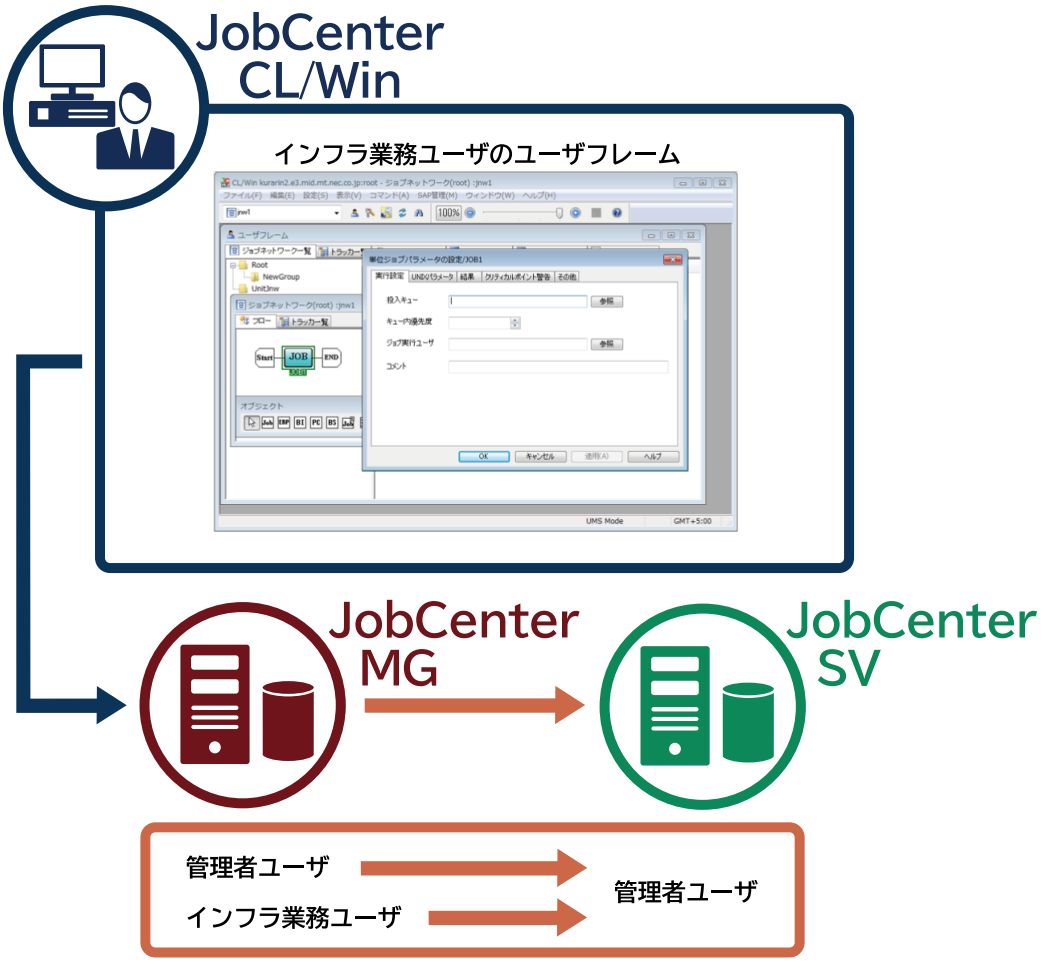


ユーザIDマッピングは個々のジョブのパラメータとして行うものではなく、システム全体の設定として、あらかじめ管理者が行っておく必要があります。

参照	<クラシックモード用環境構築ガイド> 「3.2 ユーザの関連付けを行う（ユーザマッピング）」

2.2.2.1. ユーザIDマッピングの実施例

JobCenterでは、ユーザIDマッピングを利用すると、管理者権限が必要な特定の作業をそれ以外のユーザ権限から実行することができます。



例えば、マシンのメンテナンス作業（再起動、バックアップ、パッチ適用、ログ収集など）を実行したい場合、通常は管理者ユーザ（rootやAdministrator）で行う必要があります。これらの処理をJobCenterのジョブとして登録しておけば定期実行させることができますが、それらのジョブ定義を持つユーザとして管理者ユーザを使いたくない、すなわち、それらの業務担当者（インフラ業務ユーザ）に管理者ユーザのアカウントを教えたくない、といった要件がある場合があります。

このような場合、ユーザIDマッピングにより、上の図のように管理者ユーザ同士のユーザIDマッピングに加えてインフラ業務ユーザと管理者ユーザのユーザIDマッピングを行っておくことで実現できます。この場合、CL/Winから接続するMGのユーザはインフラ業務ユーザであり、そのユーザが持つジョブをSVへ転送して実行する場合、それぞれのSVの管理者ユーザで実行されることになります。

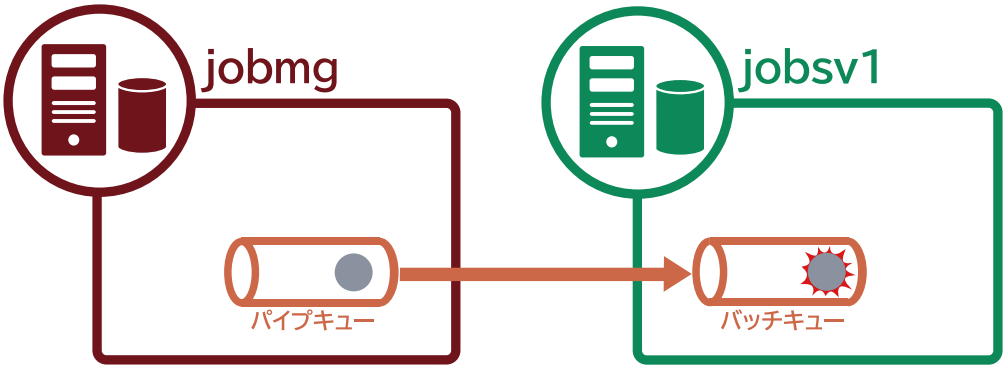
また、JobCenterのアクセス権限の設定を行うことで、インフラ業務ユーザに対して自身のジョブ定義を変更できないようにすることが可能です。この場合、インフラ業務ユーザはあらかじめ作成されているジョブを実行・監視・制御することはできますが、ジョブ定義そのものを変更することはできません。これにより、インフラ業務ユーザが任意の処理を管理者ユーザの権限で実行できてしまう、といったことを防ぐことができます。

2.2.3. キューの設定

MGからSVへリクエストを転送してジョブを実行する場合、次のキューをそれぞれ作成します。

■MG側にリクエスト転送用のキュー（パイプキュー）

■SV側にリクエスト実行用のキュー（バッチキュー）



単位ジョブの投入先を指定する際、MG上のパイプキューではなく、SV上のバッチキューを直接指定することも可能です。ただし、MG上のパイプキューを指定することを推奨します。理由は次の通りです。

■SVの更改等によって投入先SVの変更が必要になった際、各単位ジョブの投入先キューを変更することなく、そのパイプキューの転送先の変更を行うだけで済むため。

■パイプキューの転送先として複数SVのバッチキューを指定することが可能なため。この指定をすると、あるマシンがダウンした場合は、自動的に別のマシンのバッチキューに転送されます。

2.2.3.1. 優先的に重要なジョブを実行したい

キューは、処理内容やその負荷に応じて複数用意しておくと便利です。

例えば、通常行っているジョブと緊急メンテナンス用のジョブが重なった場合に、どちらも同じキューで実行されるように指定していると、緊急メンテナンス用ジョブがキューの中で順番待ちしてしまい、すぐには実行されない場合があります。

このような場合、重要ジョブを実行するための別のキューを用意し、緊急メンテナンス用ジョブはそちらに投入することで、通常ジョブの有無に依存せず、すぐに実行させることができるようになります。

参照 <クラシックモード用環境構築ガイド> 4章 「キューの作成」

2.2.3.2. ジョブの負荷分散をしたい

JobCenterでは、ジョブの負荷に応じて実行させるマシンを分散させることができます。ただし、ここでいう負荷というのは、CPUやメモリの消費量ではなく、キューに空きがあるかどうかです。キューの空きが多い=負荷が低い、キューの空きが少ない=負荷が高いと見なします。

負荷分散には次の2種類の方式があります。

■ラウンドロビン方式

■デマンドデリバリ方式

JobCenterでは負荷分散機能を指定したパイプキューを作成し、転送先に複数のバッチキューを指定することで実現できます。そのため、1台停止しても残りのマシンでジョブを実行できます。

例えば、大量のバッチ処理があり、かつ、あらかじめマシンに固定でジョブを割り振った場合、あるマシンではジョブが早く終わり空きがある一方で、他のマシンでは処理待ちでジョブが遅延する状況が発生することがあります。

このような場合、JobCenterでは、ジョブのリクエストを複数台のマシンへ均等に分散させて、特定のマシンの待ちを発生させずに効率的に早くバッチ処理の実行を行わせることができます。

ただし、このような場合では、n台（1、2、3～n）のマシンにn個（1、2、3～n）のジョブが定期的に発生し、そのn個のジョブが、必ずn個目には処理が重いジョブが割り当てられると、いつもマシンnには処理が重いジョブが滞留しマシンnで処理遅延するような状況が発生してしまいます。このような場合、ジョブの負荷に応じたキュー（低負荷用ジョブキュー、高負荷用ジョブキュー）を作成し、それぞれで分散させることで、より均等な処理を行うことができます。

参照 <クラシックモード用環境構築ガイド> 4章 「キューの作成」

参照 <クラシックモード用NQS機能利用の手引き> 「6.7 負荷分散環境」

3. 開発編

JobCenterでジョブネットワークを開発する上での概要、開発のポイントについて説明します。

3.1. ジョブネットワークの概要

ジョブネットワークは、ジョブをグループ化したものでJobCenter制御の最も基本的な単位です。JobCenterにおけるジョブの実行順序、即時投入やスケジュール実行などのジョブの運用はすべてジョブネットワークを基本単位として行われます。

3.2. ジョブネットワークの設計

ジョブネットワークを設計する上で必要な考え方を説明します。

3.2.1. ユーザ設計について

ジョブネットワーク内のジョブは、明示的に設定しない場合、その所有者の権限で実行されることを考慮した上で、JobCenterとしてどのユーザに何のジョブネットワークを実行させるのかを検討してください。

参照 「2.2.2 ユーザIDマッピング」

ここでは、例として2つのパターンを紹介します。

1. 1ユーザに全ジョブネットワークを持たせる

   1ユーザに全ジョブネットワークを持たせる方法です。この場合、次のメリットがあります。

   ■小規模システム（ジョブネットワーク数：1,000以下）に向いています。

   ■ほとんどAdministratorまたはrootでジョブを実行するシステムの場合、ジョブネットワークを1ユーザの持ち物にする方が、ジョブネットワークの作成が効率的です。

   他のユーザに実行させたい場合のみ、ジョブのパラメータで切り替えます。

   ■1ユーザにジョブネットワークを集約できるため、管理コストを低減できます。

2. 複数のユーザにジョブネットワークを分担させる

   サブシステムごとに管理単位や担当者を分け、複数のユーザにジョブネットワークを持たせる方法です。次のメリットや注意点があります。

   ■規模の大きなシステム（ジョブネットワーク数：1,000以上）に向いています。

   ■JobCenter内でのアクセス権限を設計時に決めておく必要があります。

   参照 「1.3.1.1 アクセス権限の設計」

3.2.2. ジョブネットワークグループについて

ジョブネットワークグループとは、root配下に作成するジョブネットワークをまとめて管理するものです。ジョブネットワークグループの下にジョブネットワークグループを作るといった階層を用いた管理もできます。

設計時に、ジョブネットワークグループをサブシステム単位、処理種別単位、日別・月別といった実行日単位などの、どの単位で管理するのかを決めてください。また、ジョブネットワークグループの階層は3～4階層に留めておくことを推奨します。

3.2.3. ジョブネットワークの名前について

ジョブネットワークの名前は、1ユーザの中で完全にユニークである必要があります。

1ユーザ内の場合、別グループでも同じ名前は使用できません（別ユーザであれば、同じ名前を使用できます）。そのため、設計時に命名規則を決めておくことを推奨します。

命名規則を作成する際に、考慮に入れておくべき点を次に示します。

■名前の長さは、最大40バイト

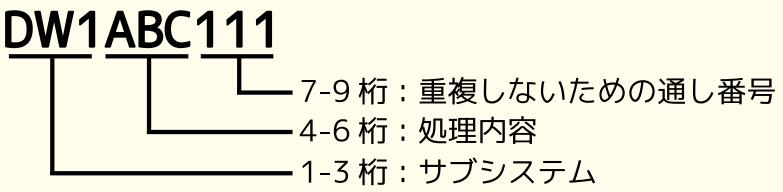
■「ジョブネットワーク名：サブジョブネットワーク名：サブジョブネットワーク名：…」の長さは、最大80バイト

そのため、サブジョブネットワークを使用する場合、1ジョブネットワークの名称の長さ、ジョブネットワークの最大階層数を決める必要があります。例えば、1ジョブネットワーク名は最大25バイト、サブジョブネットワークの階層は最大3階層までなどです。

命名規則のポイント

ジョブネットワーク名を決める場合、処理内容をそのままジョブネットワーク名にすると内容がわかり便利な面がありますが、名前が長くなる傾向があります。この場合、名前の長さに制限があるので、深い階層を用いたジョブネットワークにするのが難しくなります。このような場合に対応する1つの例として、ジョブネットワーク名はあるルールで採番した英数字のみの名称とし、処理内容はジョブネットワークのコメントに記述するという方法があります。

【例】

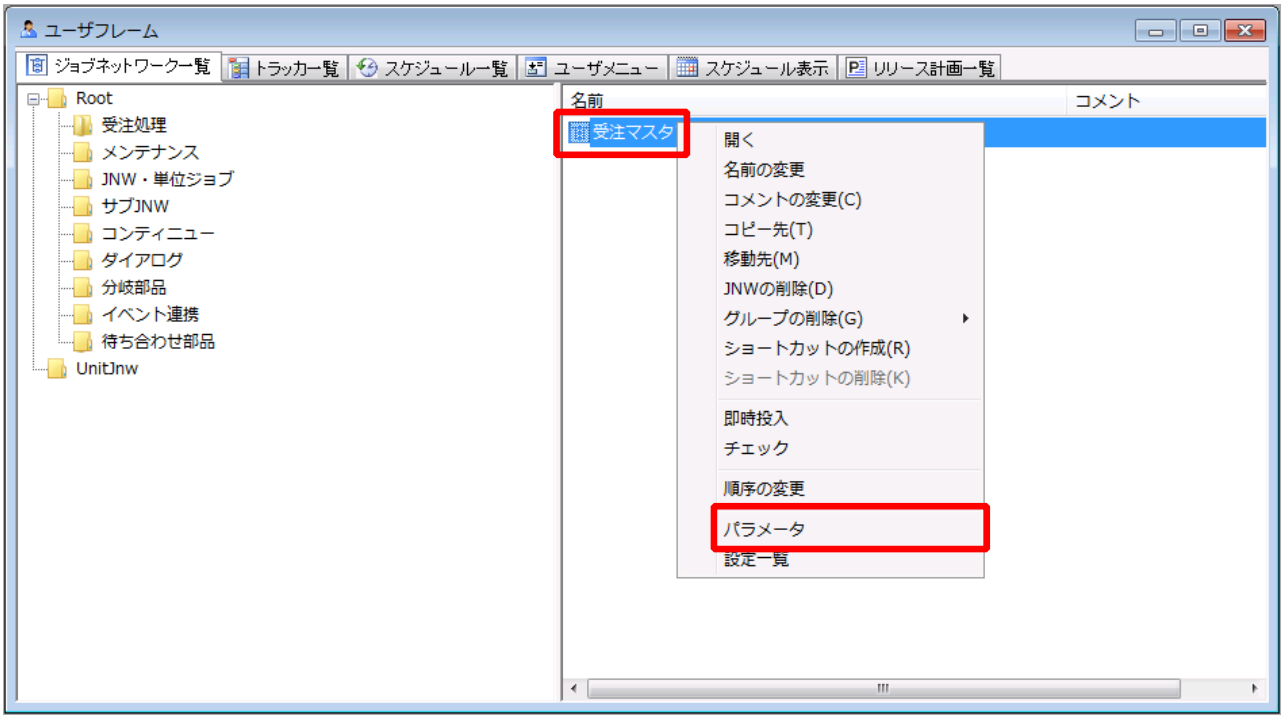


3.3. ジョブネットワークの作成

ここでは、様々な要件や状況に合わせたワークフローの作成方法を紹介します。

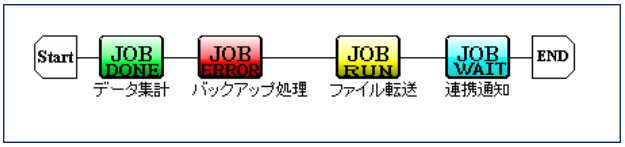
3.3.1. ジョブネットワークのパラメータを表示する

これ以降の説明で共通の手順となるジョブネットワークのパラメータを表示する方法は、パラメータを設定するジョブネットワークを右クリックし、表示されたメニューから[パラメータ]を選択します。



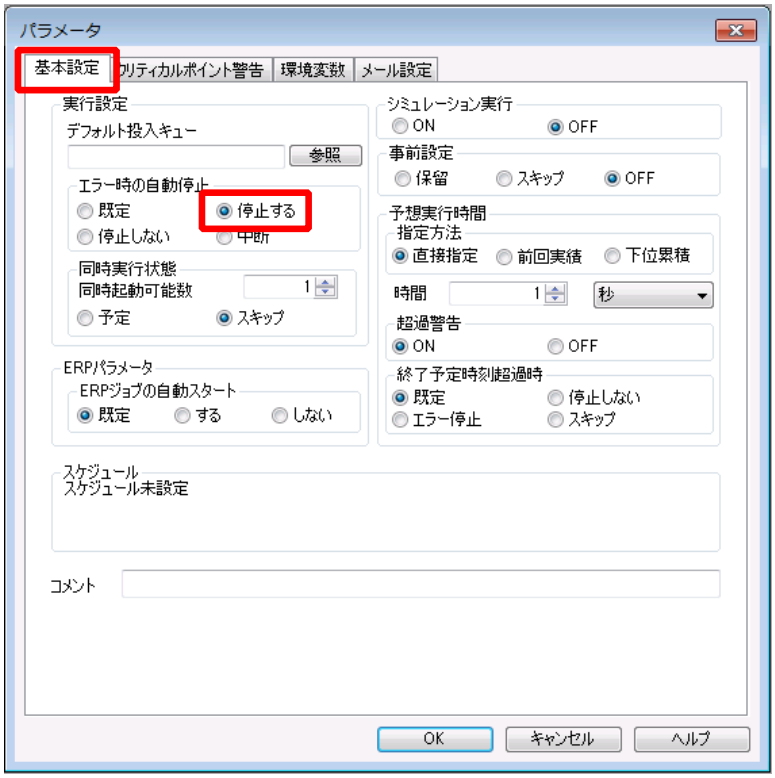
3.3.2. ジョブが異常終了したとき、フローを停止したい

デフォルトの設定では、ジョブが異常終了してもフローは止まらず、先に進みます。



ジョブが異常終了したときに、フローを停止するには次の設定を行ってください。ジョブが異常終了すると、フローの進行を停止できます（エラー停止状態）。

この場合、設定を行うジョブネットワークのパラメータを表示して、[基本設定]タブの[エラー時の自動停止]で[停止する]を選択します。



エラー停止と中断の違いについて

ここでは、［パラメータ］ダイアログの[エラー時の自動停止]で[停止する]を選択した場合と[中断]を選択した場合の違いを説明します。

[停止する]を選択した場合、トラッカは[エラー停止]となります。フローの進行が途中でSTOPした状態で、後続の部品はすべて未実行（WAIT）状態になります。そのため、フローはENDに到達せず、トラッカのアーカイブ処理が行われません。

実行完了していると見なされないため、
アーカイブされない！

[中断]を選択した場合、トラッカは[中断]となります。異常終了したジョブより後続の部品はすべてスキップされ、フローはENDに到達します。そのため、一定時間後にトラッカのアーカイブ処理が行われますフローはENDに到達します。

実行完了状態なので、
10 分後にアーカイブされる

選択項目	トラッカの表示	後続の部品の処理	トラッカの アーカイブ処理
［停止する］	［エラー停止］	すべて未実行（WAIT）状態（フローの進行が途中でストップした状態のため、フローはENDに到達しない）	行われない
［中断］	［中断］	すべてSKIPし、フローはENDに到達する	アーカイブされる

エラー停止と中断の使い分け方

エラー停止は、異常終了したジョブの対処を行い、対象トラッカを再実行させる場合に使用します。「エラー停止」トラッカはアーカイブされずに残り続けるため、再実行などの操作を受け付けてくれるためです。ただし、このようなトラッカを放置すると、エラー停止状態のトラッカはいつまでも残り続けるため、ディスクやメモリを消費したままとなるので、ご注意ください。

中断は、後続ジョブの処理は行いたくないが、トラッカは実行完了させたい（エラー停止として残したくない）場合に使用します。中断されたトラッカは10分後にはアーカイブされるため、エラー停止のようにメモリやディスクも消費しません。

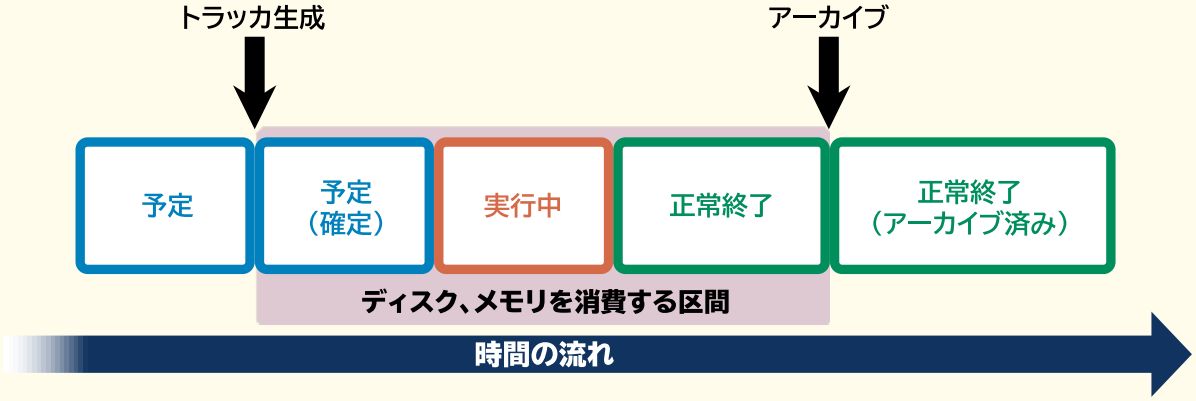
トラッカについて

トラッカはジョブネットワークを実行した結果であり、ジョブネットワークを実行する度に生成されます。基本的には、即時投入やスケジュールに登録（予定トラッカ）された予定開始時刻になったときにジョブネットワークが実行開始されるタイミングで生成されます。
ただし、予定トラッカの場合、何かの操作（「保留」など）を行うと、そのトラッカは予定（確定）状態となり、その操作を行った時点でトラッカが生成されます。

予定トラッカは、操作の情報を格納した領域を確保したものであり、実体としてのトラッカ情報は持っていません。

トラッカが生成されると、対象トラッカの各種データを格納するための領域（トラッカディレクトリ）が生成され、フロー制御プロセスのメモリにロードされるため、ディスクとメモリが消費されます。

トラッカの実行完了後、10分[注]経過するとJobCenterはアーカイブ処理を行い、トラッカディレクトリを削除し、アーカイブファイルにまとめ、メモリを解放します。



注　時間は設定で変更可能です。

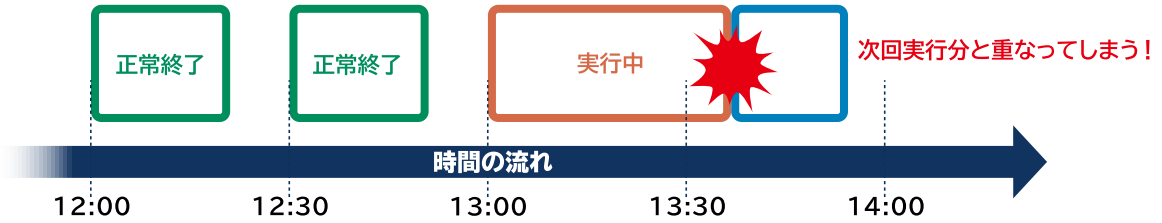
■トラッカがアーカイブされるとメモリから解放されるため、対象トラッカのスキップや保留といった操作はできなくなります。

■トラッカのアーカイブ時、トラッカはアーカイブファイルにまとめられるだけで、削除されないため、ディスク消費量は減りますが、まったくディスクを消費されるわけではありません。

3.3.3. 同じジョブネットワークの実行が重ならないようにしたい

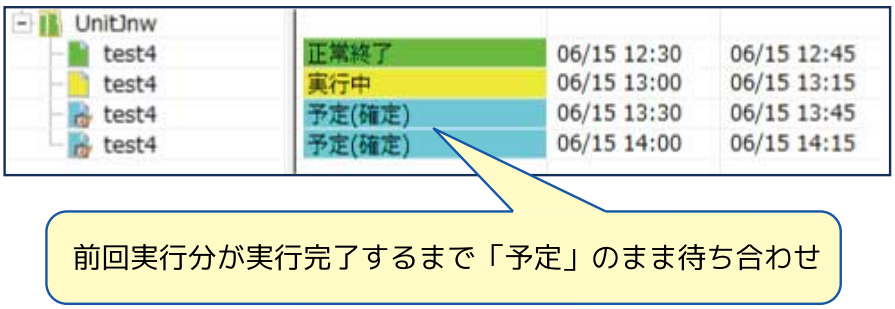
ジョブネットワークを重複して実行させたくない場合、次回実行分を開始しないように抑止する必要があります。JobCenterではジョブネットワークのパラメータの[同時実行状態]タブの「JNW単独の排他」の設定を行うことで同時に実行状態となるトラッカ数を制限できるため、対応できます。

例えば、30分間隔のような比較的短い実行間隔でフローを実行させている場合、通常であれば15分で実行完了するので特に問題は起きないが、処理量が多かったり、マシン負荷が高かったりして、通常より時間がかかり、30分を超過した場合、次回実行分と重なります。

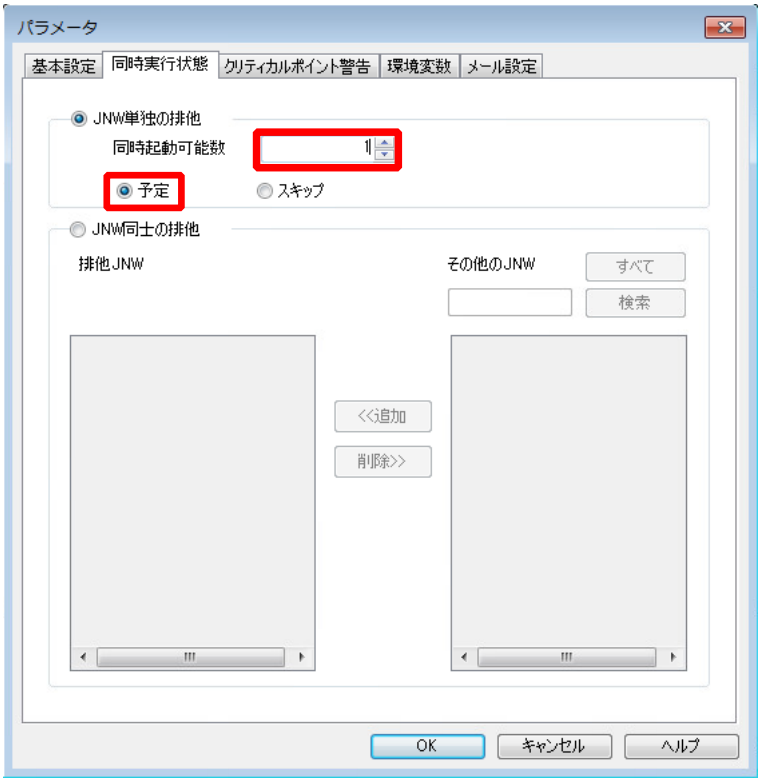


■次回実行分を、前の分が終わるまで待たせる場合

「JNW単独の排他」のパラメータの[同時起動可能数]を「1」、制限超過した場合の動作を「予定」にします。

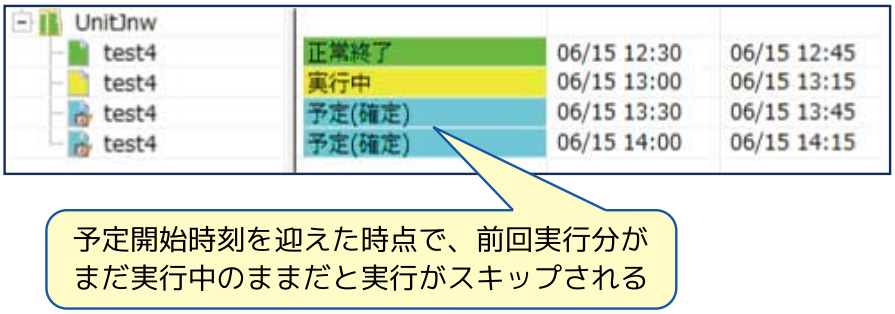


前回実行分が実行完了するまで「予定」のまま待ち合わせ

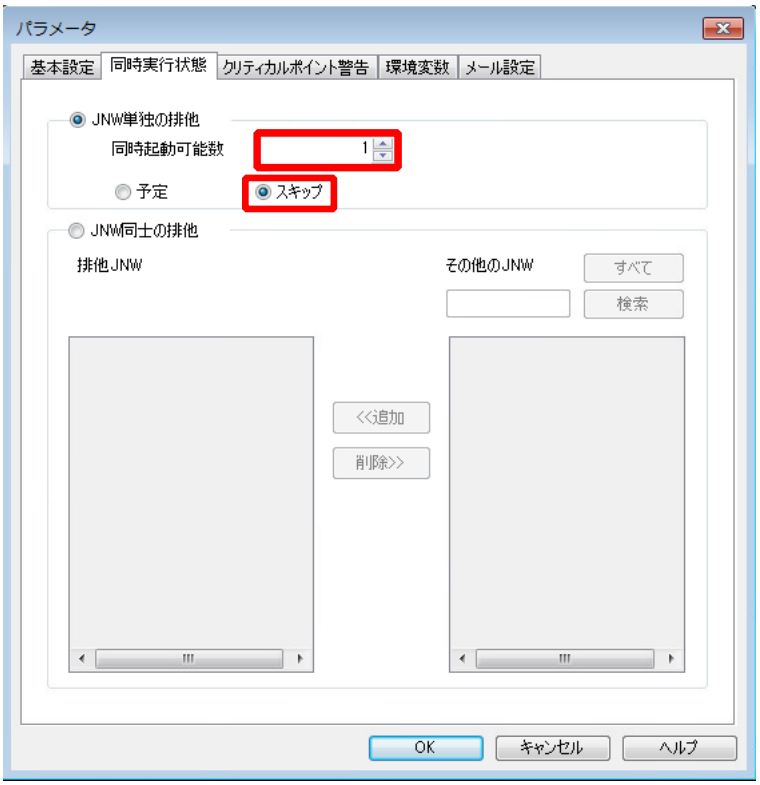


■次回実行分をまったく実行させずスキップさせる場合

「JNW単独の排他」のパラメータの[同時起動可能数]を「1」、制限超過した場合の動作を「スキップ」にします。



予定開始時刻を迎えた時点で、前回実行分がまだ実行中のままだと実行がスキップされる



エラー停止トラッカは実行中と同じ扱いになるため、同時起動可能数のカウント対象となりますので、ご注意ください。

例えば、[同時起動可能数]を「1」に制限した場合、該当ジョブネットワークのトラッカがエラー停止している場合、それ以降にスケジューリングされた同じジョブネットワークのトラッカは実行開始されません。

3.3.4. 他のジョブネットワークと実行が重ならないようにしたい

他のジョブネットワークと重なって実行させたくない場合、次回実行分を開始しないように抑止する必要があります。JobCenterではジョブネットワークのパラメータの[同時実行状態]タブの「JNW同士の排他」の設定を行うことで同時に実行状態となるトラッカを制限できるため、対応できます。

■NewJnw1とNewJnw2のトラッカの実行が重ならないようにする場合

排他JNWを、以下のように設定します。

- NewJnw1の排他JNW：NewJnw2
- NewJnw2の排他JNW：NewJnw1

■NewJnw1、NewJnw2、NewJnw3の中で１つのトラッカしか実行できないようにする場合

排他JNWを、以下のように設定します。

- NewJnw1の排他JNW：NewJnw1, NewJnw2, NewJnw3
- NewJnw2の排他JNW：NewJnw1, NewJnw2, NewJnw3
- NewJnw3の排他JNW：NewJnw1, NewJnw2, NewJnw3

ジョブネットワークのパラメータの[同時実行状態]タブの「JNW同士の排他」の設定については、次のマニュアルの[排他JNW]パラメータをご覧ください。

参照	<クラシックモード用基本操作ガイド> 「3.3.4.2 同時実行状態」

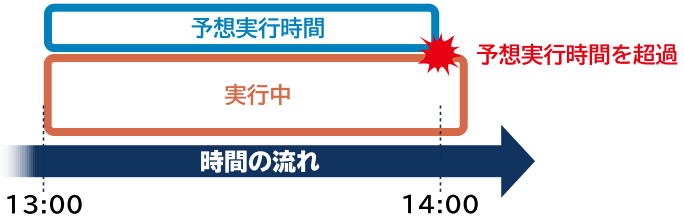
3.3.5. 時間超過したものを自動検知したい（または打ち切ってしまいたい）

ジョブネットワークが一定時間経過しても実行完了しないものがあり、それを素早く検知して対処を検討したい場合、JobCenterでは「超過警告」を利用する方法と「クリティカルポイント警告」を利用する方法があります。

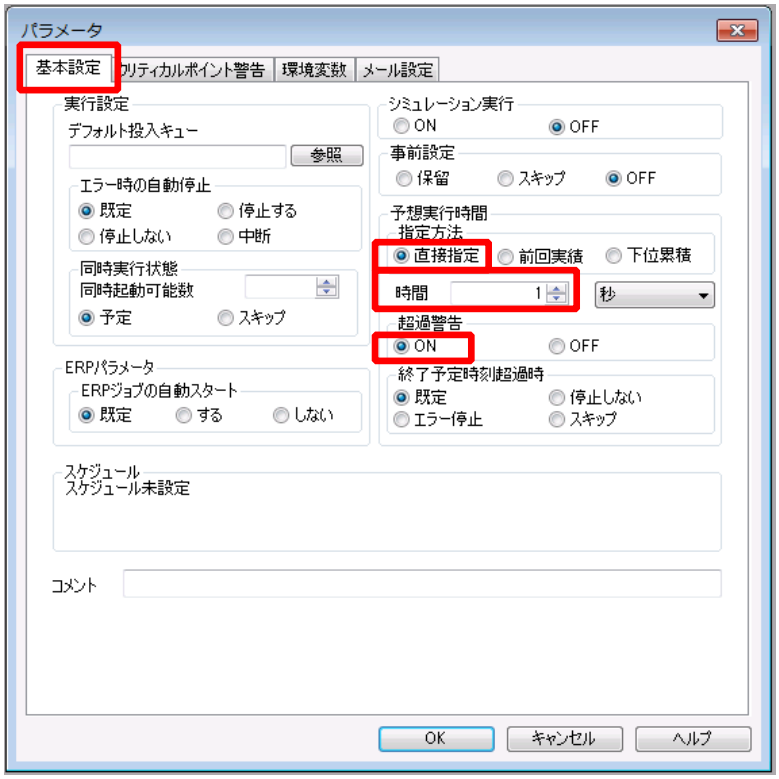
「超過警告」は、予想実行時間でチェックを行い、警告を出します。実行開始が遅れていても、実行が開始されてからの時間でチェックすることができます。「クリティカルポイント警告」は時刻でチェックを行い、警告を出します。実行開始時刻にかかわらず、その時刻（例えば、15:00時点など）で実行終了していなければ警告を出します。

3.3.5.1. 超過警告を使用する場合

超過警告は、あらかじめ指定した予想実行時間を超えても実行完了しない場合に警告を出します。



例えば、予想実行時間を1時間、実行開始時刻が12:00に設定し、時刻通り12:00に実行開始されたが、1時間後の13:00の時点で実行終了していないと警告を出す設定を行うには、ジョブネットワークのパラメータを表示し、[基本設定]タブで、[予想実行時間]の[指定方法]で[直接指定]を、[時間]を「1」に、[超過警告]の[ON]を選択します。



警告を出すだけでなく、実行を停止する場合、[終了予定時刻超過時]のパラメータを選択します。例えば、「エラー停止」を選択した場合、実行中のジョブは強制停止になり、フローはエラー停止状態になります。

■あらかじめ予想実行時間を決めておく必要があります。

■[予想実行時間]の[指定方法]に[前回実績]を選択しないでください。[前回実績]にすると、1回前の正常終了したときの実績値がそのまま予想実行時間として採用されます。この場合、毎回値が変わるため、頻繁に警告が出力されてしまいます。基本的には[直接指定]を選択して、ある程度の余裕を持たせた時間を入力することを推奨します。

■「下位累積」の場合、フロー中に配置した各部品の予想実行時間の合計値となるため、予想実行時間を[直接指定]を選択してください。

3.3.5.2. クリティカルポイント警告を使用する場合

クリティカルポイント警告は、あらかじめ指定した時刻を過ぎた場合、警告を出します。警告を出す判断基準として、開始点と終了点を設定できます。

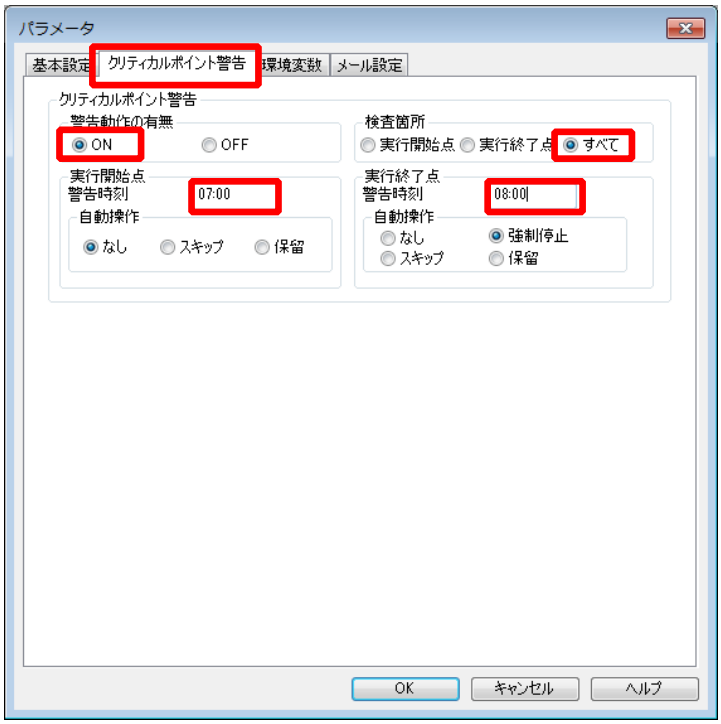
開始点は、指定した時刻になっても実行開始していない場合に警告を出します。同時起動可能数の制限で待ち合わせされられていたり、対象トラッカに保留をかけて放置していたりした場合に利用できます。

終了点は、指定した時刻になっても実行終了していない場合に警告を出します。通常、終了点を使用します。

設定を行うジョブネットワークのパラメータの[クリティカルポイント警告]を表示します。次に[クリティカルポイント警告]の[警告動作の有無]、[検査箇所]の[すべて]を選択し、[実行開始点]、[実行終了点]の時刻を指定します。

例えば、次のような場合、「クリティカルポイント警告」を利用すると有効です。

■夜間に日次バックアップを行う場合、朝8時までに終了しなければ、通常業務に影響が出るので、指定した時刻になっても終わらないときは強制的にエラー終了として中止したい。当日できなかったバックアップは、翌日の日次バックアップでリカバリする。



警告を出すだけでなく、実行を停止する場合、[自動操作]のパラメータを選択します。例えば、実行終了点の検査で「強制停止」を選択した場合、実行中のジョブは強制停止になり、フローはエラー停止状態になります。

3.3.5.3. 超過警告とクリティカルポイント警告の使い分けについて

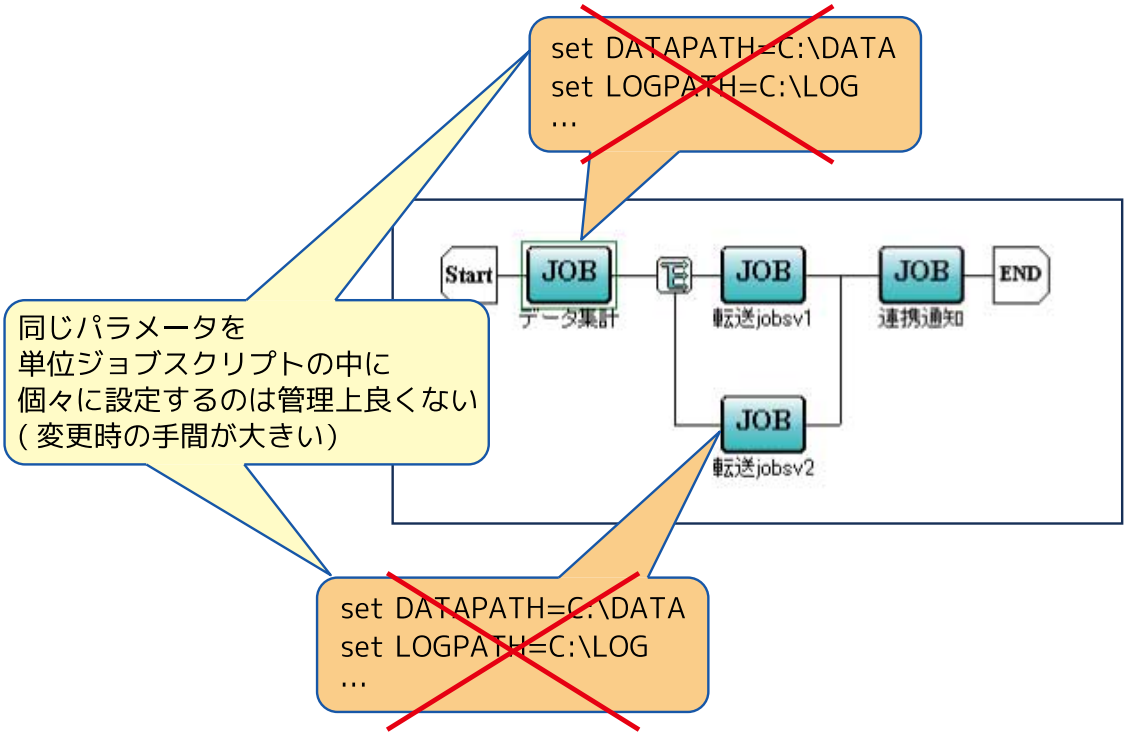
超過警告は予想実行時間、すなわち時間でチェックを行い警告を出します。実行開始が遅れていても、実行が開始されてからの時間でチェックできます。

クリティカルポイント警告は時刻でチェックを行い警告を出します。実行開始時間にかかわらず、指定した時刻、例えば15:00時点で終わっていなければ警告を出すことができます。

超過警告、クリティカルポイント警告は、ジョブネットワークまたは単位ジョブのパラメータとして設定できます。ここで説明したのはジョブネットワークの例ですが、単位ジョブのパラメータとして上記の手順で設定できます。違いは制御の単位で、単位ジョブに限定されます。

3.3.6. フロー内に配置した各単位ジョブに共通の環境変数を定義したい

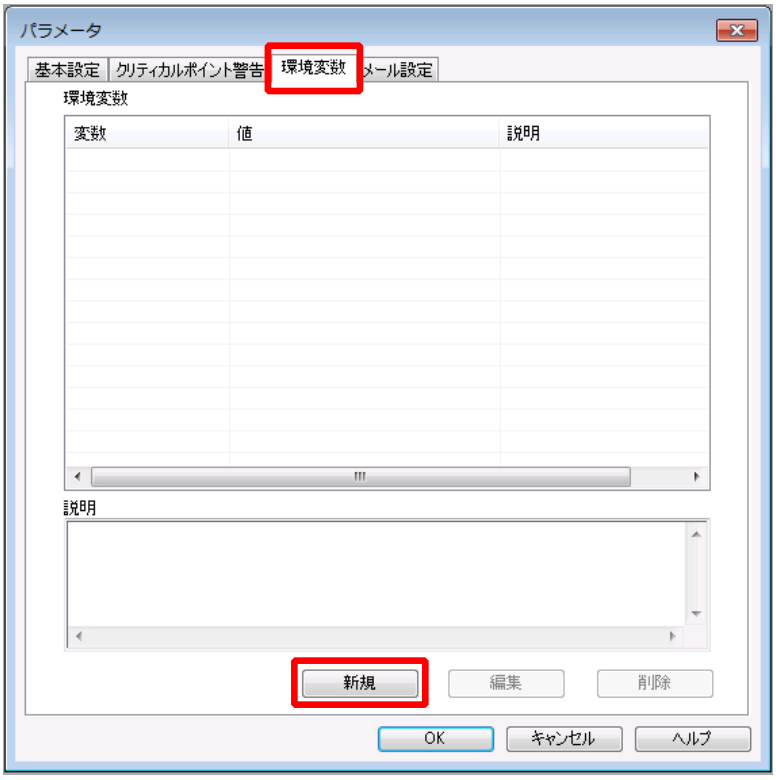
JobCenterでは、データの格納先やログの出力など、フロー内の各単位ジョブで共通のパラメータを設定できるため、各単位ジョブスクリプトに個々に設定する手間や管理上のコストを省けます。



ジョブネットワークパラメータの[環境変数]を設定することで、そのフローに配置されているすべての単位ジョブ部品の共通の環境変数として定義できます。

この設定は、ジョブネットワークのパラメータの[環境変数]タブで行います。

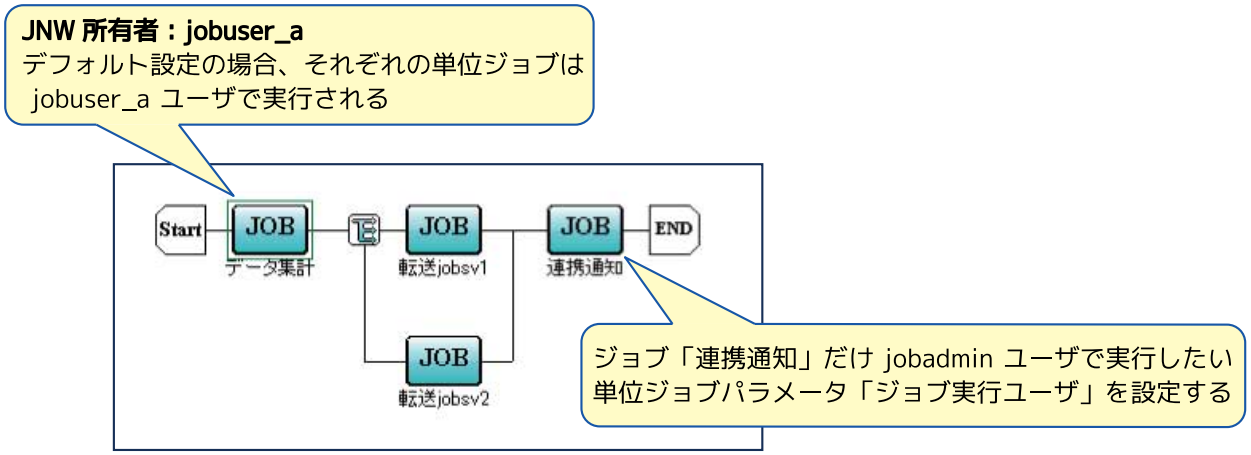
ジョブネットワークのパラメータの[環境変数]タブの[新規]をクリックして表示される画面で、変数、値、説明を設定します。

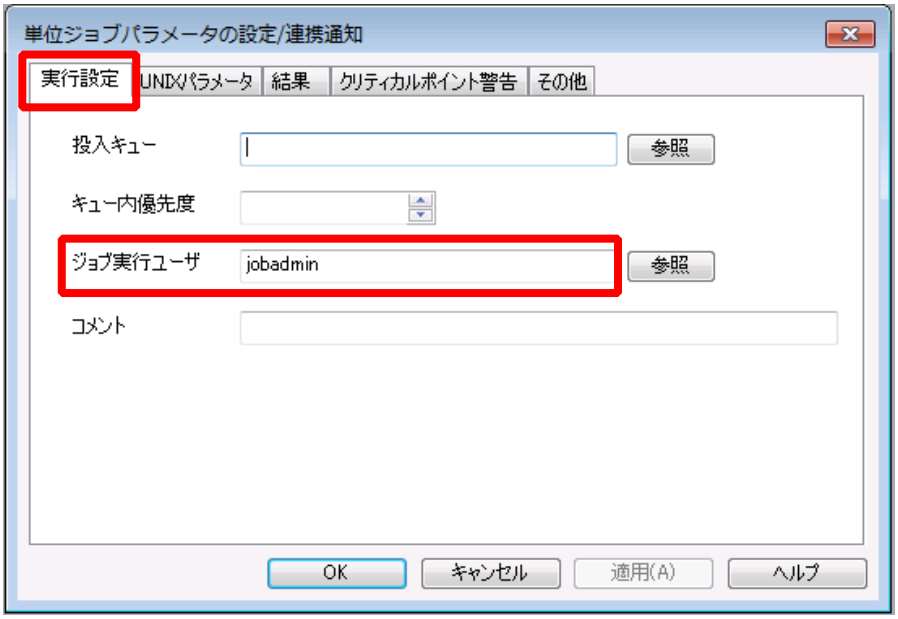


3.3.7. ジョブの実行ユーザを切り替えたい

ジョブネットワーク内に配置した単位ジョブは、デフォルトではそのジョブネットワークの所有者の権限で実行されます。これを変更したい場合は、単位ジョブパラメータの「ジョブ実行ユーザ」を変更してください。

ジョブネットワークで設定する単位ジョブ部品のパラメータを表示し、[実行設定]タブの[ジョブ実行ユーザ]で切り替えるユーザを設定します。





■「ジョブ実行ユーザ」を変更するには、そのジョブネットワークを編集するユーザが「他ユーザのジョブネットワークの編集/更新/削除」の権限を持っている必要があります。（所有者ではなく、編集する人が権限を持っているかどうかがポイントです）

■「ジョブ実行ユーザ」の選択肢として表示されるのは以下の通りです。

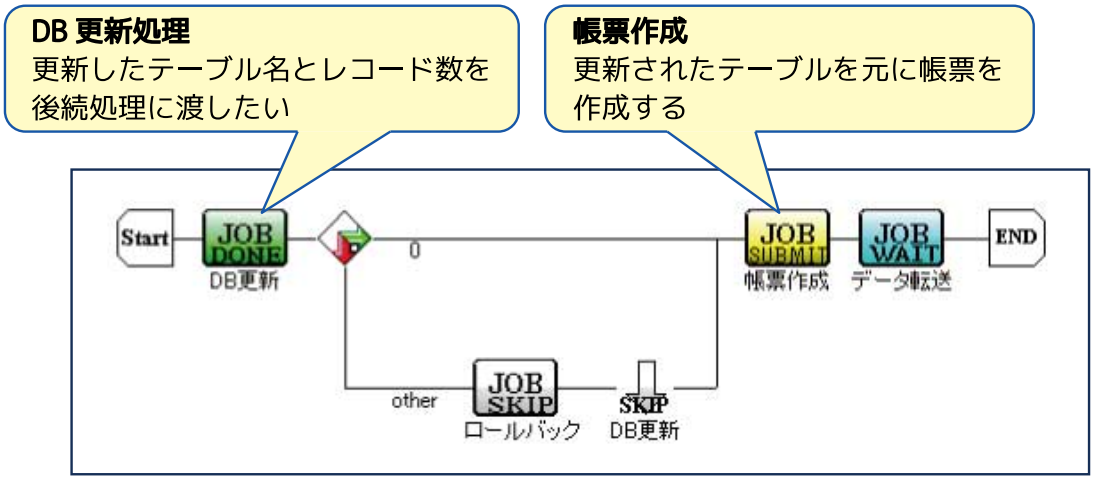
- Windows版：JobCenterグループに所属しているユーザ
- UNIX/Linux版：対象マシン（OS）に登録されている全ユーザ

■リモートマシンに転送して実行させる場合、ローカルマシン上のユーザとリモートマシン上のユーザの紐付け（ユーザマッピング）が必要です。

参照 「2.2.2 ユーザIDマッピング」

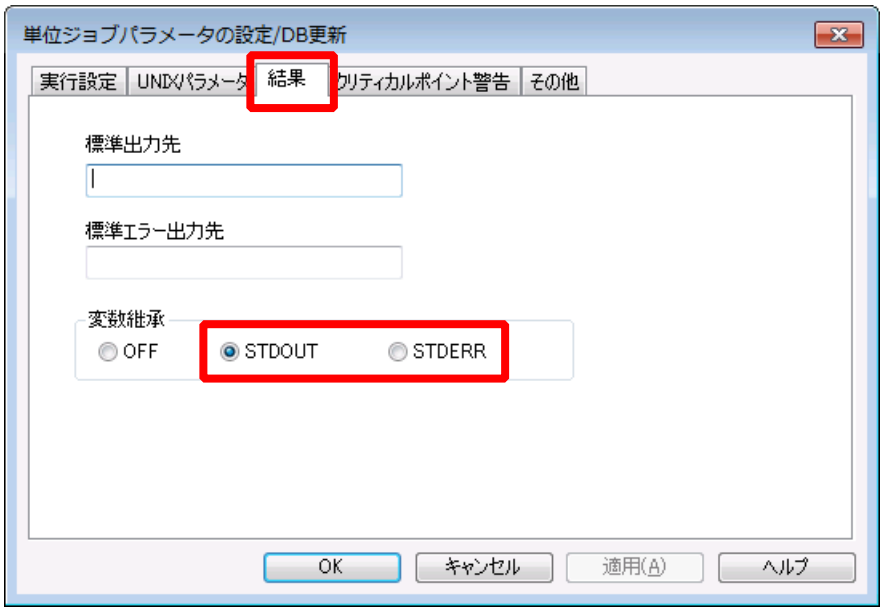
3.3.8. 単位ジョブの出力結果を後続の単位ジョブで参照したい

「ジョブの実行ユーザを切り替えたい」の単位ジョブで行った処理結果のアウトプットを、後続の処理の中で参照したい場合、単位ジョブ部品の「変数継承」機能を利用すると、環境変数として後続部品に継承できます。

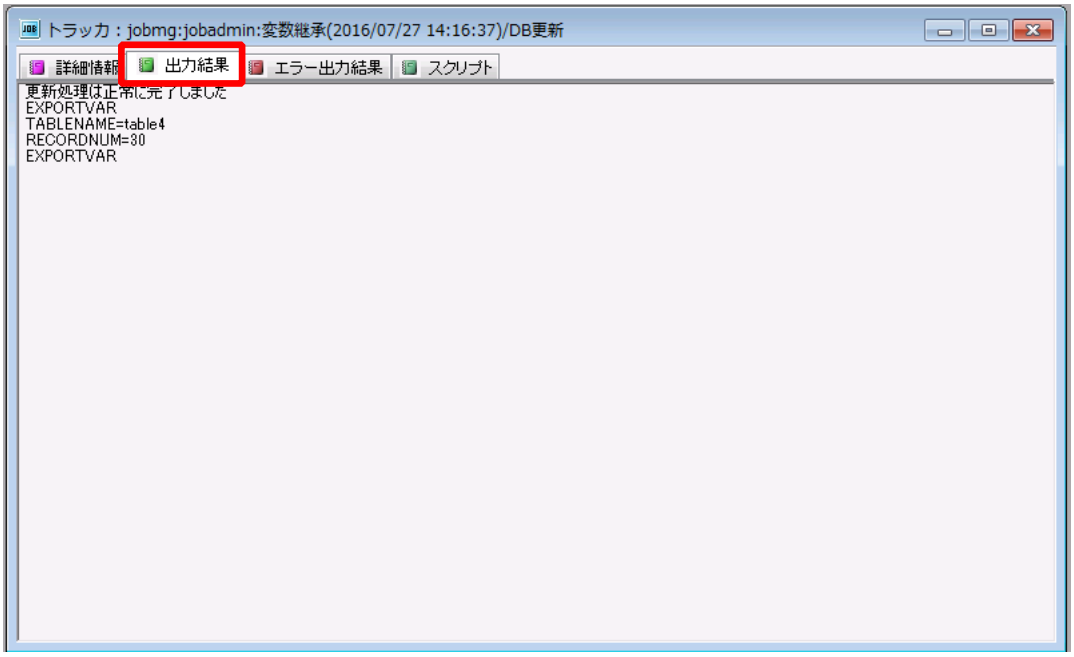


継承元単位ジョブ部品のパラメータの[結果]タブの[変数継承]で[STDOUT]（標準出力）または[STDERR]（標準エラー出力）を選択します。

確認する場合、ジョブネットワークを実行後、継承元単位ジョブ部品で詳細情報の[出力結果]タブを表示します。



「EXPORTVAR」という文字列で囲まれた部分が環境変数として継承され、後続の部品で参照可能になります。



■環境変数はトラッカ単位で管理されます（トラッカごとに独立した環境変数データセットを持ちます）。

■継承された環境変数は、そのトラッカ内の後続のどの単位ジョブでも参照できます。

■同じ環境変数を後続部品でさらに継承設定した場合、継承設定した内容で上書きされます。

例えば、仮想マシンでジョブを実行している場合、仮想マシンが物理的にどのホストマシンにあるのかをジョブで取得し、後続のジョブはホストマシン名を意識した処理を行わせたい場合があります。

このような場合、仮想マシンで先行して動作するジョブでホストOSを取得するスクリプトを実行します。後続のジョブには、JobCenterの環境変数にホストOSの情報を設定し、環境変数を引き継がせます。 これで情報を連携した処理が行えます。

参照	<クラシックモード用環境構築ガイド> 15章 「ジョブ実行時の環境変数の取り扱い」

3.3.9. ジョブの正常終了・異常終了の条件を変更したい

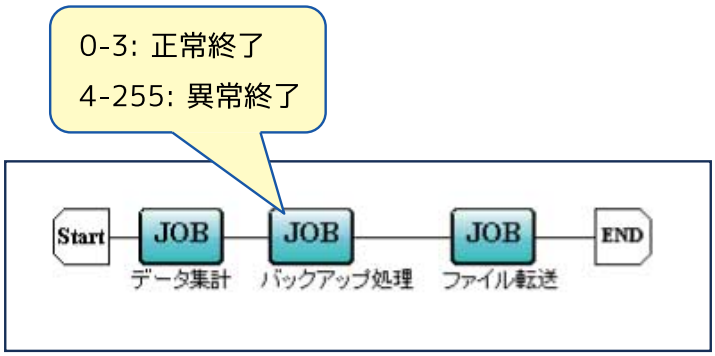
デフォルトの単位ジョブの正常終了、異常終了の条件は次のようになっています。

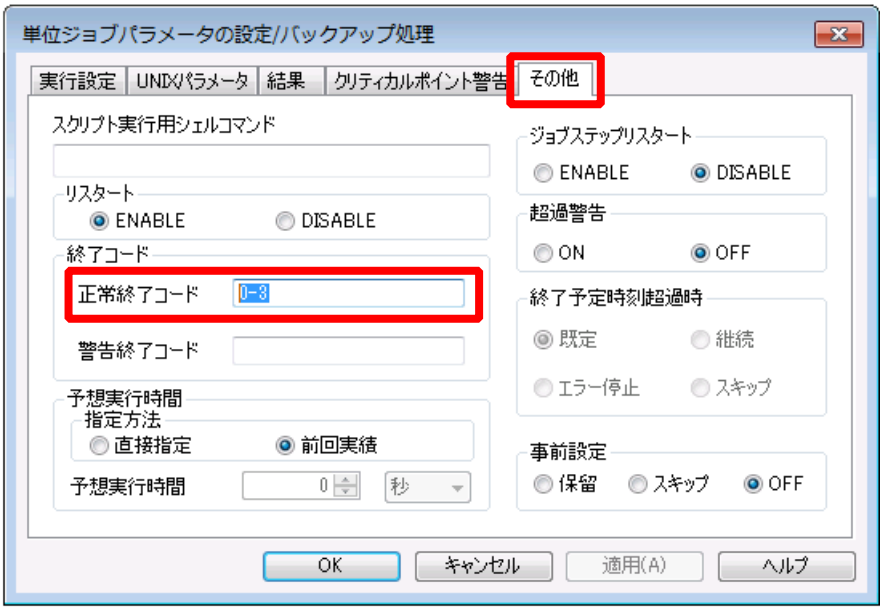
■正常終了：終了コードが0

■異常終了：終了コードが0以外

JobCenterでは、正常終了・異常終了の条件を変更できます。

例えば、正常終了を「0-3」、それ以外は異常終了とする場合、単位ジョブのパラメータの[その他]タブの[終了コード]の[正常終了コード]に「0-3」を入力します。





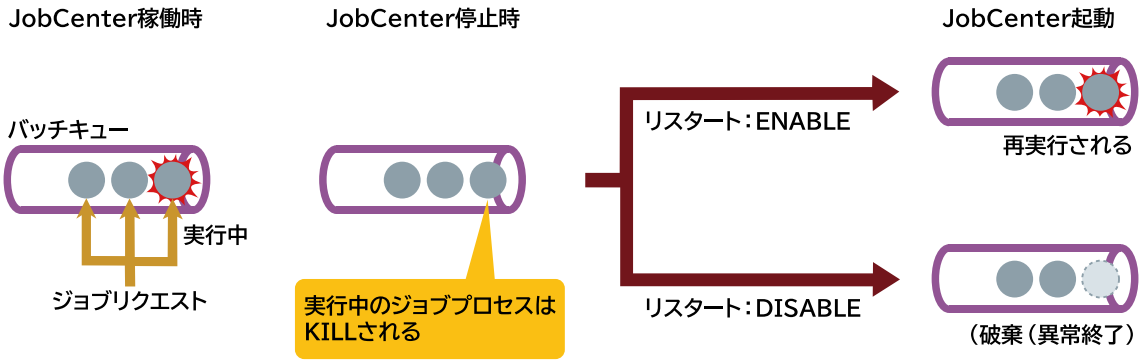
例えば、次のような利用方法があります。

■ジョブのスクリプトのエラーコードによって、ある値未満であれば正常終了（対処不要）とし、ある値以上であればエラーで対処が必要とすることで、開発者への通知を自動的に振り分けたい。また、オペレータが目視、手動でエラーコードを判別しなくても、自動で開発者への通知したい。

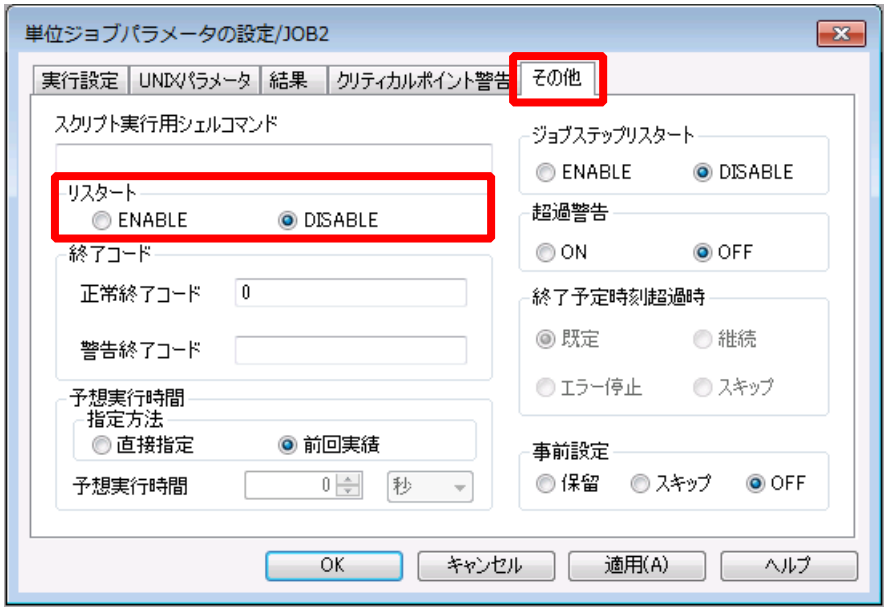
3.3.10. ジョブ実行中にJobCenterを停止した場合の制御について

ジョブ実行中にJobCenterを停止した場合、ジョブは停止します。

再起動した場合、停止したジョブを自動再実行するかどうかを、単位ジョブのパラメータで設定できます（デフォルトでは自動再実行をする設定になっています）。



設定は、単位ジョブのパラメータの[その他]タブの[リスタート]で[ENABLE]または[DISABLE]を選択します。JobCenter再起動時、ジョブを自動再実行したい場合、[リスタート]の[ENABLE]を選択してください。再実行しない場合は、[DISABLE]選択してください。[DISABLE]を選択した場合、停止したジョブは破棄され、異常終了扱いになります。



自動再実行されるかどうかはキューの再起動属性にも依存します。

参照 「4.2.2 JobCenter起動時の制御」

3.3.11. ジョブネットワークから他のジョブネットワークを実行したい

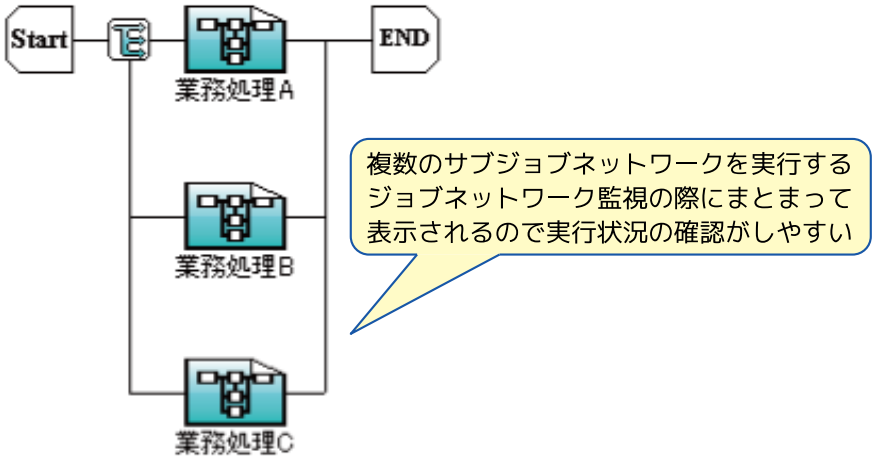
あるジョブネットワークから他のジョブネットワークを実行したい場合、「サブジョブネットワーク部品」を利用すると、あるジョブネットワークを他のジョブネットワークの部品として配置し、実行できます。

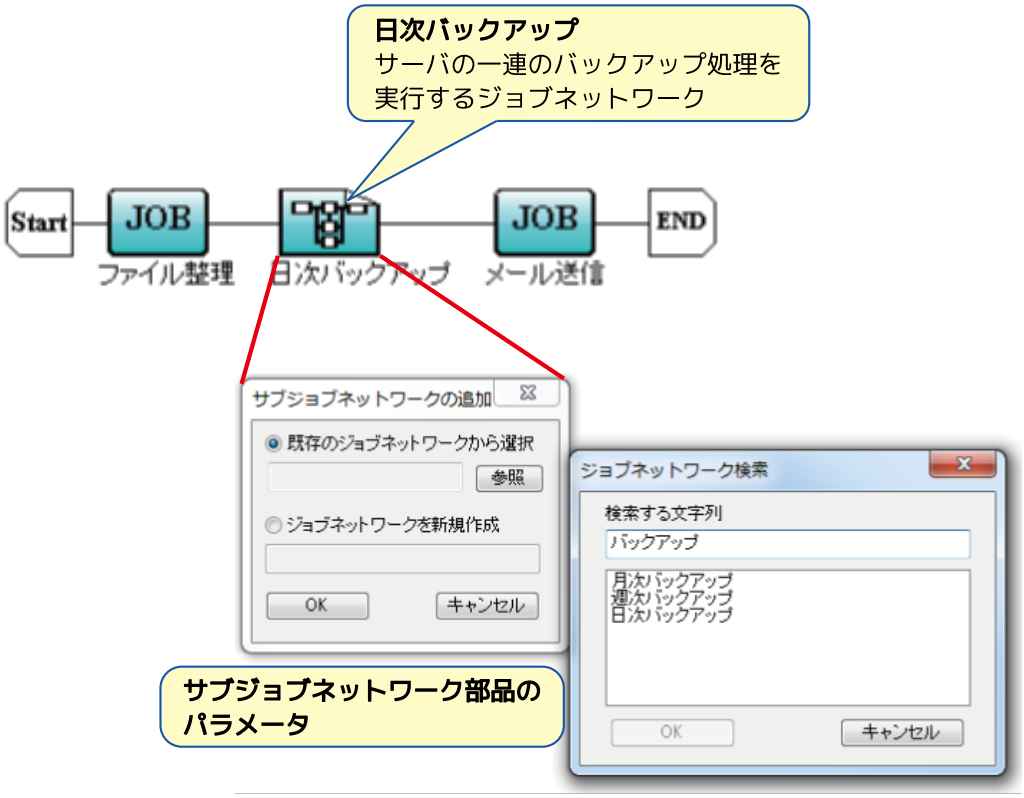
「サブジョブネットワーク部品」で次のような利用法があります。

■ある一連の処理を実行するジョブネットワークを、他のジョブネットワークの1つの部品として呼び出したい

■複数のジョブネットワークの実行を管理したい

フローに「サブジョブネットワーク部品」を配置すると、[サブジョブネットワークの追加]が表示されます。このダイアログの[既存のジョブネットワークから選択]を選択し、[参照]をクリックすると、[ジョブネットワークの検索]が表示されます。ここで追加するサブジョブネットワークを設定します。





サブジョブネットワーク利用時の注意

サブジョブネットワークとして利用する際にはいくつかの注意事項があるため、次のマニュアルをご覧ください。

参照	<クラシックモード用基本操作ガイド> 「4.2.9.3 サブジョブネットワークオブジェクトに関する注意事項」

特にオブジェクト名（xxxx:xxxx:xxxx）の文字数制限が厳しいため、あまり長い階層は指定できません。最大10階層使用できますが、実質的に使える階層数はもっと小さくなります。そのため、システムにおける次の点についてルールを決めておくことをお勧めします。

■ジョブネットワーク名の命名規則

■サブジョブネットワークの最大階層数

メンテナンスや管理コストを考慮すると、サブジョブネットワークの最大階層数は3階層程度（多くても4階層程度）にしておくことが望ましいです。

3.3.12. ユーザからの応答を待ち合わせてから、後続の処理を実行したい

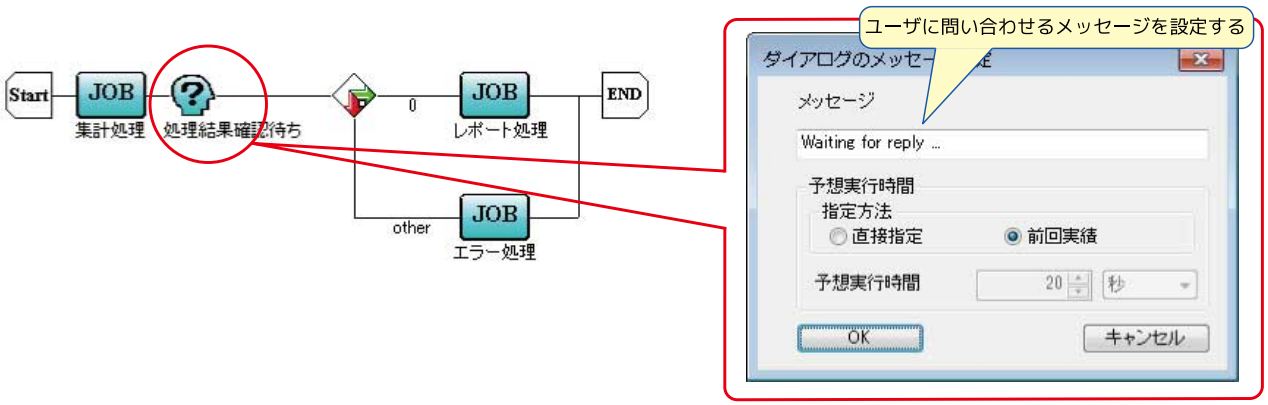
ユーザからの応答を待ち合わせてから後続の処理を実行したい場合は、「ダイアログ部品」を使用します。

フローに「ダイアログ部品」を配置すると[ダイアログのメッセージ]が表示されます。このダイアログの[メッセージ]にユーザに問い合わせるメッセージを設定します。

「ダイアログ部品」は次のような利用法があります。

■処理の途中で、ユーザが目視で処理の結果を確認してから、後続の処理を実行したい

■処理の途中で、ユーザが画面で手動の業務処理を行う必要があり、処理を止めたい



「ダイアログ部品」への応答は「Ok」と「Error」の2通りの応答が可能です。

例えば、次のような場合に「ダイアログ部品」を使うと有効です。

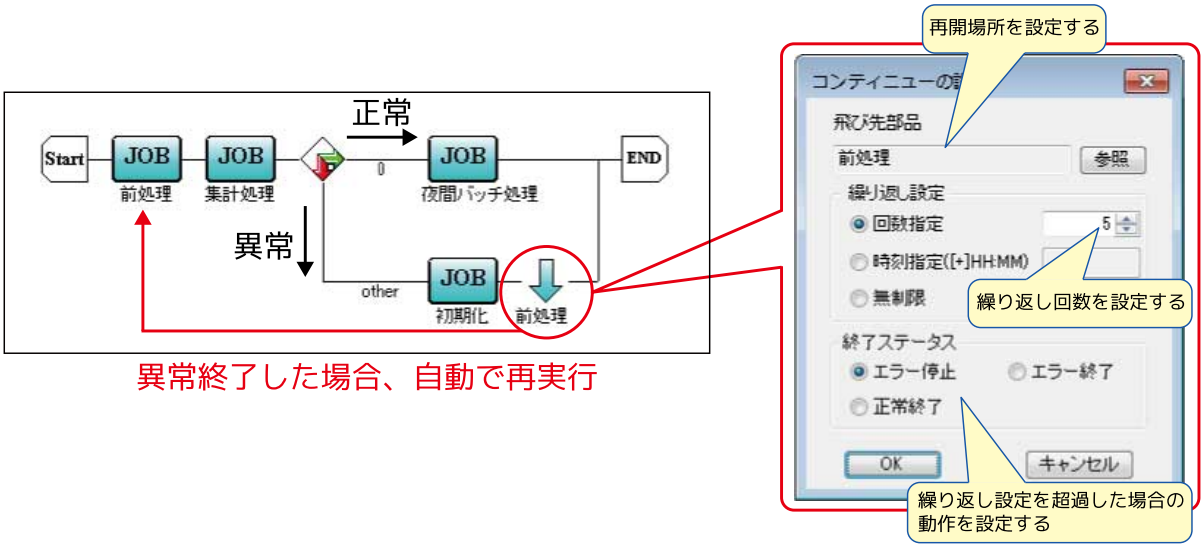
■ジョブネットワークの実行途中で人間による目視の確認を行い、その結果で上司の承認を得た（「Ok」）上で、ジョブを進めたい。上司の承認が下りなかった場合（「Error」）、そのジョブのみスキップし、次に進めたい。

次のジョブネットワークで「集計処理」の後に、ダイアログ部品「処理結果確認待ち」を配置する手順を説明します。

3.3.13. ジョブが異常終了した場合に自動でリトライしたい

ジョブが異常終了した場合に自動でリトライしたいといったような、同じ処理を繰り返し行いたい場合、「コンティニュー部品」を使用します。

フローに「コンティニュー部品」を配置すると、[コンティニューの設定]が表示されます。このダイアログで[飛び先部品]に再開する部品を、[回数指定]に繰り返す数を、[終了ステータス]に[繰り返し設定]を超過した場合の動作を設定します。



設定できる終了ステータスは次の通りです。

■[エラー停止]：繰り返し設定の設定値を超えるとコンティニュー部品はエラー停止となり、フローの実行が停止し、後続の部品は実行されません。

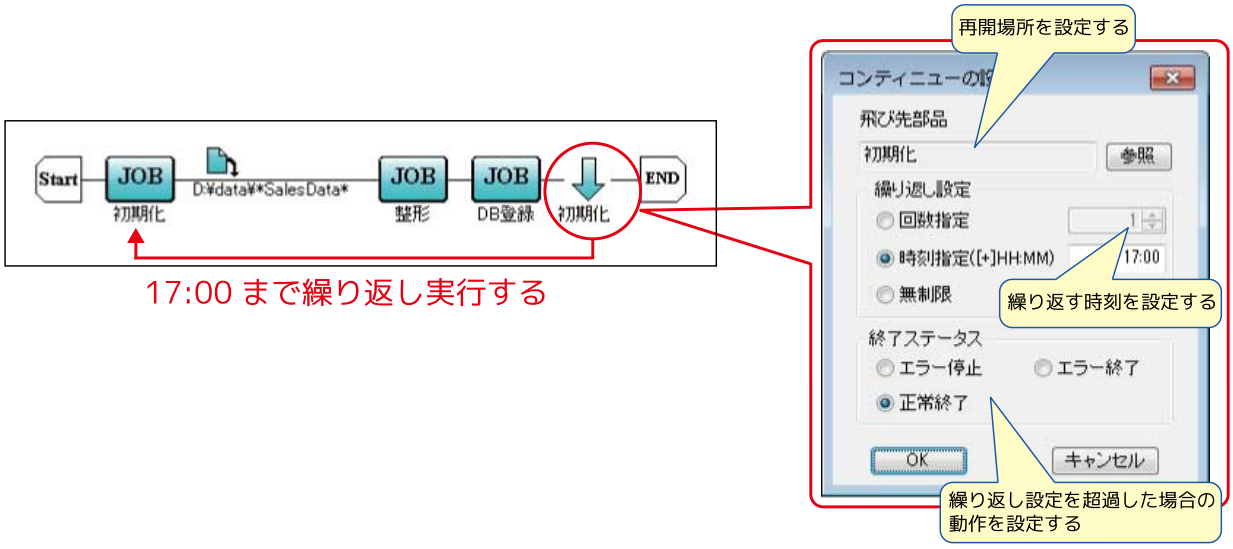
■[エラー終了]：繰り返し設定の設定値を超えるとコンティニュー部品はエラーとなり、後続の部品は実行されます。

■[正常終了]：繰り返し設定の設定値を超えるとコンティニュー部品は正常終了となり、後続の部品は実行されます。

3.3.14. 同じ処理を何度も繰り返し実行し、特定の時刻になったら処理を終了したい

同じ処理を何度も繰り返し実行し、特定の時刻になったら処理を終了したい場合、「コンティニュー部品」を利用します。

フローに「コンティニュー部品」を配置すると、[コンティニューの設定]が表示されます。このダイアログで[飛び先部品]に再開する部品を、[時刻指定]に繰り返す時刻を、[終了ステータス]に[繰り返し設定]を超過した場合の動作を設定します。



繰り返し設定の時刻指定について

■絶対時刻を指定した場合、ジョブネットワークの開始時刻を基準に指定した時刻まで繰り返します。

ジョブネットワークの開始時刻より前の時刻が指定されていた場合、翌日の指定時刻まで繰り返します。

■相対時刻を指定する場合は先頭に「+」を付けます。

例えば、1時間後に終了する場合は、「+01:00」と入力します。

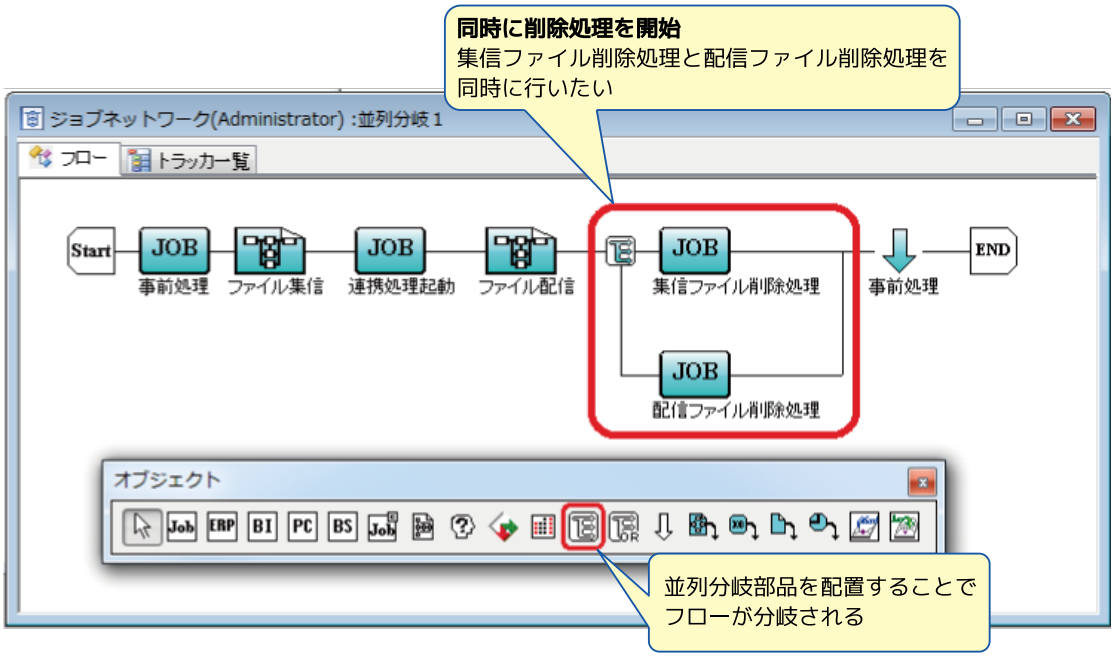
相対時刻の基準時刻はジョブネットワークの開始時刻になります。

3.3.15. 複数の処理を同時に行わせたい

複数のジョブ、またはサブジョブネットワークを同時に開始させ、すべてが終わるまで待ち合わせるようなフローを作りたい場合、「並列分岐部品」を利用して複数の処理を並列に配置します。

並列分岐部品をフローに配置することで複数のジョブ、サブジョブネットワークが同時に実行開始されます。並列分岐部品の次に配置された部品が実行されるのは分岐したすべての処理が完了した後になります。

次のジョブネットワークで並列分岐部品を配置し、単位ジョブ「集信ファイル削除処理」と単位ジョブ「配信ファイル削除処理」を並列で処理させる方法を説明します。



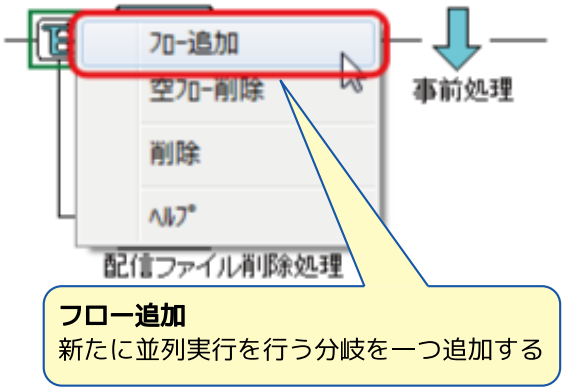
並列にした各フローで投入先のキューが同じだった場合、キューが詰まってしまうと後から来た処理はそのキューが空くまで待ってしまうので同時に実行が開始されない場合があります。投入するキューを分岐ごとに分けて実行するように設定した方が効果的です。

3.3.16. 並列分岐の数を変更したい・分岐そのものを消したい

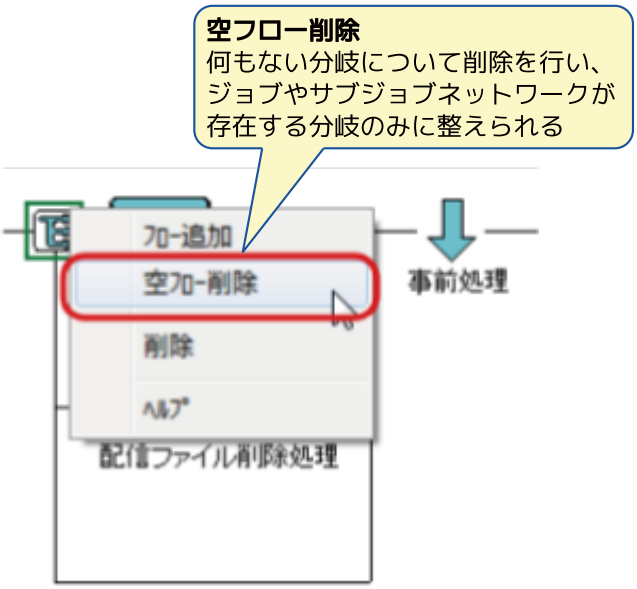
並列に行う処理を追加するため分岐を追加する、必要なくなった分岐を削除する、および分岐そのものを削除する方法を説明します。

「並列分岐部品」を右クリックして表示されるメニューより操作を選択します。

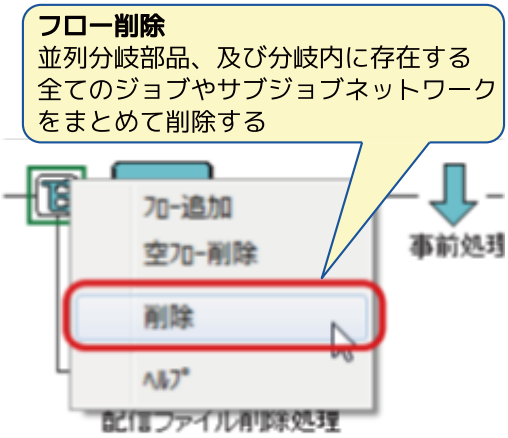
■フロー追加



■空フロー削除



■フロー削除



■なにも処理が存在しない分岐をそのままにしても、特に問題はありません。

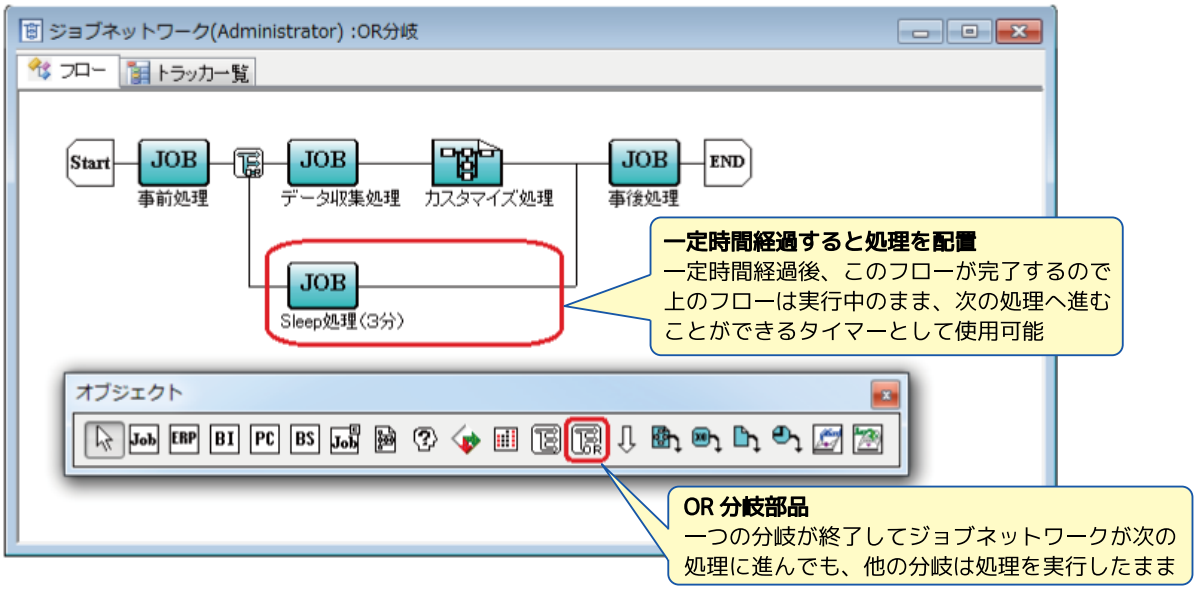
■並列分岐そのものを削除すると、その中のすべてのジョブやサブジョブネットワークが削除されます。

3.3.17. メインの処理を実行中に一定時間経過すると次の処理に進めたい

メインの処理を行っていて、一定時間経過した場合に次の処理に進むようなフローを作りたい場合、「OR分岐部品」を利用してメインの処理と一定時間待ったら進む部品を並列に配置します。

OR分岐部品をフローに配置することでいずれか1つのフローが完了すると次の処理に進むジョブネットワークを作ることができます。このOR分岐のフローの1つに一定時間経過すると進む単位ジョブを置くと、その一定時間経過後に次の処理へ進めることができるジョブネットワークを作ることができます。

例えば、次のジョブネットワークの場合で、上のフローは実行中のままで、下のフローは3分経過後完了し次の処理に進みます。そのため、下のフローはタイマーとして機能しています。



■OR分岐のフローの中に、待ち合わせ部品やイベント連携部品を配置することもできます。

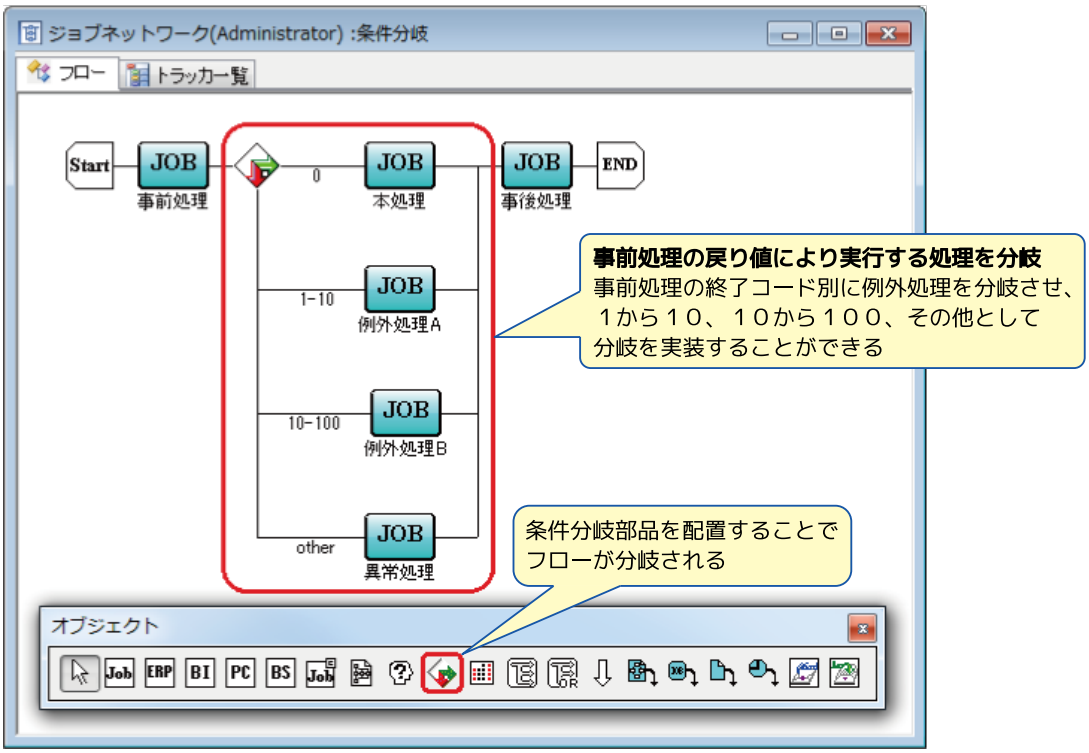
■OR分岐のフロー追加やフロー削除、OR分岐の削除の方法は並列分岐と同じように行うことができます。

3.3.18. ジョブ結果、またはサブジョブネットワーク結果により動作を変更したい

直前に実行した単位ジョブ、またはサブジョブネットワークの終了コードによって実行する処理の流れを変更させるようなフローを作りたい場合、「条件分岐部品」を利用して条件別に複数の処理を配置できます。

条件分岐部品をフローに配置することで直前の単位ジョブ、またはサブジョブネットワークの結果（終了コード）を複数のパターンで分けた処理を実装できます。

条件分岐部品をフローに配置し、フローを追加する場合は、配置した部品を右クリックし「フロー追加」をクリックします。分岐したフローにある[1]などの数字をダブルクリックすると、条件分岐設定が表示されるので、数値を入力し、条件を設定します。



分岐部品直前の部品動作について

単位ジョブ以外の部品は一部を除いて終了コードを持たないため、それらの部品の直後に条件分岐を配置すると、どこに分岐するのかが明確になりません。

次に、それぞれの部品を条件分岐の直前に配置した場合の分岐動作についてお知らせします。

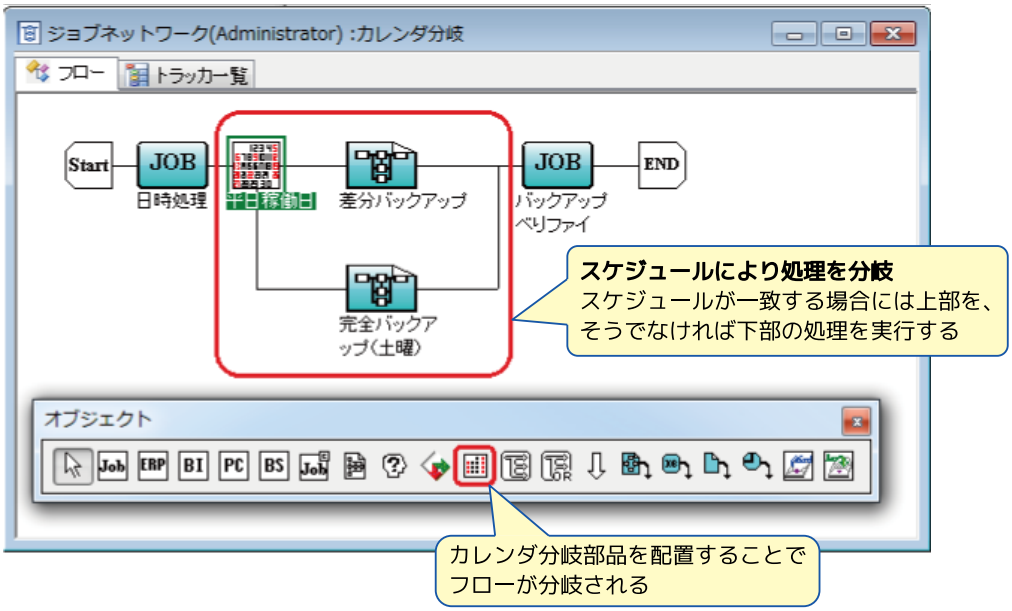
部品	終了コードの扱い				備考
	0	1	2 - 100	その他	
サブジョブネットワーク	サブジョブネットワークの結果がそのまま反映される				
ダイアログ部品	Ｏｋを実行	Errorを実行	N/A	N/A	
コンティニュー部品	正常終了	エラー終了	N/A	N/A	※コンティニュー部品の飛び先の部品の結果には無関係
サブジョブネットワーク待ち合わせ	待ち合わせのサブジョブネットワークの結果がそのまま反映される				
ジョブネットワーク待ち合わせ	待ち合わせのジョブネットワークの結果がそのまま反映される				実行したジョブネットワーク部品の結果
ファイル待ち合わせ	正常通過時	タイムアウト時	N/A	N/A	
時刻待ち合わせ	指定時刻で通過後	N/A	N/A	N/A	
イベント送信	送信成功	送信タイムアウト	N/A	N/A	
イベント受信	受信成功	受信タイムアウト	N/A	N/A	

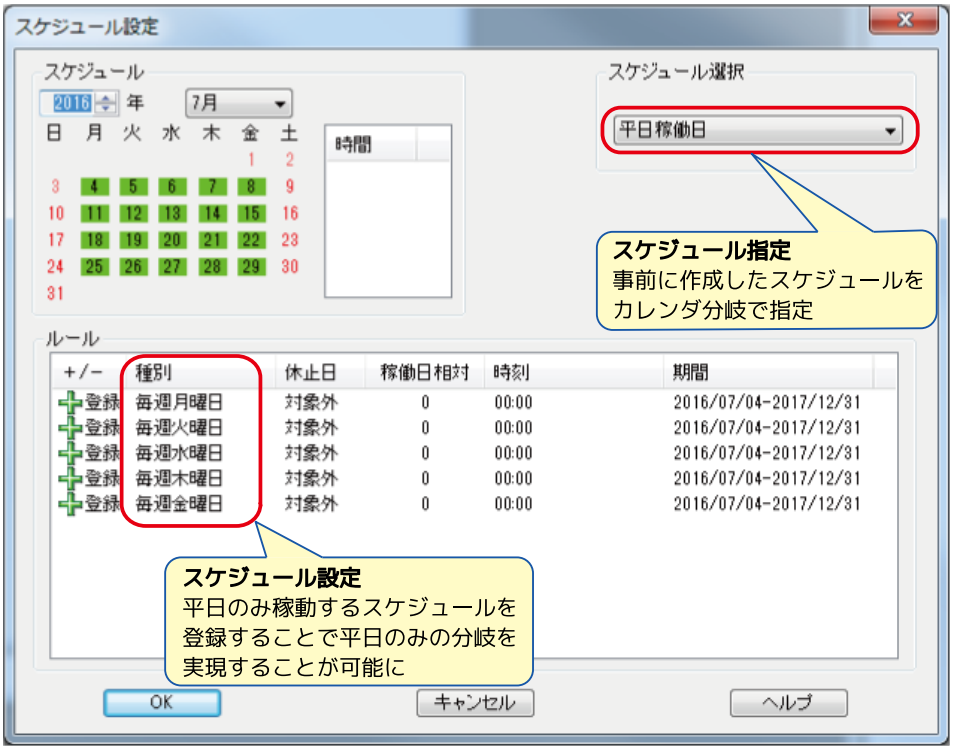
3.3.19. 日付によって動作を変更したい

曜日で処理を変えたい場合、カレンダ分岐部品を利用すると、その部分だけを変更できます。

例えば、平日は差分バックアップ、週末（土曜日）は完全バックアップを行うようなジョブネットワークを作成したい場合などに使用します。また、他の処理は曜日により変えたくない場合にカレンダ分岐部品を利用することで、その一部分だけを変更することができます。

フローにカレンダ分岐部品を配置するとフローが分岐され、[スケジュール設定]が表示されます。この画面で[スケジュール選択]でスケジュールを指定し、[ルール]でスケジュールを登録して設定します。





補正時刻により動作基準時刻が変化するのでどのタイミングで判定するかを確認する必要があります。

3.3.20. 同期処理をするためのポイント

同期処理を実現するために利用できる部品は次の通りです。

JobCenterにおける処理の同期/待ち合わせは、基本的にイベント連携機能を利用して同期/待ち合わせを行います。ただし、条件によっては、ジョブ/ジョブネットワーク待ち合わせ部品を利用できます。待ち合わせ/同期する対象や範囲によって適切な部品は変わりますので、要件に応じて使い分けることをお勧めします。ここでは、使い分けのポイントを説明します。

■イベント送信/受信部品

■ジョブ/ジョブネットワーク待ち合わせ部品

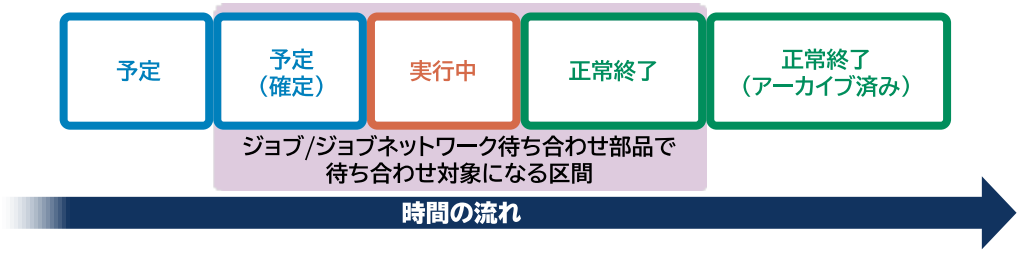
■ファイル待ち合わせ部品

3.3.20.1. 待ち合わせ対象による使い分け

対象のトラッカ	ジョブ/ジョブネットワーク待ち合わせ部品	イベント送信/受信	ファイル待ち合わせ部品
他のマシンのトラッカ	×[注1]	○	○[注3]
他のユーザのトラッカ	×[注1]	○	○[注3]
同一ユーザのトラッカで異なるルートトラッカ	×[注1]	○	○[注3]
同一ユーザのトラッカで同一ルートトラッカ	○	△[注2]	○[注3]

注1　ジョブネットワーク/ジョブ待ち合わせの対象決定はトラッカの投入順序に影響を受けるため、意図せず（待ち合わせされず）部品を通過する場合があります。

参照	<クラシックモード用環境構築ガイド> 12章 「イベント連携」



注2　機能的な問題はないが、ジョブ/ジョブネットワーク待ち合わせ部品の方が専用のため、わかりやすい。

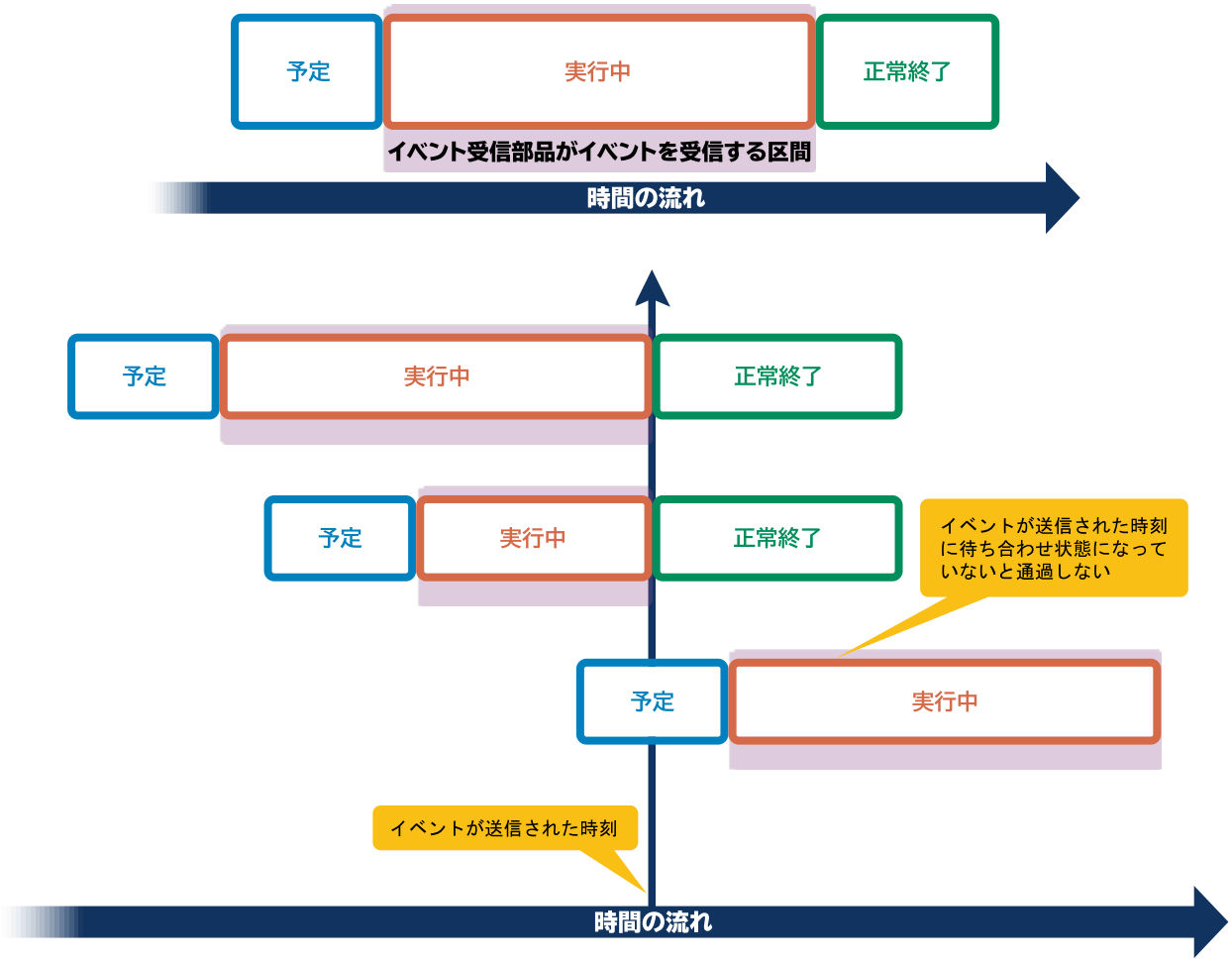
注3　トリガになるファイルの作成、削除を別途行う必要あります。

3.3.20.2. 同期範囲による使い分け

同期したい範囲	ジョブ/ジョブネットワーク待ち合わせ部品	イベント送信/受信	ファイル待ち合わせ部品
JobCenter外から制御したい	×	○注1	○
あるタイミングの時点で待ち合わせている部品を通過させたい	○	○	○
あるタイミング以降に待ち合わせになった部品を通過させたい	○	×注2	○注3

注1　jnwevtcmd　コマンドを利用して、手動やJobCenterと関連のないプログラムからのイベント発行が可能です。

注2　送信されたイベントは受信された時点で削除されるため、イベントが送信されたときにイベント受信部品がまだ起動していなかった場合、そのイベント受信部品は通過しません。



注3　通過条件になるフラグファイルの状態が通過条件を満たす必要があります。ファイル自身が動作フラグとして振る舞うことに注意してください。

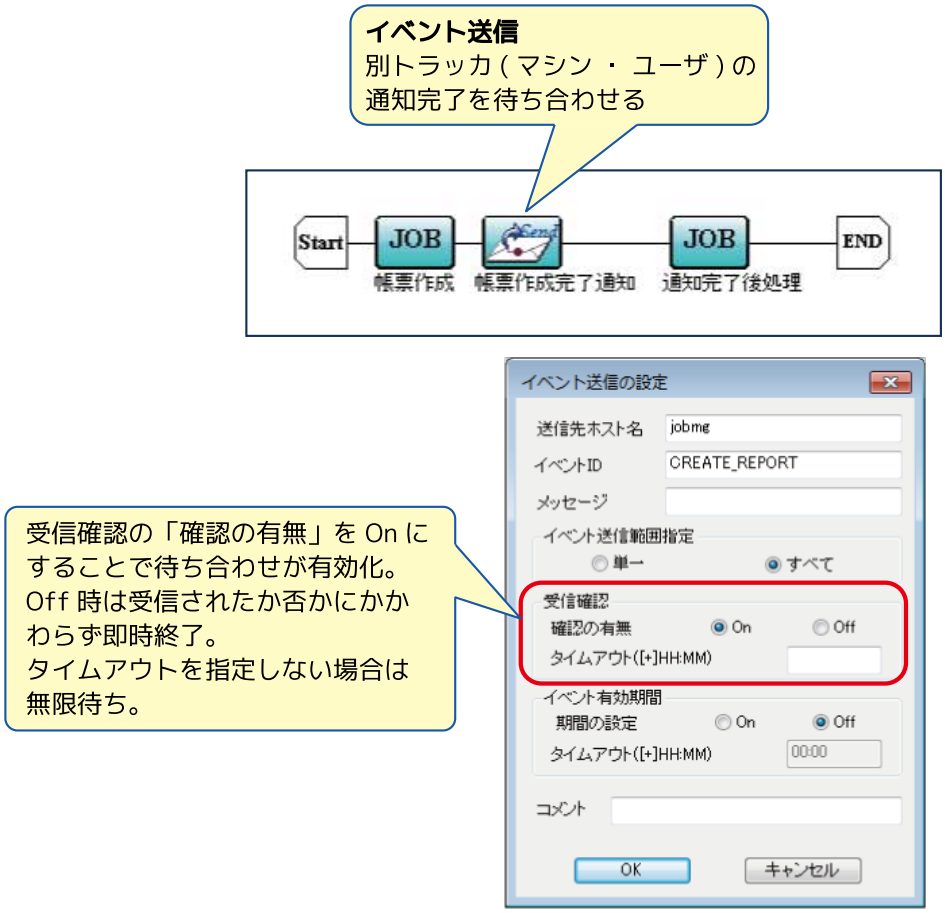
3.3.21. 他のマシン・他のユーザのトラッカと同期して処理したい

他のマシン・他のユーザのトラッカと同期して処理したい場合、「イベント送信部品」と「イベント受信部品」を対で作成することで、互いを待ち合わせることができます。その場合、イベント送信部品では[受信確認]の[確認の有無]で[On]を選択し、イベント受信部品では[タイムアウトの有無]で[On]を選択してください。

イベント送信側で、フローにイベント送信部品を配置すると[イベント送信名の設定]が表示されるので[イベント送信名]に名前を入力します。[OK]を選択すると[イベント送信の設定]が表示されます。このダイアログで、[送信ホスト名]に送信するホスト名を、[イベントID]にイベントID名を、[受信確認]の[確認の有無]で[On]を選択します。

イベント受信側で、フローにイベント受信部品を配置すると[イベント受信名の設定]が表示されので[イベント受信名]に名前を入力します。[OK]を選択すると[イベント受信の設定]が表示されます。このダイアログの[条件設定]タブで、[送信元ホスト名]に送信されるホスト名を、[イベントID条件一覧]にイベントIDを設定します。次に[タイムアウト]タブで、[受信タイムアウト]の[タイムアウトの有無]で[On]を選択します。

[Off]を選択すると受信できたイベントの有無にかかわらず即時正常終了します。また、[タイムアウト（[+]HH:MM）]を指定しない場合、無限に待ち合わせます。



3.3.22. 同一ユーザのトラッカで異なるルートトラッカと同期して処理したい

同一ユーザのトラッカで異なるルートトラッカと同期して処理したい場合の対応については、次の項目をご覧ください。

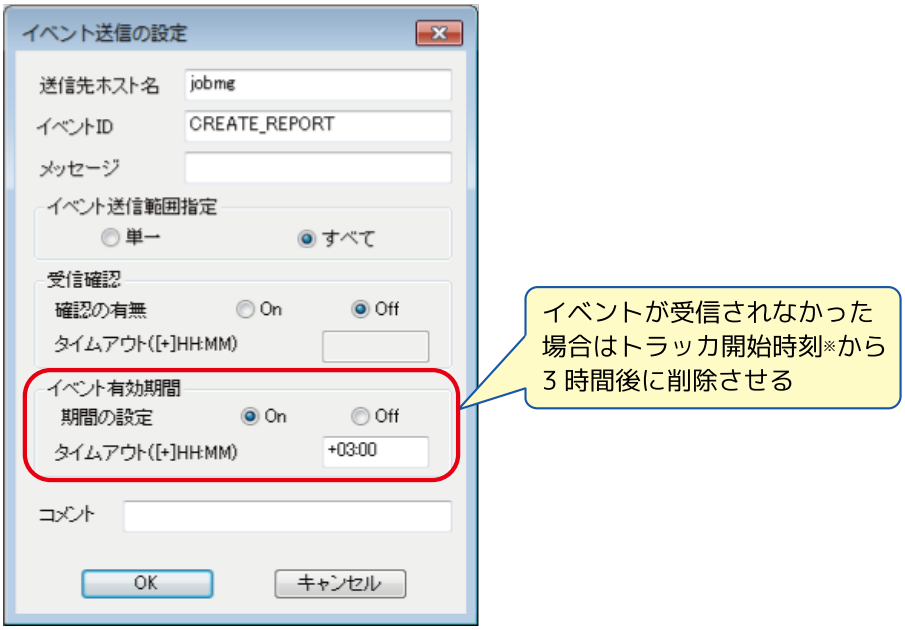
参照 「3.3.21 他のマシン・他のユーザのトラッカと同期して処理したい」

3.3.23. 不要なイベントを残したくない

イベント送信部品パラメータの「イベント有効期間」を利用すると、有効期間内に受信されないイベントを削除できます。イベント送信部品を右クリックして表示されるメニューから[設定]を選択し、[イベント送信の設定]を表示します。このダイアログで[イベント有効期間]の[期間の設定]で[On]を選択し、[タイムアウト（[+]HH:MM）]に時間を入力します。

トラッカの異常終了や停止などで、対になるイベント受信部品が実行されないと、イベントは送信先ホストで保持されたままになります。次回、対になるイベント受信部品が実行されると、保持されていたイベントを受信してしまうため、意図した動作にならない可能性があります。

このような場合、イベント送信部品パラメータの「イベント有効期間」が利用できます。



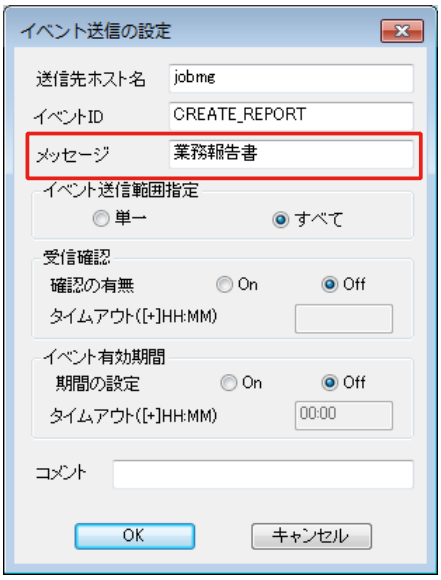
相対時間の基準は最上位のジョブネットワークの開始時刻です。ただし、オプション指定で当該部品の開始時刻を基準にすることもできます。

3.3.24. 連携するトラッカで作成したファイルの名前/パスを受信側で利用したい

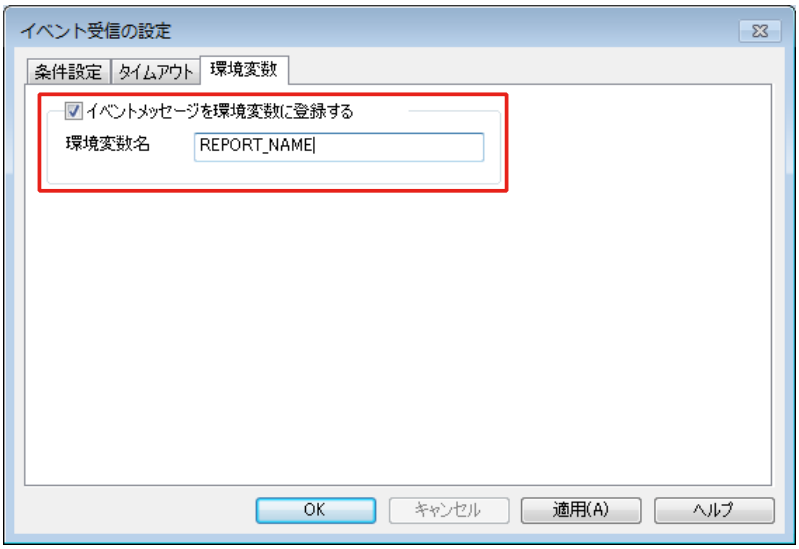
イベントを同期トリガとして使うだけでなく、イベントに添えられたメッセージによって動的に処理を行いたい場合、「イベント受信部品」の[イベントメッセージを環境変数に登録する]で指定します。

イベント受信部品を右クリックし、表示されたメニューで[設定]を選択すると、[イベント受信の設定]が表示されます。このダイアログの[環境変数]タブの[イベントメッセージを環境変数に登録する]を選択し、[環境変数名]に入力します。

送信側



受信側



■環境変数はトラッカ（ジョブネットワーク）単位で管理されます。

■継承された環境変数は、後続のどの単位ジョブでも参照できます。

■同じ環境変数を後続部品でさらに継承設定した場合、その内容で上書きされます。

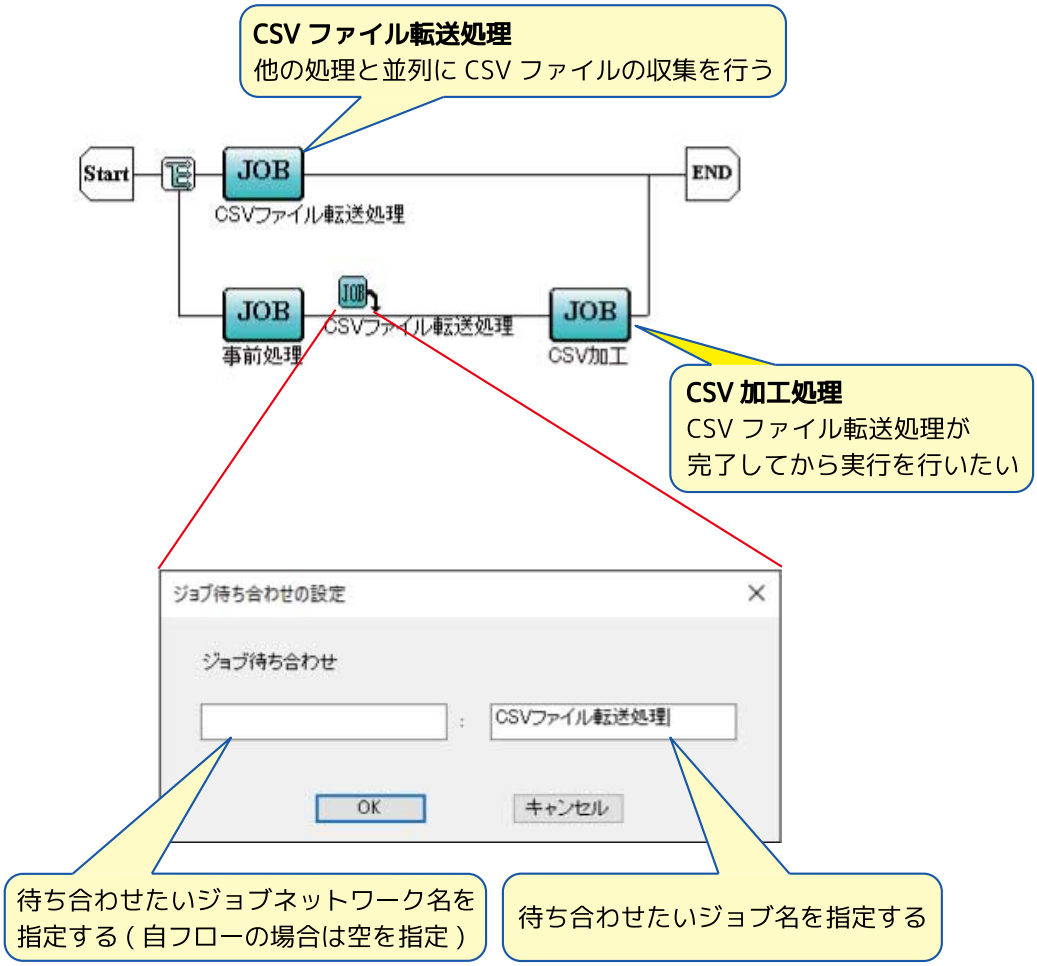
3.3.25. 単位ジョブが終了するまで待ち合わせてから処理を継続したい

「ジョブ待ち合わせ部品」を利用すると、自ジョブネットワークまたは他のジョブネットワーク中の他のジョブの終了を待ってから後続のフローを実行することができます。次のような場合、「ジョブ待ち合わせ部品」を利用します。

■並列処理を行っている際に、別フロー中の単位ジョブの実行が完了したことを保証してから処理を実行したい

■ジョブ間で依存関係は存在するが総実行時間を短縮するために並列に実行させたい

フローにジョブ待ち合わせ部品を配置すると[ジョブ待ち合わせの設定]が表示されます。このダイアログで、左の欄はジョブネットワーク名を入力しますが、同じフローの場合、空欄にします。右の欄に待ち合わせる部品名を設定します。



サブジョブネットワーク中のジョブを待ち合わせる場合は、「:」でつなぎ記述します。

【例】

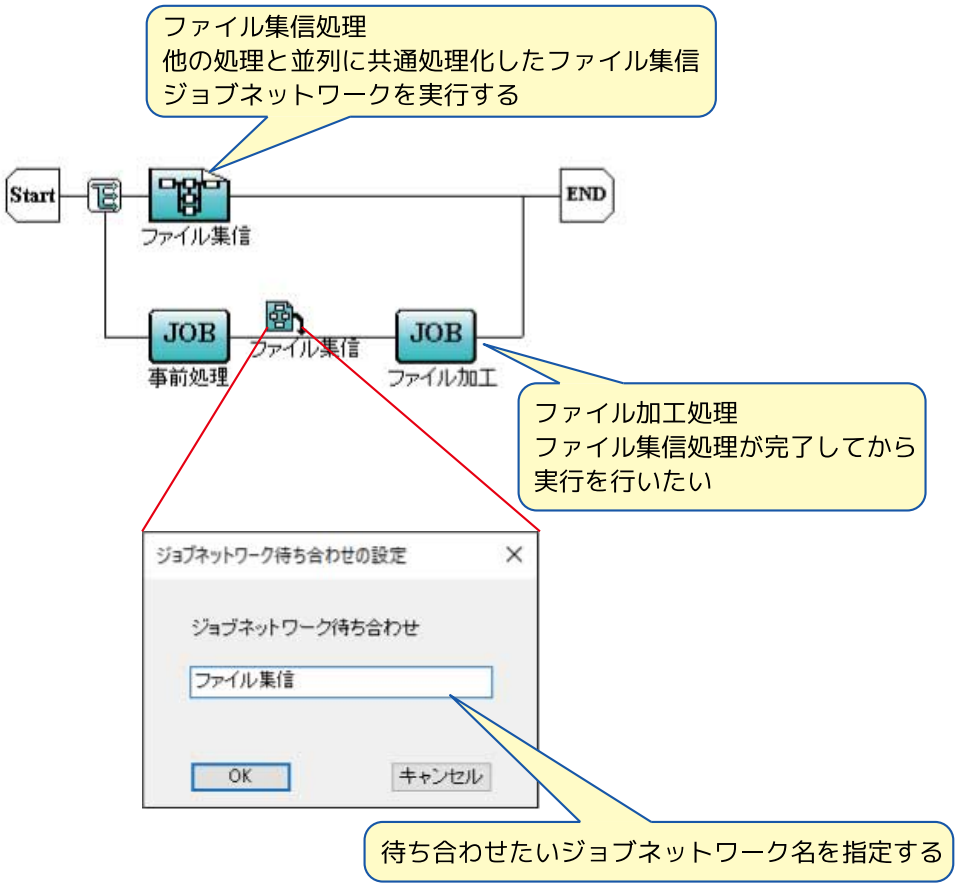
[最上位ジョブネットワーク名:サブジョブネットワーク名:・・・]

3.3.26. 他ジョブネットワークが終了するまで待ち合わせてから処理を継続したい

「ジョブネットワーク待ち合わせ部品」は他のジョブネットワークの終了を待ってから後続のフローを実行することができます。

並列処理を行っている際に、ジョブネットワークの実行が完了したことを保証してから処理を実行したい（自ジョブネットワーク中のサブジョブネットワークの完了を待ち合わせたい）場合、「ジョブネットワーク待ち合わせ部品」を利用します。

フローに「ジョブネットワーク待ち合わせ部品」を配置すると[ジョブネットワーク待ち合わせの設定]が表示されます。このダイアログの[ジョブネットワーク待ち合わせ]に待ち合わせるジョブネットワーク名を設定します。



サブジョブネットワーク中のジョブを待ち合わせる場合は、「:」でつなぎ記述します。

【例】

[最上位ジョブネットワーク名:サブジョブネットワーク名:・・・]

3.3.27. ファイルの作成や削除を待ち合わせてから処理を継続したい

「ファイル待ち合わせ部品」を利用すると、次のようにJobCenter（単位ジョブ）とは別で処理されたファイルが作成されたのを契機で処理を継続させることができます。

■ある集計用アプリケーションが集計結果CSVファイルを作成する

■バックアップソフトウエアがバックアップファイルを作成する

■FTPでファイルの転送処理を行う

フローに「ファイル待ち合わせ部品」を配置すると[ファイル待ち合わせの設定]が表示されます。このダイアログの[待ち合わせファイル]タブで[ディレクトリ名]に待ち合わせファイルが格納されるディレクトリ名を、[待ち合わせ対象ファイル名またはディレクトリ名]の欄に待ち合わせ対象ファイル名またはディレクトリ名を、[ディレクトリまたはファイル名に日付マクロを使用する]を選択し、設定します。

■ファイル名に日付が入るような場合は、日付マクロが有効です。

【例】日時バックアップする際に、ファイル名が「backup-20160603.zip」のような場合、「backup-%YYYY%%MM%%DD%.zip」のように指定できます。毎日指定フォルダにファイルが生成され続けるような場合、当日データのみ待ち合わせし、過去日のデータを待ち合わせたくない場合に有効です。

■部分一致などで条件づけをした場合、条件を満たしたファイル名を環境変数に入れることができます。これにより、後続のジョブでマッチしたファイルを処理したり、ログやメール通知等に残したりできます

デフォルトでは存在の有無のみチェックするため、作成が確認されたら通過します。ファイルの更新完了まで待ち合わせる場合は設定の変更が必要です。

3.3.28. ある時刻まで待ってから処理を継続したい

「時刻待ち合わせ部品」は指定した時刻まで待ち合わせる部品です。時刻は、絶対時間（HH:MM）か相対時間（+HH:MM）で設定します。

次のような理由である時刻まで待ってから処理を継続したい場合、「時刻待ち合わせ部品」を利用します。

■データベース（DB）への登録が完了する時刻がわかっていて、それまで待ち合わせたい

■フローの進捗が早くても、システム負荷を抑えるために特定のポイントで一定時間待ち合わせたい

フローに「時刻待ち合わせ部品」を配置すると[時刻待合の設定]が表示されます。このダイアログで[日付]、[時刻]に待ち合わせ時刻を設定します。

■相対時刻の場合は先頭に「+」を付与して指定します。

【例】5分後に指定する場合

+00:05

■絶対時刻の場合は、ジョブネットワークの開始日時を基準にして指定した時刻まで待ち合わせを行います。

【例】0日後の15:00を待ち合わせに指定した場合

14:00にジョブネットワークが開始した場合は、当日の15:00まで待ち合わせを行います。

16:00にジョブネットワークが開始した場合は、翌日の15:00まで待ち合わせを行います。

3.3.29. 大量のジョブ定義を効率的に作成したい

作成しなければならないジョブが大量にある場合、それらをCL/Winから1つ1つ作成し、かつ、パラメータを1つ1つ定義するのは手間がかかります。

このような場合、JobCenter Definition Helper（オプションライセンス）が利用できます。JobCenter Definition Helperでは、数多くのジョブの定義を作成しなければならないような場合、Excelの機能を利用して、部品の作成、削除、コピー、検索が容易に行えるため、効率的に作成できます。

参照	<Helper機能利用の手引き> 「2.1 概要」

3.4. スケジューリング

3.4.1. スケジューリングのポイント

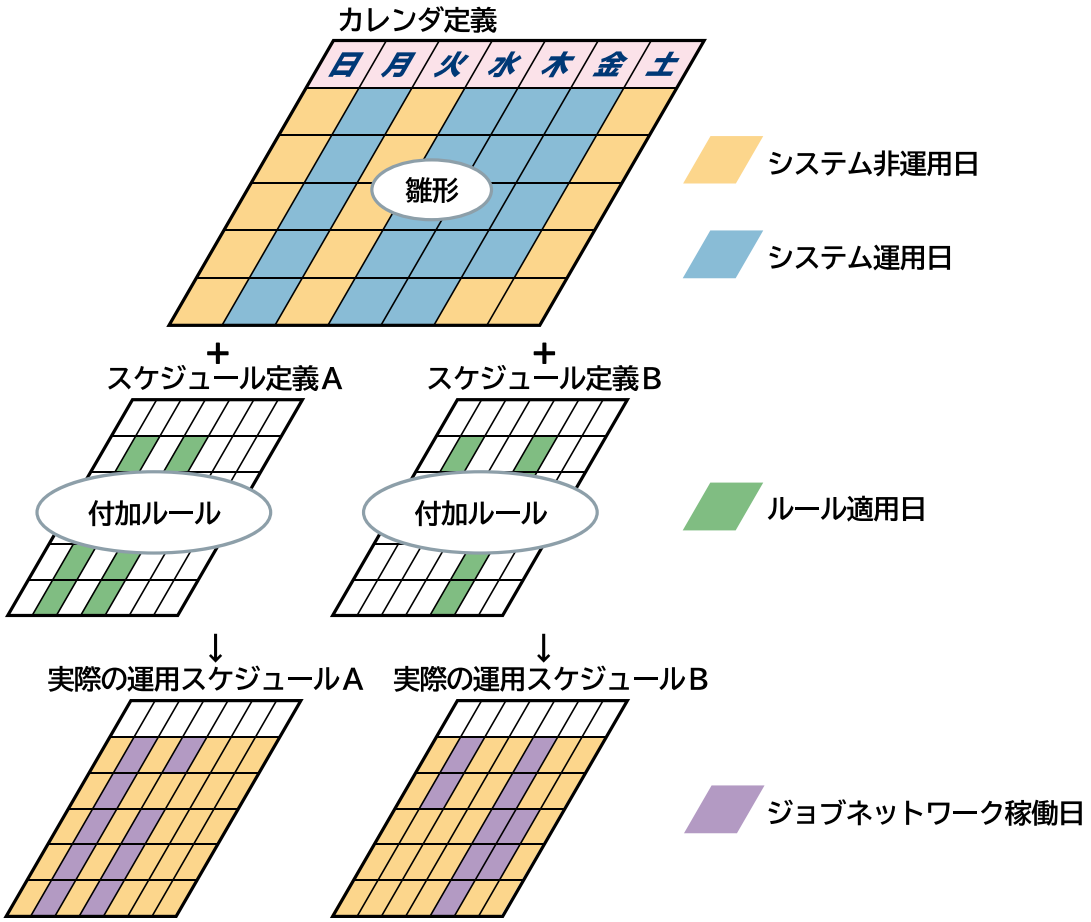
スケジューリングのポイントを説明します。

3.4.1.1. JobCenterにおけるスケジュールの概念

JobCenterのスケジュール管理機能は、構築から運用まで、開発者やオペレータのニーズに合うための様々な機能やインタフェースを提供しています。

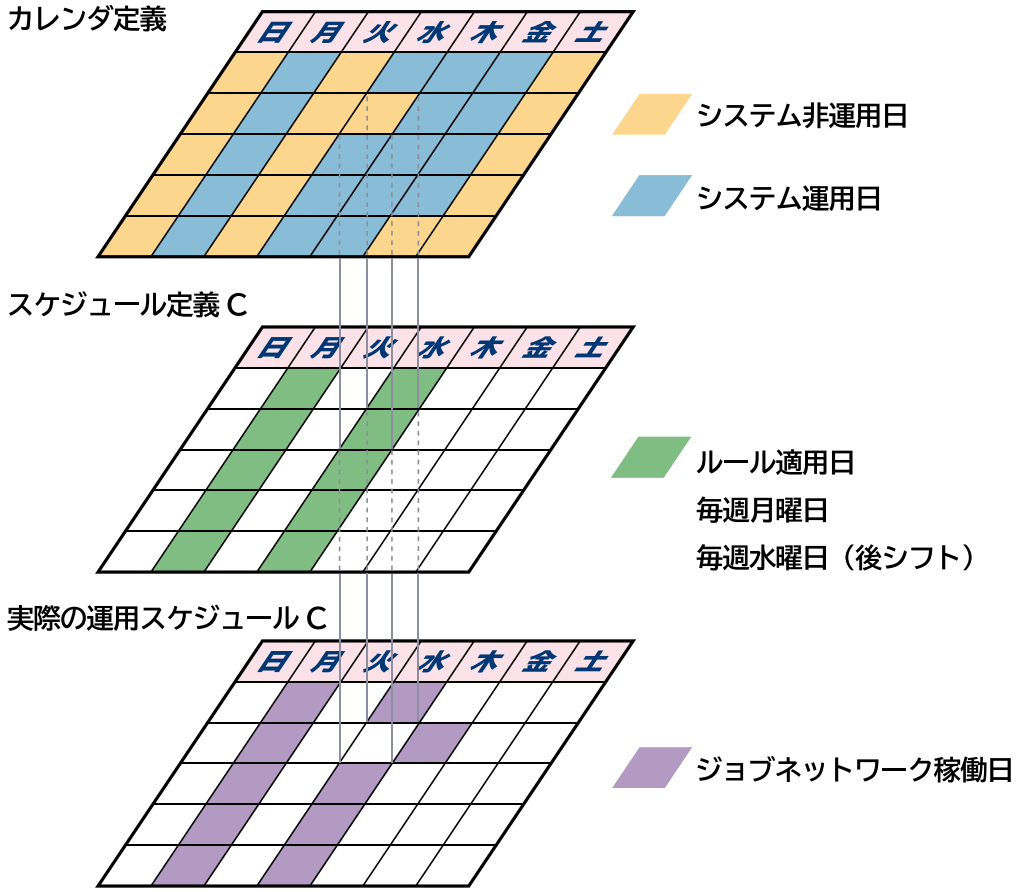
■稼働日カレンダ定義とスケジュール定義を用いた効率的なスケジュール設定が可能

ジョブの実行スケジュールの定義は、システムの運用日に関しての雛形となる「稼働日カレンダ」と呼ばれる定義と、その雛形をより詳細にカスタマイズする「スケジュール」と呼ばれる定義の2層構造となっているため、大規模な構築が効率的に行えます。 （カレンダ定義を使わず、スケジュールのみでも構築可能です）



■ルールを加算できるので複雑な運用スケジュールの定義が可能

「第3水曜日、毎月6日、毎年4月1日、運休日と重なる場合は1日シフトする」など、ルールを加算していくことで、複雑な業務形態でも対応し、自動運用を実現します。



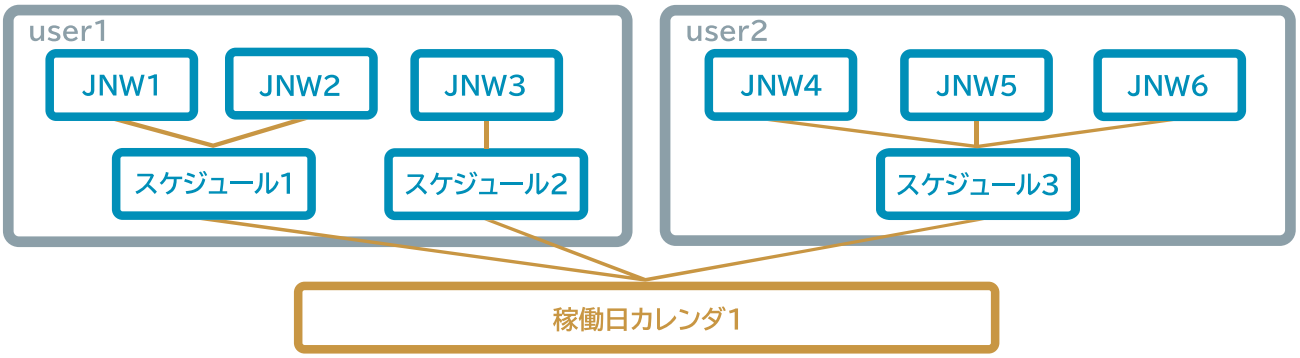
3.4.1.2. 各定義の関連について

各定義の関連を説明します。

■JobCenterにおけるスケジュールは、それ自体が1つの定義情報となっているため、「スケジュール定義」と呼びます。　他のジョブ管理ツールの場合、ジョブネットワークのパラメータや属性的な扱いとなっているものが多くありますが、JobCenterはそうではありません。

■どのジョブネットワークがどのスケジュールで動作するかについては、ジョブネットワークとスケジュールの関連付けを行うことで実現します。また、稼働日カレンダはスケジュールと関連付けられます。

- 稼働日カレンダとスケジュールは1:Nの関係
- スケジュールとジョブネットワークは1:Nの関係
- 稼働日カレンダはシステム全体で共通の所有物。ジョブネットワークとスケジュールは各ユーザの所有物。



3.4.2. カレンダの作成

カレンダの作成を説明します。

3.4.2.1. 稼働日カレンダについて

稼働日カレンダは、システムで共通の定義情報でJobCenter管理者ユーザのみが作成・更新できます。一般ユーザは参照のみ可能で、すでに作成されている稼働日カレンダを、一般ユーザ自身のスケジュール定義と関連付けすることができます。

3.4.2.2. 稼働日カレンダ作成時のポイント

稼働日カレンダを作成するポイントを次にお知らせします。

■稼働日カレンダは、対象システムの運用日（稼働日）・休止日（非稼働日）を決めるためのもので、基本的には1つ作成すればよいです。複数のサブシステムがあり、それごとに休止日が異なる場合は複数作成することもあります。

■稼働日カレンダを作成する場合、最初に「毎日」を稼働日として登録した上で、非稼働日となる日を除外で重ね合わせて作成するとわかりやすいため、お勧めします。

■1から作成すると非常に時間がかかります。日本の祝祭日を非稼働日として作成したJapanカレンダをサンプルとして製品に同梱しているので、それを利用して、それぞれのシステム向けにカスタマイズして作成すると便利です。

3.4.3. スケジュールの作成

スケジュールの作成を説明します。

3.4.3.1. スケジュールについて

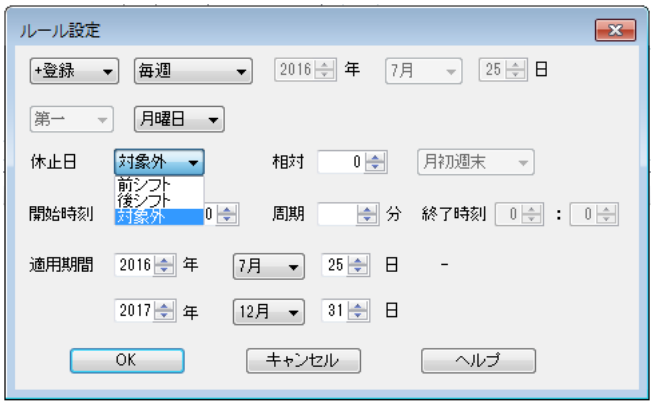
スケジュールは、ユーザごとの所有物（定義情報）です。スケジュールをジョブネットワークに関連付けることで、複数のジョブネットワークを同じスケジュールで運用できます。

3.4.3.2. スケジュール作成のポイント

■ルールを重ね合わせることで複雑なスケジュールを作成できます。

■休止日シフトについて

休止日シフトは、ルールで設定したジョブの運用を行う日（稼働日）が運用を行わない日（休止日）に当たった場合に稼働日を変更する機能です。[ルール設定]の[休止日]で「対象外」、「前シフト」、「後シフト」から選択して、利用します。



例えば、毎週月曜日にジョブの運用を行うルールで、月曜日が祝日だった次のカレンダの場合、「後シフト」、「前シフト」、「対象外」では次のように動作が異なります。

稼働日カレンダ(2016/7)

日	月	火	水	木	金	土
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

後シフト：翌日以降の稼働日で最も近い稼働日の19日にシフトします。

前シフト：前日以前の稼働日で最も近い稼働日の15日にシフトします。

対象外：第4週は実行されません。

■稼働日相対の考え方

起点となる日を決め、そこから稼働日だけを数えて±X日した日に実行します。起点となる日は、ルール種別と休止日シフトで決まるため、休止日シフトが「対象外」になっていると起点となる日自体が消えてしまうので、ご注意ください。

例として、毎月第5営業日（その月の5回目の稼働日）に実行したい場合は、次のような設定にします。

「毎月1日」：起点を「毎日1日」に設定します。

「後シフト」：1日が非稼働日の場合、起点となる日が対象外と判断されるため、その月が実行されなくなるのを防ぐため、「後シフト」を設定します。

「稼働相対+4」：起点の1日が土曜日だった場合、後シフトにより月最初の稼働日が3日の月曜になるため、稼働相対で「+4」を設定します。

3.4.3.3. グルーピングについて

スケジュールの定義数に比べて、ジョブネットワークの定義数は少ない場合がほとんどです。そのため、グループ化して管理する必要性は低いが、スケジュールの定義数が多い場合は分類して管理することを推奨します。種別ごと、サブシステムごとなど論理的な単位でグループ化することをご検討ください。グループ単位で一括して、スケジュールの有効化/無効化ができるため、このことも考慮に入れ、スケジュールの分類をすることを推奨します。

3.4.3.4. スケジュールの無効化

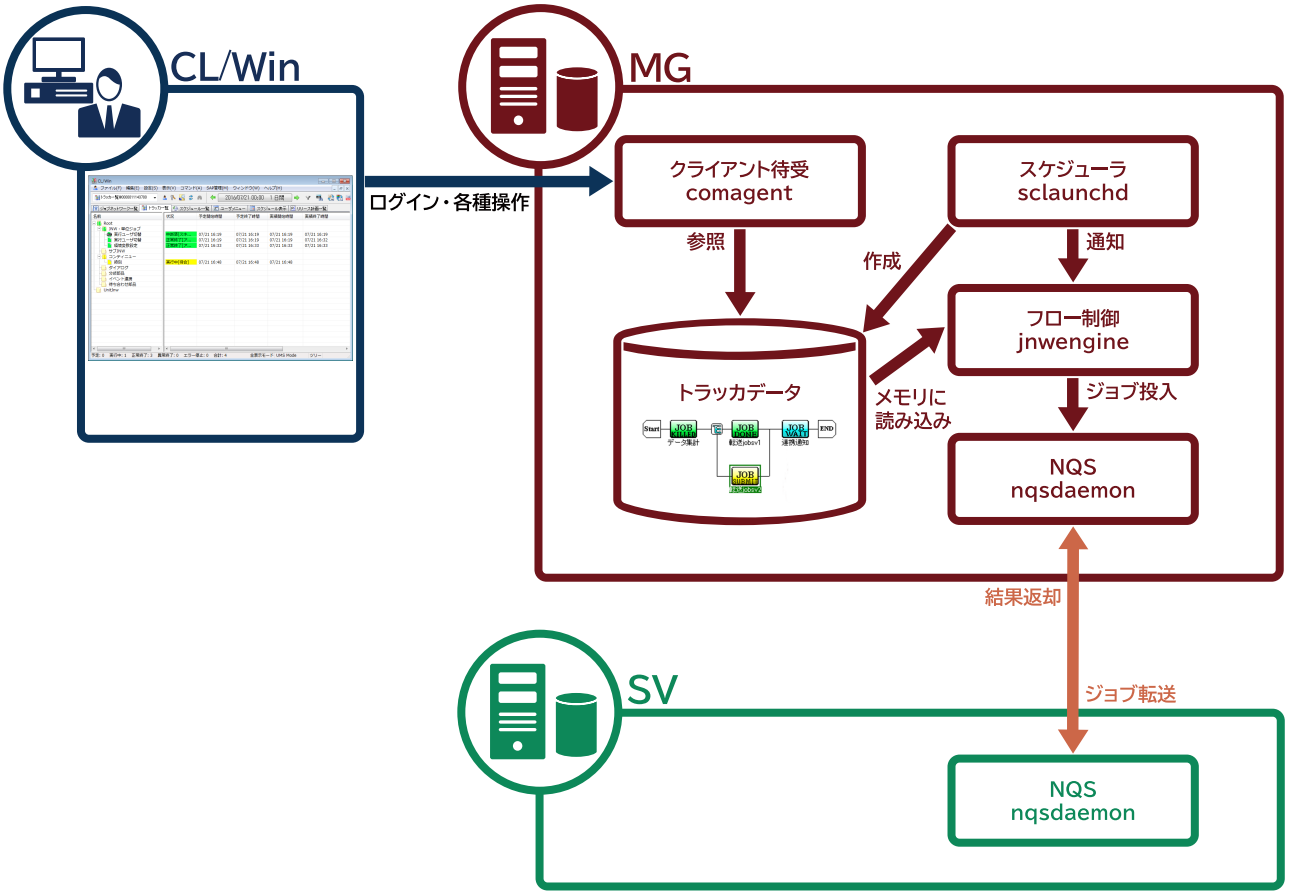
システムのメンテナンス日など、スケジュール実行を一時的に止めたい場合にスケジュールの無効化を行うことができます。スケジュール単体またはスケジュールグループに対して無効化ができます。

システムメンテナンスなどの理由により、ジョブの実行を一時的に抑止する方法はいくつかあります。「スケジュールの無効化」はその中の1つです。

参照 「4.2.1 キューの制御」

3.5. 補足：ジョブ実行時の動作について

ここでは、スケジューリングされたジョブネットワークが起動し、正常終了してアーカイブされるまでの流れを順に説明します。

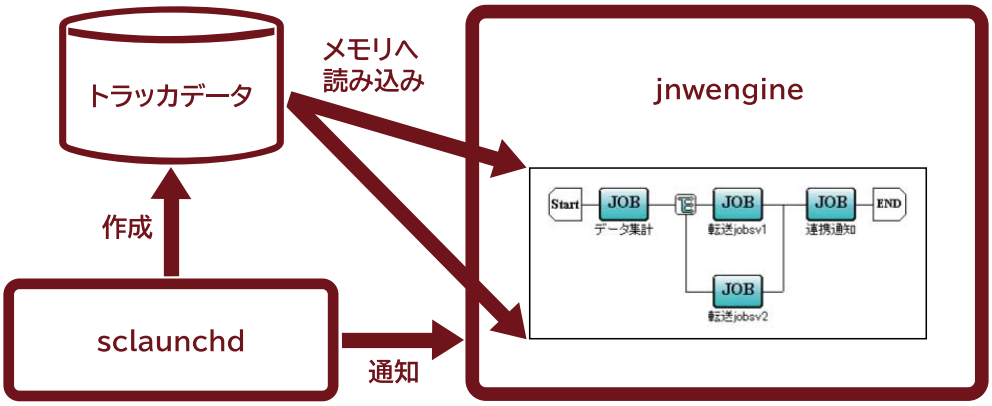


1. スケジュール起動

   スケジューリングされているトラッカが予定開始時刻を迎えると、スケジューラ（sclaunchd）はそのトラッカ用のデータをディスク上に生成し、フロー制御プロセス（jnwengine）に通知を行います。jnwengineはそれを受けて、対象データをメモリ上に読み込んで、フローを実行開始します。

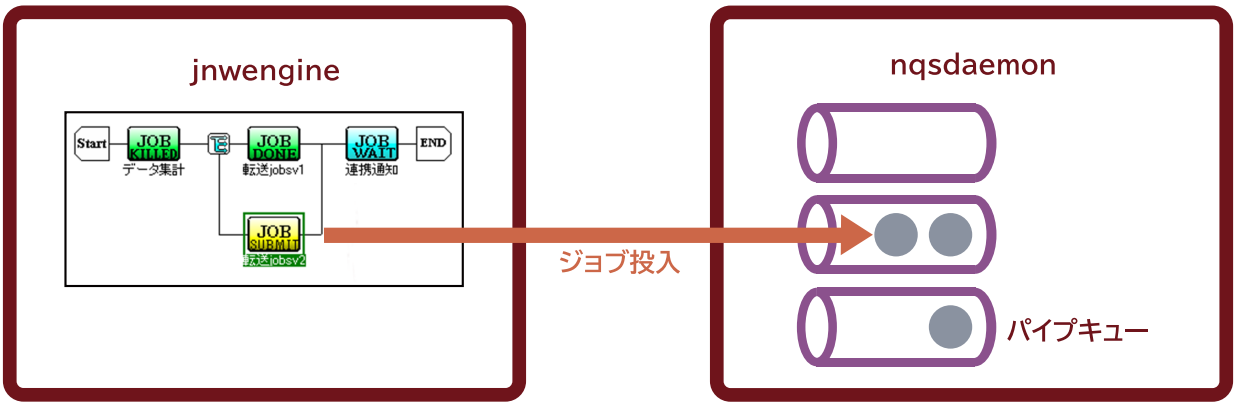
   次の2～5はジョブの数だけ繰り返し行います。

参照	「4.2.1 キューの制御」



2. フロー制御、ジョブ投入

jnwengineはフロー制御を行い、各部品の処理を順に実施していきます。フローが単位ジョブに到達した場合、jnwengineはそのジョブの実行依頼をNQS（nqsdaemon）へ依頼します。このとき、単位ジョブはリクエストとして指定されたキューにキューイングされることになります。

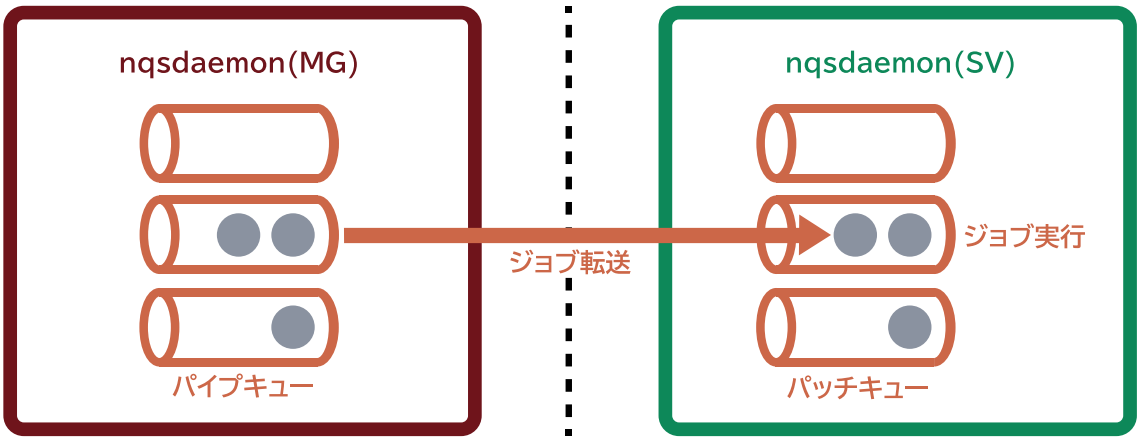


3. ジョブ転送

nqsdaemonは転送処理を行ってSVにジョブリクエストを転送します。

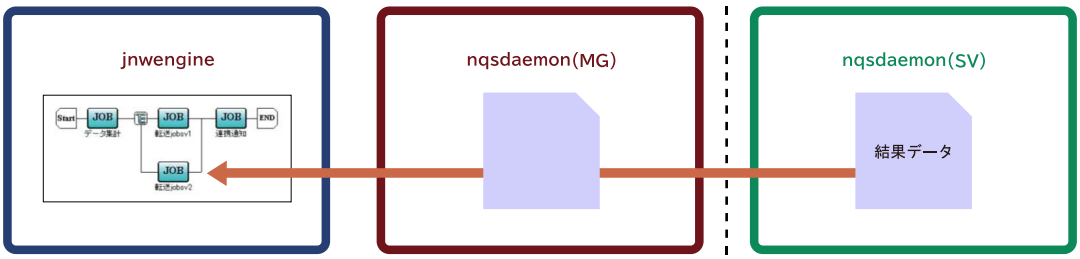
4. ジョブ実行

キューリングされたリクエストは、そのキューの中で順番待ちとなり、キューの同時実行可能数に空きができた際に実行処理されます。パイプキューであれば他キューへの転送処理、バッチキューであればジョブの起動処理が行われます。



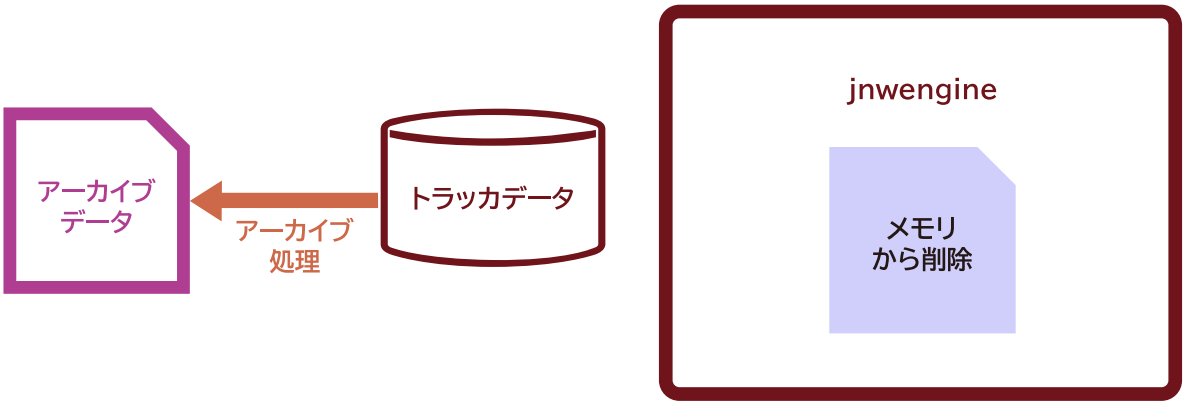
5. 結果返却

ジョブの実行が完了すると、その結果が依頼元へ返却されます。リモートのSVで実行された場合は、NQSを経由してjnwengineへ、MGで実行されていた場合は直接jnwengineへ返却されます。



6. 実行完了、アーカイブ

フローがENDに到達すると、そのトラッカは実行完了となります。再実行などの操作をしなければ、デフォルトでは10分後にアーカイブ処理（ディスク上にあるトラッカデータのアーカイブ化）が行われ、jnwengineのメモリからも該当トラッカデータが削除されます。



4. 運用編

本章ではJobCenterを運用する上で必要とされる考え方やポイントについて説明します。

4.1. 実行状況の監視

運用の場面において、ジョブやジョブネットワークが予定通りに実行されているか、正常に実行されているか、といったことを確認したい場合、トラッカ一覧画面で確認できます。トラッカ一覧画面では、目的に応じて様々な表示ができます。例えば、グラフィックモード形式での表示にすると各トラッカの進捗状況を視覚的に把握することできます。トラッカを各フィールドでソートすることもできます。

4.1.1. テキスト形式 / グラフィックモード形式での表示

デフォルトではテキスト形式で、それぞれのトラッカが何時に開始され何時に終了したのかといった情報がテキストで表示されます。

グラフィックモードの場合、この情報が横バーグラフ、いわゆるガントチャート形式で表示され、それぞれのトラッカの実況状況が一目でわかるようになります。各トラッカの進捗状況の様子を一目で視覚的に把握したい場合はグラフィックモードが適しています。

参照	<クラシックモード用基本操作ガイド> 「8.2 トラッカ一覧をテキスト／グラフィックモードで表示する」

4.1.1.1. ツリー表示 / ソート表示

デフォルトではツリー形式で、ジョブネットワーク一覧で定義したジョブネットワークグループのツリーに即した形で表示されます。この場合、トラッカ色がその優先順位に応じて上位ノード（グループノード）にも表示されるので、異常を素早く察知するのに適しています。例えば、トラッカ一覧画面を常時表示し、異常がないかどうかをたまにチェックするといったような目的に適しています。

ソート表示の場合、グループツリーを表示せず、すべて均等に表示させ、各フィールドでソートできるようになります。予定開始時刻やステータスなど、目的に応じてソートしたい場合はこちらで表示します。

参照	<クラシックモード用基本操作ガイド> 「8.4 トラッカ一覧をツリー表示する」

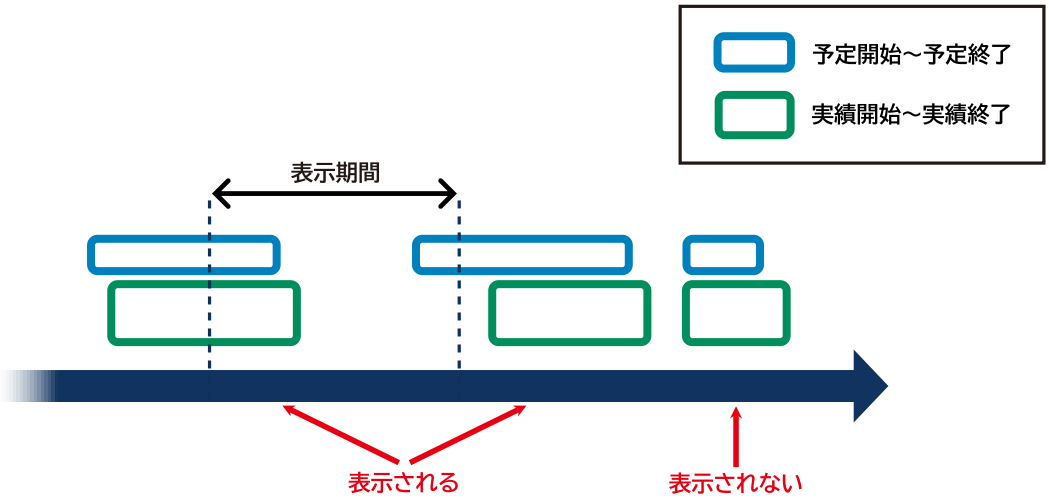
参照	<クラシックモード用基本操作ガイド> 「8.5 トラッカ一覧を予定開始時間の昇順で表示する」

4.1.1.2. 表示期間の設定

デフォルトでは当日1日分のトラッカが表示されています。ここに表示される条件は、対象の表示期間内に次のいずれかが重なっている場合です。

■予定開始時刻～予定終了時刻

■実績開始時刻～実績終了時刻



これを変更して表示期間を変更することができます。

参照	<クラシックモード用基本操作ガイド>「8.7 トラッカ一覧の表示期間を指定して表示する」

参照	<クラシックモード用基本操作ガイド>「8.9 トラッカ一覧の表示開始時間を指定して表示する」

4.1.1.3. 条件を指定してフィルタリング

表示されるトラッカ数が多い場合、条件を指定して絞り込むことができます。例えば、次のような条件でトラッカの絞り込みができます。

■特定のジョブネットワーク、特定グループに所属するジョブネットワークのトラッカだけを表示したい

■異常が発生しているトラッカだけを表示したい

参照	<クラシックモード用基本操作ガイド>「8.10 トラッカ一覧をフィルタリングして表示する」

4.1.1.4. 最新トラッカ表示

短い周期で繰り返し実行している場合、対象の表示期間内に同一ジョブネットワークのトラッカが数多く表示されることになります。このような場合に、直近で実行された最新の1回分だけを表示したい場合には、最新トラッカ表示モードを選択してください。

参照	<クラシックモード用基本操作ガイド>「8.3 最新のトラッカだけを表示する」

4.1.1.5. 応用的な監視（外部からの察知）

基本的にはジョブは自動実行されると正常終了されるはずであり、異常終了や遅延など何らかの確認の必要が生じた場合にのみトラッカ一覧画面を確認したいといったケースがあります。このような場合、その目的の事象（異常終了や遅延など）が発生した場合に、対象ユーザに通知してほしい場合、次の2つの方法があります。

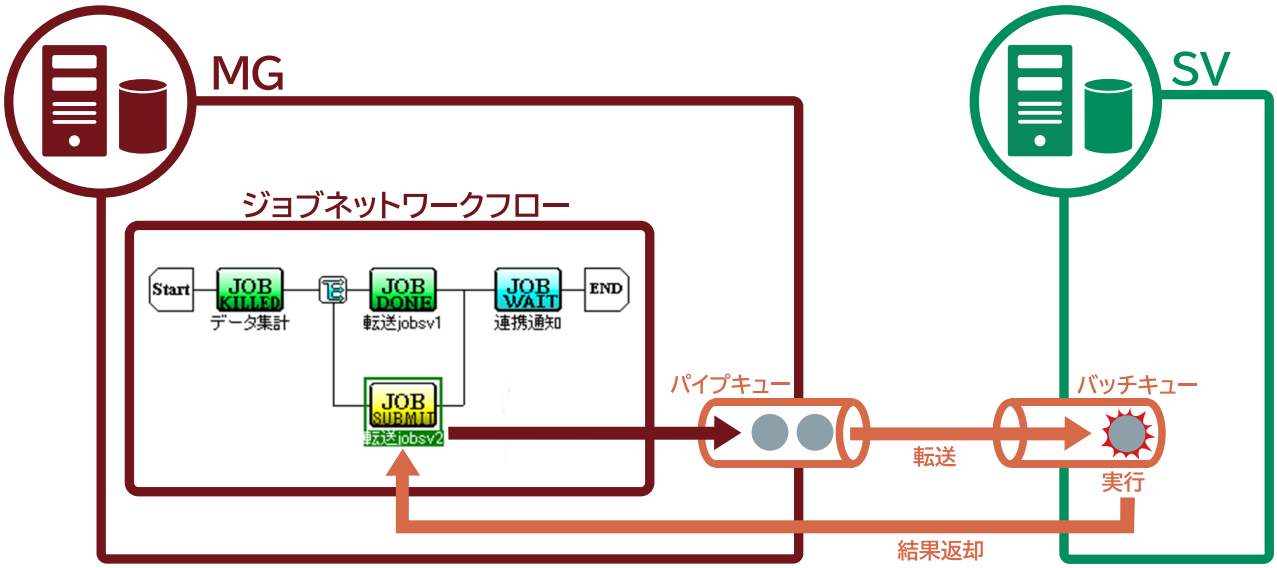
■メール通報

■イベント監視連携

参照	「1.3.2 監視設計」

4.1.1.6. キューの確認

トラッカを確認した結果、ジョブがSUBMITのままになっていたり、実行中のまま想定より時間がかかったりしているような場合、投入先のキューを確認して、リクエストの状況がどうなっているのかを確認します。



例えば、長時間SUBMITのままになっているような場合、以下の可能性が考えられます。

■キューの状態がSTOPになっている

■キューの同時実行数に空きがなく、順番待ちをしている

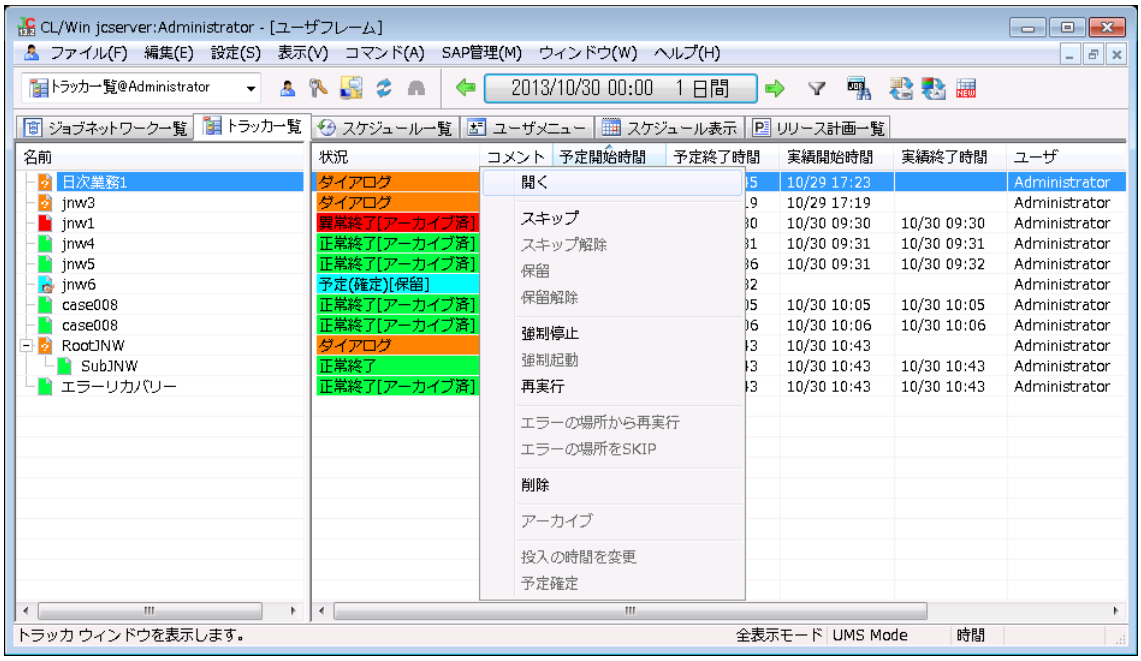
トラッカ上のジョブのステータスと、キュー内でのリクエスト状態の関係については次の表をご覧ください。

トラッカでのジョブのステータス	キューでのリクエストの状態
WAIT	・未実行 ・MGのパイプキューへの投入処理開始 ～
SUBMIT	・MGのパイプキューへの投入処理完了 ・SVのバッチキューへの転送処理処理 ～ 完了 ・SVのバッチキューでの順番待ち（キュー内での同時実行数に空き待ち）
RUN	・（空ができたため）SVのバッチキューで実行開始 ～ 実行完了 ・SVのバッチキューへの転送処理処理 ～ 完了 ～
DONE	・MGへの結果返却処理完了

4.1.2. トラッカの制御

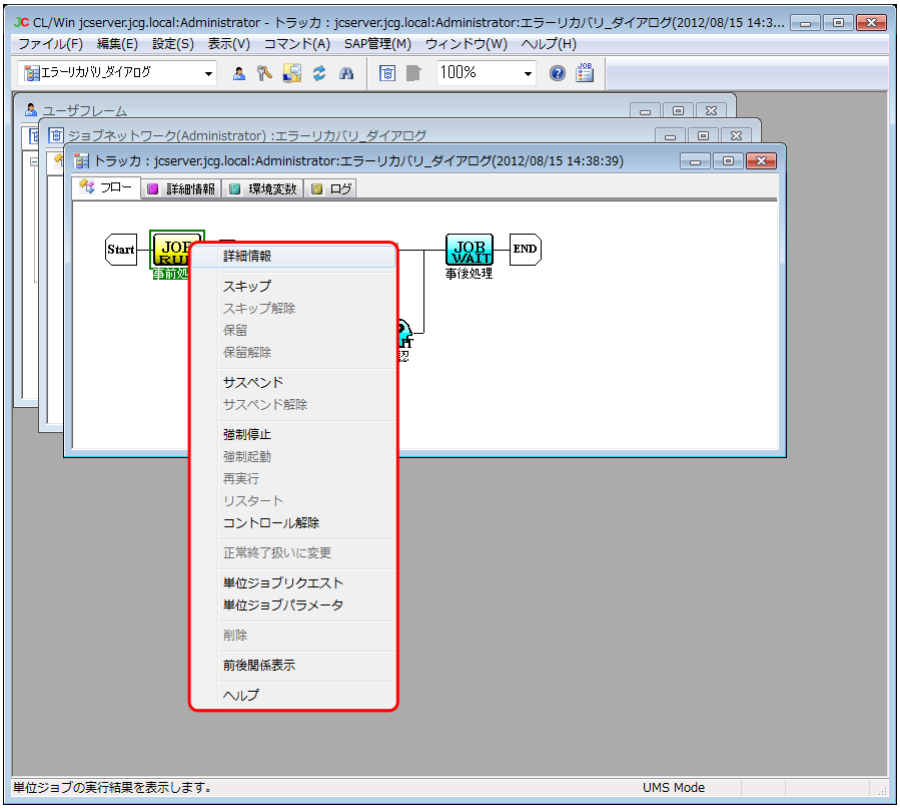
ジョブやジョブネットワークの異常終了を検知した場合、その異常の原因を取り除いた後、再実行して正常に処理させたい場合があります。または実行完了しない処理をスキップして、後続の処理を流してしまいたい場合があります。このような場合、トラッカに対して操作を行うことで対応ができます。トラッカの操作は、フロー全体に行う操作と部品個別に行う操作の2種類があります。

4.1.2.1. フロー全体



操作	説明
スキップ	トラッカをスキップ状態にします。実行前のトラッカの場合には、スキップ予定とし予定開始時刻がきても実行せずに終了します。 実行中のトラッカの場合には実行を中断します。
スキップ解除	スキップ状態を解除します。
保留	実行前のトラッカを保留状態にします。保留状態のトラッカは、予定開始時刻がきても保留解除を行うまで実行を待ち合わせます。
保留解除	保留状態を解除します。予定開始時刻がきて保留により待ち合わせていたトラッカは実行を開始します。
強制停止	実行中のトラッカを強制的に停止します。
強制起動	予定状態のトラッカを強制的に起動します。
再実行	フローの先頭から再実行を行います。
エラーの場所から再実行	エラー停止している部品からトラッカを再実行します。
エラーの場所をスキップ	エラー停止している部品をスキップしてトラッカを再実行します。
アーカイブ	アーカイブ時刻に到達する前に、トラッカをアーカイブします。アーカイブの操作ができるのは、正常終了、異常終了、警告終了しているトラッカのみです。
投入の時間を変更	トラッカの投入時刻を変更します。
予定確定	予定の状態のトラッカを予定確定状態に変更します。

4.1.2.2. 部品個別



操作	説明
スキップ	ジョブをスキップ状態にします。実行前のジョブの場合には、フロー到達時に実行することなく次に進みます。 実行中の場合には、実行を中断してフローを次に進めます。
スキップ解除	スキップ状態を解除します。
保留	ジョブを保留状態にします。保留状態のジョブは、フローが到達しても保留解除を行うまで実行を待ち合わせます。実行中のジョブに対して保留を行うことはできません。
保留解除	保留状態を解除します。
サスペンド	実行中の単位ジョブを一時停止状態にします。ジョブのプロセスにはSIGSTOPシグナルが送られます （Windows版では対応していません）。
サスペンド解除	サスペンド状態の単位ジョブを再開します。ジョブのプロセスにはSIGCONTシグナルが送られます （Windows版では対応していません）。
強制停止	実行中の単位ジョブを強制的に停止します。ジョブのプロセスにはSIGKILLシグナルが送られます。このときトラッカも停止します。
強制起動	この単位ジョブ以降のジョブを強制的に開始します。
再実行	この単位ジョブ以降の実行済みジョブを再実行します。
リスタート	ジョブステップリスタート機能を使用して単位ジョブのリスタートを行います。
コントロール解除	この単位ジョブをJobCenterの監視下から切り離します。対象となる単位ジョブは表示上エラー状態となり、実行結果の取得は行いません。ジョブネットワークの処理はコントロールを解除された単位ジョブの、次の部品から引き続き行われます。
正常終了扱いに変更	異常終了、警告終了している単位ジョブを正常終了扱いに変更します。
前後関係表示	この単位ジョブの前後関係を表示します。

4.2. キューの制御

4.2.1. キューの制御

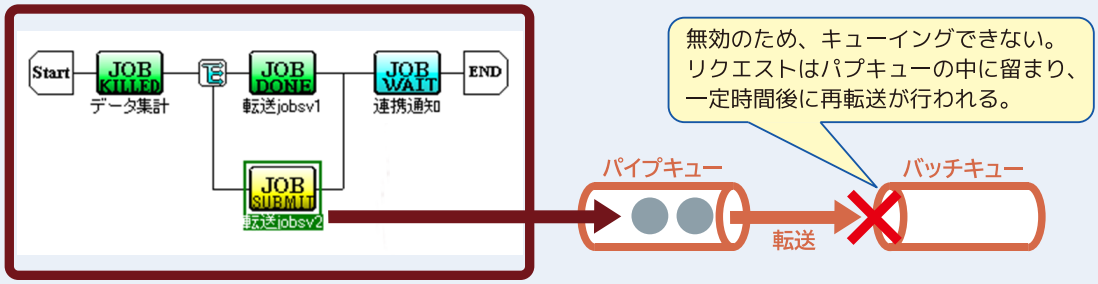
マシンメンテナンスなどの理由により一時的にそのマシンへのジョブ投入を止めたい場合、キューの制御を行います。

操作	画面上の表示	説明
開始	START	停止状態のキューを開始します。
停止	STOP	キューを停止します。停止状態のキューはリクエストの実行をしません。
有効	ENABLE	無効状態のキューを有効化します。
無効	DISABLE	キューを無効化します。無効化状態のキューはリクエストの受け付けを行わず、新たにキューイングされることはありません。

（※）パイプキューの場合はリクエスト転送
パッチキューの場合はリクエスト実行

投入先キューが無効でリクエストが受け付けられなかった場合、そのリクエストがどうなるかについては投入元によって異なります。

■トラッカフローの単位ジョブからキューへ投入しようとしていた場合、ジョブは異常終了します。



■パイプキューを経由して対象の無効状態のキューで転送しようとしていた場合、パイプキューの中に留まり、一定時間後に再転送が試みられる。

4.2.2. JobCenter起動時の制御

ジョブ実行中にJobCenterを再起動した場合、自動的にジョブが再実行される可能性があります[注]。　このような動作を抑止するため、JobCenter起動時のキューの状態を次の2つの方法で変更できます。

> 注　キューがSTART、かつ、対象ジョブのジョブパラメータ「RESTART」がENABLEに設定されている場合

■キューの再起動属性

キューの再起動属性を変更することで、JobCenter起動時のキューの状態を変更することができます（キューごとに設定が必要です）。

参照	<クラシックモード用NQS機能利用の手引き>「6.1.3.7 再起動属性」

【利用例】

- STOP

  キューを停止状態で起動し、ジョブが自動的に再実行されないようにしておきます。管理者やオペレータがシステムやアプリケーションの状況を確認した後にキューを開始して、処理を再開させます。

- PURGE

  キューへ投入されたジョブはすべて破棄（異常終了）します（処理に依存関係があるため、そのジョブを再実行してもうまくいかない）。状況を確認した後、JNWを再投入して、フローの先頭からやり直します。

■デーモンの設定（daemon.conf）

デーモン（常駐プロセス）の設定を変更し、JobCenter起動時に必ずすべてのキューを停止状態で起動させることができます。　上記のキューの再起動属性と異なり、こちらはシステム全体の設定となります。また、こちらの設定はSTOP状態での起動のみで、PURGEはありません。

参照	<クラシックモード用環境構築ガイド>「5.2 デーモン設定ファイルの使用可能パラメータ」の「2.NQSDAEMON_OPT」の-sオプション

4.2.3. 補足：JobCenter起動・停止の影響について

通常運用においては、MGが停止したり、SVのみが停止したりといったケースが発生します。このような場合の様々な懸念事項を次にまとめます。

4.2.3.1. JobCenter MG停止・再起動

■スケジュール起動の処理

- MG停止中はスケジュール起動が行われません。
- MG再起動時、停止中に予定開始時刻を迎えたトラッカが一斉に投入されます。

  例えば、MGが丸1日ダウンしており、その間、100個のトラッカが未投入となっていた場合、再起動時にその100個が一斉に投入されて実行開始されます。これを抑止したい場合、デーモン設定ファイル（daemon.conf）で設定変更が可能です。

参照	<クラシックモード用環境構築ガイド> 「5.2 デーモン設定ファイルの使用可能パラメータ」の「1.SCLAUNCHD_FORGET」

■トラッカフローの処理

トラッカ実行中にMGを停止した場合、再起動後に停止したときの状態から再開されます。

■ジョブの処理

MG上のキューにリクエストが存在する場合にMGが再起動された場合、そのリクエストがどうなるかについては、単位ジョブパラメータ「RESTART」の設定内容と、投入先キューの再起動属性の設定内容によって変わります。

		キューの再起動属性		
		RESTARAT	STOP / MIGRATION_STOP	PURGE / MIGRATION_PURGE
単位ジョブパラメータ「RESTART」	ENABLE	自動的に再実行される	キュー内に留まる。 キューをSTARTすると再実行される。	異常終了
	DISABLE	異常終了	異常終了	異常終了

■JobCenter CL/Winの接続

接続中だったCL/Winのセッションは強制切断されます。

■JobCenter SVへのリクエスト転送

リクエスト転送処理中にMGを停止しようとした場合、転送処理は強制停止されリクエストはキューに戻されます。

参照	再起動後の動作については、「4.2.3.1 JobCenter MG停止・再起動」の「ジョブの処理」をご覧ください。

■JobCenter SVからのリクエスト結果返却処理

SVにジョブリクエストの転送した後にMGが停止した場合、SV側では関係なくにジョブは実行されます。その後、結果返却がMGへと行われますが、MGがダウンしている場合にはこれが行えません。

ただし、一定時間ごとに再転送が繰り返されるので、MG再起動後に返却は行われることになります。

4.2.4. JobCenter SV停止・再起動

■ジョブの処理

SV上のキューにリクエストが存在する場合にSVが再起動された場合、MGと同じ動作になります。

参照	「4.2.3.1 JobCenter MG停止・再起動」の「ジョブの処理」を参照してください。

■JobCenter MGへの結果返却処理

結果返却処理は強制停止されます。SV再起動後に再開され、MGへ結果返却されます。

4.3. JobCenterサイトの状況の監視

4.3.1. JobCenterの自動復旧

JobCenterが停止する前に以下のような状況が起きると、ディスクアクセスができずデータファイル（AOFファイル）の更新が完了しないためAOFファイルが不正な状態となります。

■OSの停止

■マシンの電源喪失

■JobCenterプロセスの強制終了(kill -9等)

■ストレージ・ファイルシステムの破損

AOFファイルが不正な状態だとJobCenterが起動しないため、JobCenter起動時にAOFファイルの状態をチェックし、データの不正を検出した時にはAOFファイルの修復処理を行います。

この修復処理によって不正な箇所より後ろのデータは切り捨てられるため、AOFファイルの更新が完了しなかった場合、最後に行ったデータ更新が反映されずJobCenterの状態が巻き戻ります。

AOFファイルはJobCenterに対して行われたデータの更新を記録しているジャーナルファイルです。

AOFファイルの破損箇所によってはJobCenterの状態が大きく巻き戻ることがあります。

AOFファイルの修復が行われる場合は、予めAOFファイルのバックアップを作成します。バックアップファイルはマシンの以下の場所に出力されます。

■UNIX

ローカルサイト	/usr/spool/nqs/database/jcdbs.aof-backup-YYYYMMDDhhmmss
クラスタサイト	<JobCenterDBパス>/database/jcdbs.aof-backup-YYYYMMDDhhmmss

■Windows

ローカルサイト	%InstallDirectory%\spool\database\appendonly.aof-backup-YYYYMMDDhhmmss
クラスタサイト	<JobCenterDBパス>\database\appendonly.aof-backup-YYYYMMDDhhmmss

■%InstallDirectory%はJobCenter本体のインストールディレクトリを表します。(既定値はC:\JobCenter\SV)

■「YYYYMMDDhhmmss」はバックアップ作成時の"年月日時分秒"です。

■バックアップファイルは自動削除は行いません。データの巻き戻りによって失われた操作が問題なく再現できた後、手動で削除してください。

JobCenter起動時のAOFファイルのチェック結果はログファイルに出力します。AOFファイルの破損を検知した場合、以下のメッセージをログファイルに出力します。データ復旧による状態の巻き戻りを検知したい場合は、このメッセージを監視してください。

```
AOF is not valid
```

ログファイルはマシンの以下の場所に出力されます。

■UNIX

ローカルサイト	/usr/spool/nqs/log/checkaof.log
クラスタサイト	<JobCenterDBﾊﾟｽ>/log/checkaof.log

■Windows

ローカルサイト	%InstallDirectory%\spool\log\error\checkaof.log
クラスタサイト	<JobCenterDBﾊﾟｽ>\log\error\checkaof.log

■%InstallDirectory%はJobCenter本体のインストールディレクトリを表します。(既定値はC:\JobCenter\SV)

■ログファイルのサイズが1メビバイト（MiB）を超えた場合、バックアップファイル（checkaof.log.bak）に保存し、それ以降のログは新しいファイルに出力します。

4.4. 分析

4.4.1. トラッカ情報の分析

日々のジョブネットワークやジョブの実行量を確認すると、処理のピーク時間帯を確認し、今後のジョブスケジューリングの指針とすることができます。トラッカ情報を分析するにはAnalysis Helperを利用します。

Analysis Helperについて詳しくは次のマニュアルをご覧ください。データの見方、分析の指針についてもこちらにまとめられています。

参照	<Helper機能利用の手引き> 4章 「Analysis Helper」

4.4.2. 実行状況の確認

大量のジョブを同時に転送し、実行しようとする場合、以下のような状態が発生する可能性があります。

■フロー上で、前のジョブが実行完了しているのに、次のジョブがなかなか処理されない（ステータスがWAITからSUBMITに遷移しない）

■キュー内で滞留してしまい、なかなか実行開始されない（ステータスがSUBMITからRUNに遷移しない）

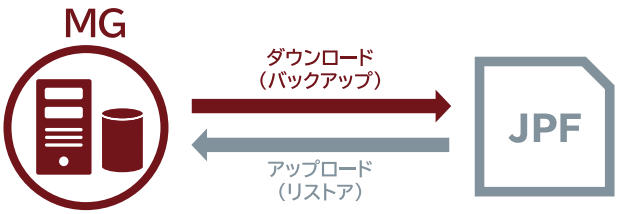
上記はそれぞれボトルネック箇所が異なります。前者はフロー制御の処理負荷、後者はキューの同時実行数を超えるジョブがキューイングされたため起こった滞留が起因しています。これらの状況は、前述のトラッカ情報を分析するだけでは判断ができません。これらを確認するにはjc_perfコマンドを利用して、事前に実行状況のトレースを取得するように設定しておき、状況を確認する必要があります。

参照	<クラシックモード用コマンドリファレンス> 「3.25 jc_perf 実行状況の分析」

4.5. メンテナンス

4.5.1. バックアップ・リストア

万一に備えてデータのバックアップを取得することができます。データはすべてJPF（JobCenter Portable Format）という形式でマシンから取得します。データの種類は大きく分けて次の3種類になります。



■定義情報

提示情報は次の4種類のデータを指します。

- ジョブネットワーク
- スケジュール
- 稼働日カレンダ
- カスタムジョブ定義（カスタムジョブ定義アイコン）

参照	<クラシックモード用基本操作ガイド> 「12.1.1 定義情報のダウンロード」

参照	<クラシックモード用基本操作ガイド> 「12.2 定義情報(JPFファイル)アップロード」

■トラッカ情報

実行中のトラッカ、または、実行完了してアーカイブされたトラッカを指します。リストアはできずバックアップのみができます。バックアップしたデータは専用Viewerで確認することができます。

参照	<クラシックモード用基本操作ガイド> 「12.1.3 トラッカアーカイブ情報のダウンロード」

参照	<クラシックモード用基本操作ガイド> 9章 「トラッカアーカイブのオフライン参照機能」

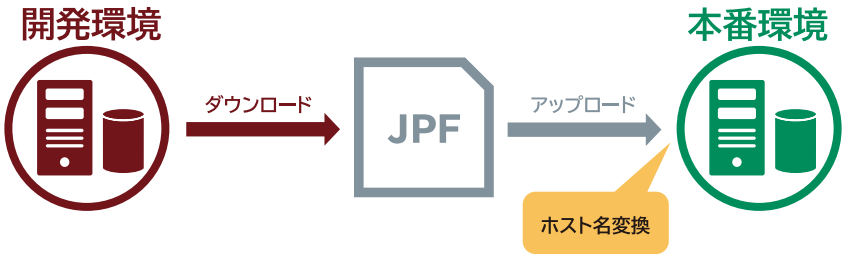
■構成情報

構成情報は、インストール時に決定するマシンIDやセットアップ言語の他、インストール後に設定するマシン一覧やキュー一覧、パーミッション設定などシステム全体にかかわるようなパラメータ・設定のことを指します。

参照	<クラシックモード用環境構築ガイド> 17章 「JobCenter環境のバックアップ・復元」

4.5.2. データの移行

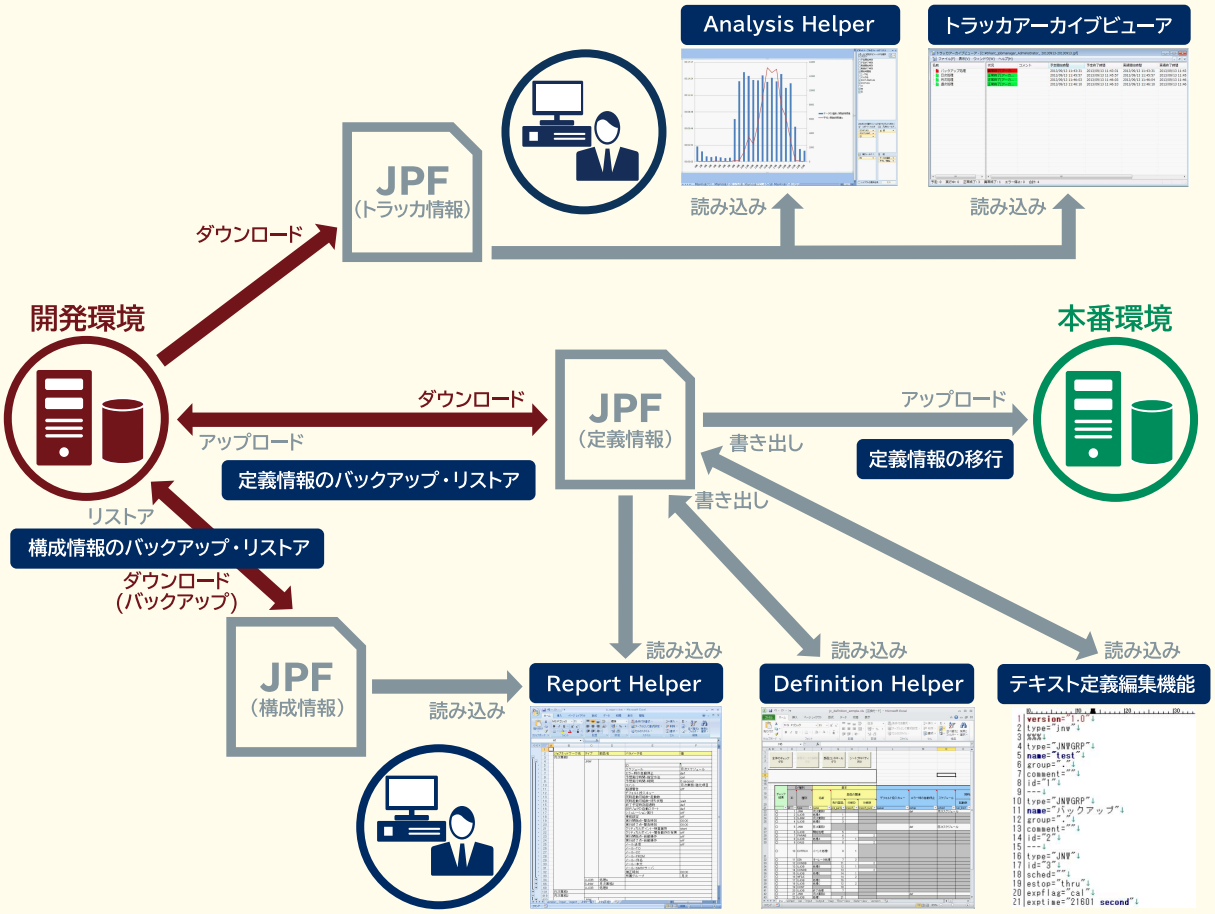
開発環境で作成した定義情報を本番環境へ移行したり、システム更改時に旧システムから新システムにデータを移行したりする場合、移行元のマシンからデータをJPF形式でダウンロードし、それを本番環境にアップロードすることで移行をスムーズに行うことができます。移行可能なデータは定義情報のみでトラッカ情報、構成情報はできません。また、データをアップロードするときは、ホスト名が含まれる一部パラメータを移行先のホスト名に変換することが可能です。



JPFファイルについて

JPF（JobCenter　Portable　Format）は、JobCenter内部のデータを外部ファイルとして扱うための形式です。

バックアップ・リストアやデータ移行の他に次のような機能で利用されます。



5. 逆引きリファレンス

本章では、知りたい内容のキーワード別にどのマニュアルのどこを参照すればよいかについてまとめています。

■Analysis Helper

- Analysis Helperについて知りたい

参照	<Helper機能利用の手引き> 4章 「Analysis Helper」

■Definition Helper

- Definition Helperについて知りたい

参照	<Helper機能利用の手引き> 「2.1 概要」

■NQS

- ネットワークキューイングシステム（NQS）について知りたい

参照	<クラシックモード用基本操作ガイド> 10章 「ネットワークキューイングシステム（NQS）の利用方法」

■Report Helper

- Report Helperについて知りたい

参照	<Helper機能利用の手引き> 「3.1 概要」

■イベント

- イベントの監視と管理をしたい

参照	<クラシックモード用基本操作ガイド> 「11.3 イベント一覧からイベントの監視と管理を行う」

■インストール

- バージョンアップについて知りたい
  - ローカル環境

参照	<クラシックモード用インストールガイド> 5章 「バージョンアップ」

  - クラスタ環境（Windows版）

参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 1章「MSCS(WSFC)環境の場合」
参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 2章 「その他のWindowsクラスタ環境の場合」

- クラスタ環境（UNIX/Linux版）

参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 3章 「UNIX環境の場合」

■ クラスタ環境（Windows）でのバージョンアップ方法を知りたい

参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「1.1 バージョンアップ」
参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「1.2 パッチ適用」
参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「2.1 バージョンアップ」
参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「2.2 パッチ適用」
参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「2.3 ローリングアップデートによるパッチ適用」

■ クラスタ環境（UNIX）でのバージョンアップ方法を知りたい

参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「3.1 バージョンアップ」
参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「3.2 パッチ適用」
参照	<クラシックモード用クラスタ環境でのバージョンアップ・パッチ適用ガイド> 「3.3 ローリングアップデートによるパッチ適用」

■ サイレントインストールをするには

- LicenseManager（Windows版）

参照	<クラシックモード用インストールガイド>　「2.2.5　Windows版　(サイレントインストール)」

- MG/SV（UNIX/Linux版）

参照	<クラシックモード用インストールガイド>　「3.2　JobCenterのセットアップ(サイレントセットアップ)」

- MG/SV（Windows版）

参照	<クラシックモード用インストールガイド>　「2.4.5　Windows版　(サイレントインストール)」

- CL/Win

参照	<クラシックモード用インストールガイド>「2.5.2 サイレントインストール」

■ JobCenterの動作環境を確認するには

参照	<クラシックモード用リリースメモ> 3章 「動作環境」

■ JobCenterのバージョンを確認するには

参照	<クラシックモード用インストールガイド> 6章 「バージョンの確認方法」

■エラー

■ エラーログファイルの設定を変更したい

参照	<クラシックモード用環境構築ガイド>「25.2 エラーログファイルの設定を変更する」

■ 指定された宛先にメールを送信してエラーを通知する方法を知りたい

参照	<クラシックモード用基本操作ガイド> 15章 「エラー発生時のメール送信機能の設定方法」

■ CL/Winのエラーメッセージ一覧を確認したい

参照	<クラシックモード用基本操作ガイド> 20章 「エラーメッセージ一覧」

■ 情報採取時に必要となる情報について知りたい

参照	<クラシックモード用コマンドリファレンス> 7章 「情報採取コマンド」

■環境

- 現在の環境に問題がないかどうかを検査したい

参照	<クラシックモード用コマンドリファレンス> 6章 「設定確認コマンド」

- JobCenterで使用するネットワークプロトコルについて知りたい

参照	<クラシックモード用環境構築ガイド> 「2.1 JobCenterで使用するTCPポート」

- ネットワーク接続を構築する際の注意事項について知りたい

参照	<クラシックモード用環境構築ガイド> 「2.2 ネットワーク接続を構築する際の注意事項」

- Windowsで名前解決ができない場合は

参照	<クラシックモード用環境構築ガイド> 「2.3 Windows環境における名前解決方法」

- 複数のNICを使用したときのJobCenterの設定について

参照	<クラシックモード用環境構築ガイド> 「2.4 UNIX環境における名前解決方法」

- ネットワーク上にある他マシンのマシンIDを登録する方法

参照	<クラシックモード用環境構築ガイド> 「3.1 ネットワーク上にある他マシンのマシンIDを登録する」

- デーモン設定ファイル（daemon.conf）でJobCenter起動時の設定を変更するには

参照	<クラシックモード用環境構築ガイド> 5章 「JobCenter起動時の設定を変更する」

- デーモン設定ファイルの使用可能パラメータについて知りたい

参照	<クラシックモード用環境構築ガイド> 「5.2 デーモン設定ファイルの使用可能パラメータ」

- IPv6環境の設定について知りたい

参照	<クラシックモード用環境構築ガイド> 「5.2 デーモン設定ファイルの使用可能パラメータ」

- 複数のNICを使用したときのdaemon.conf設定について知りたい

参照	<クラシックモード用環境構築ガイド> 「5.3.1 クラスタ構成／複数NIC使用時のdaemon.conf設定について」

- ユーザごとの設定を変更するには

参照	<クラシックモード用環境構築ガイド>「6.2 ユーザ環境の設定を変更する」
参照	<クラシックモード用環境構築ガイド>「6.3 トラッカ表示の設定を変更する」
参照	<クラシックモード用環境構築ガイド>「6.4 アーカイブの設定を変更する」

- システム全体の設定を変更するには

参照	<クラシックモード用環境構築ガイド>「6.5 システム環境の設定を変更する」
参照	<クラシックモード用環境構築ガイド>「6.6 色の設定を変更する」
参照	<クラシックモード用環境構築ガイド>「6.7 操作・実行ログ」

- 文字コード変換の設定について知りたい

参照	<クラシックモード用環境構築ガイド> 9章 「日本語環境での文字コード変換」

- ユーザのアクセス権限を設定したい

参照	<クラシックモード用環境構築ガイド> 10章 「ユーザ権限（パーミッション設定）」

- ホスト名を変更する際に必要な作業について知りたい

参照	<クラシックモード用環境構築ガイド>「14.3 ホスト名を変更する」

- 異なるタイムゾーンのマシンでジョブを実行させたい

参照	<クラシックモード用環境構築ガイド>「16.2 カレンダへのタイムゾーン設定機能」

- JobCenterを稼働させる際に必要なシステムリソースについて知りたい

参照	<クラシックモード用環境構築ガイド> 23章 「システム利用資源」

- ジョブやトラッカの色について知りたい

参照	<クラシックモード用基本操作ガイド>「8.1 ジョブの状態とトラッカの色の関係」

■キュー

- デフォルトで作成されるキューの優先度、多重度の初期設定値が知りたい

参照	<クラシックモード用環境構築ガイド> 「4.1 デフォルトで作成されるキュー」

■クラスタ

- クラスタ構成時の設定について知りたい

参照	<クラシックモード用環境構築ガイド> 「5.3.1 クラスタ構成／複数NIC使用時のdaemon.conf設定について」

- クラスタ環境における利用可能なコマンドを知りたい

参照	<クラシックモード用コマンドリファレンス> 4章 「クラスタ環境のコマンド」

■コマンド

- CL/Winで利用可能なコマンドを知りたい

参照	<クラシックモード用コマンドリファレンス> 2章 「JobCenter CL/Winのコマンド」

- MG/SVで利用可能なコマンドを知りたい

参照	<クラシックモード用コマンドリファレンス> 3章 「JobCenter MG/SVのコマンド」

■ジョブ

- ジョブ実行時の環境変数の取り扱いについて知りたい

参照	<クラシックモード用環境構築ガイド> 15章 「ジョブ実行時の環境変数の取り扱い」

- 部品オブジェクトについて詳しく知りたい

参照	<クラシックモード用基本操作ガイド> 4章 「部品オブジェクトの使用方法」

- カスタムジョブについて詳しく知りたい

参照	<クラシックモード用基本操作ガイド> 5章 「カスタムジョブの利用方法」

- ジョブ実行マシンの稼働状況を知りたい

参照	<クラシックモード用基本操作ガイド> 「11.1 キュー一覧から監視と管理を行う」

- 拡張カスタムジョブ部品について知りたい

参照	<拡張カスタムジョブ部品利用の手引き> 1章 「概要」

■ジョブネットワーク

- ジョブネットワークの作成方法を知りたい

参照	<クラシックモード用基本操作ガイド> 「3.1 ジョブネットワークを作成する」

■スケジュール

- 指定した時刻にジョブネットワークを実行させるには

参照	<クラシックモード用基本操作ガイド> 6章 「スケジュールの操作方法」

■データ移行

- アップロード機能を使用するには

参照	<クラシックモード用基本操作ガイド> 12章 「アップロード・ダウンロード機能」

- ダウンロード機能を使用するには

参照	<クラシックモード用基本操作ガイド> 12章 「アップロード・ダウンロード機能」

- 異なるマシンへユーザ定義データを移行するには

参照	<クラシックモード用環境構築ガイド> 「14.2 異なるマシンへユーザ定義データを移行する」

■トラッカ

- トラッカ表示の設定を変更するには

参照	<クラシックモード用環境構築ガイド> 「6.3 トラッカ表示の設定を変更する」

- MG/SVに接続しなくても、トラッカアーカイブ情報を参照するには

参照	<クラシックモード用基本操作ガイド> 9章 「トラッカアーカイブのオフライン参照機能」

■ネットワーク

- マシンへ接続しているクライアント状況を知りたい

参照	<クラシックモード用基本操作ガイド> 「11.4 クライアント一覧からクライアントの監視を行う」

■ユーザ管理

- ユーザマッピングについて

参照	<クラシックモード用環境構築ガイド>　「3.2　ユーザの関連付けを行う（ユーザマッピング）」

- ユーザに任意のキューの利用許可や禁止をするには

参照	<クラシックモード用環境構築ガイド>「4.4 キューの利用可能ユーザを制限する」

- LDAPを使用したユーザ管理方法を知りたい

参照	<クラシックモード用環境構築ガイド>　11章　「ディレクトリサービス連携によるユーザー管理」

■ログ

- 操作・実行ログの格納パスを知りたい

参照	<操作・実行ログ機能 利用の手引き>「2.1 操作ログ」

- 操作・実行ログの格納パスを知りたい

参照	<操作・実行ログ機能 利用の手引き>「2.2 実行ログ」

- 操作ログのフォーマットについて知りたい

参照	<操作・実行ログ機能 利用の手引き>「4.1 操作ログ」
参照	<操作・実行ログ機能 利用の手引き>「4.2 実行ログ」

- 操作ログフォーマットの過去バージョンとの差異について知りたい

参照	<操作・実行ログ機能 利用の手引き> 6章 「過去バージョンとの差異」

- ログのバックアップファイルについて知りたい

参照	<操作・実行ログ機能 利用の手引き>「3.3 ログのバックアップファイル作成」

発行年月 May 2024
NEC Corporation　2024