HITACHI
Inspire the Next

JP1 Version 12

JP1/Automatic Job Management System 3 設計ガイド（業務設計編）

3021-3-D23-30

前書き

■ 対象製品

適用 OS のバージョン，JP1/Automatic Job Management System 3 が前提とするサービスパックやパッチなどの詳細については，リリースノートで確認してください。

●JP1/Automatic Job Management System 3 - Manager（適用 OS：Windows）

P-2A12-3KCL JP1/Automatic Job Management System 3 - Manager 12-60

製品構成一覧および内訳形名

P-CC2A12-4KCL JP1/Automatic Job Management System 3 - Manager 12-60（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-CC2912-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-CC8212-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-CC2A12-3NCL JP1/Automatic Job Management System 3 - Print Option Manager 12-50（適用 OS：Windows Server 2019，Windows Server 2016，Windows Server 2012）

●JP1/Automatic Job Management System 3 - Manager（適用 OS：HP-UX）

P-1J12-3KCL JP1/Automatic Job Management System 3 - Manager 12-60

製品構成一覧および内訳形名

P-CC1J12-4KCL JP1/Automatic Job Management System 3 - Manager 12-60（適用 OS：HP-UX（IPF））

P-CC2912-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-CC8212-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-CC1J12-3NCL JP1/Automatic Job Management System 3 - Print Option Manager 12-50（適用 OS：HP-UX（IPF））

●JP1/Automatic Job Management System 3 - Manager（適用 OS：Solaris）

P-9D12-3KCL JP1/Automatic Job Management System 3 - Manager 12-60

製品構成一覧および内訳形名

P-CC9D12-4KCL JP1/Automatic Job Management System 3 - Manager 12-60（適用 OS：Solaris 11（SPARC））

P-CC2912-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-CC8212-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-CC9D12-3NCL JP1/Automatic Job Management System 3 - Print Option Manager 12-50（適用 OS：Solaris 11（SPARC））

●JP1/Automatic Job Management System 3 - Manager（適用 OS：AIX）

P-1M12-3KCL JP1/Automatic Job Management System 3 - Manager 12-60

製品構成一覧および内訳形名

P-CC1M12-4KCL JP1/Automatic Job Management System 3 - Manager 12-60（適用 OS：AIX）

P-CC2912-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-CC8212-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-CC1M12-3NCL JP1/Automatic Job Management System 3 - Print Option Manager 12-50（適用 OS：AIX）

●JP1/Automatic Job Management System 3 - Manager（適用 OS：Linux）

P-8112-3KCL JP1/Automatic Job Management System 3 - Manager 12-60

製品構成一覧および内訳形名

P-CC8112-4KCL JP1/Automatic Job Management System 3 - Manager 12-60（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-CC2912-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-CC8212-39CL JP1/Automatic Job Management System 3 - Web Console 12-60（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-CC8112-3NCL JP1/Automatic Job Management System 3 - Print Option Manager 12-50（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

●JP1/Automatic Job Management System 3 - Agent（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-2A12-33CL JP1/Automatic Job Management System 3 - Agent 12-60

●JP1/Automatic Job Management System 3 - Agent（適用 OS：HP-UX（IPF））

P-1J12-33CL JP1/Automatic Job Management System 3 - Agent 12-60

●JP1/Automatic Job Management System 3 - Agent（適用 OS：Solaris 11（SPARC））

P-9D12-33CL JP1/Automatic Job Management System 3 - Agent 12-60

●JP1/Automatic Job Management System 3 - Agent（適用 OS：AIX）

P-1M12-33CL JP1/Automatic Job Management System 3 - Agent 12-60

●JP1/Automatic Job Management System 3 - Agent（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-8112-33CL JP1/Automatic Job Management System 3 - Agent 12-60

●JP1/Automatic Job Management System 3 - Agent Minimal Edition（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-2A12-38CL JP1/Automatic Job Management System 3 - Agent Minimal Edition 12-60

●JP1/Automatic Job Management System 3 - Agent Minimal Edition（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-8112-38CL JP1/Automatic Job Management System 3 - Agent Minimal Edition 12-60

●JP1/Automatic Job Management System 3 - View（適用 OS：Windows）

P-2A12-34CL JP1/Automatic Job Management System 3 - View 12-50

製品構成一覧および内訳形名

P-CC2A12-44CL JP1/Automatic Job Management System 3 - View 12-50（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows 10，Windows 8.1，Windows 8，Windows Server 2012，Windows 7）

P-CC2A12-3MCL JP1/Automatic Job Management System 3 - Print Option 12-50（適用 OS：Windows Server 2019，Windows Server 2016，Windows 10，Windows 8.1，Windows 8，Windows Server 2012，Windows 7）

●JP1/Base（適用 OS：Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows Server 2012）

P-2A2C-6LCL JP1/Base 12-50

●JP1/Base（適用 OS：HP-UX（IPF））

P-1J2C-6LCL JP1/Base 12-50

●JP1/Base（適用 OS：Solaris 11（SPARC））

P-9D2C-6LCL JP1/Base 12-50

●JP1/Base（適用 OS：AIX）

P-1M2C-6LCL JP1/Base 12-50

●JP1/Base（適用 OS：Linux 6.1（x64）以降，Linux 7.1，Oracle Linux 6（x64），Oracle Linux 7，SUSE Linux 12，CentOS 6（x64），CentOS 7）

P-812C-6LCL JP1/Base 12-50

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI，Cosminexus，Groupmax，HiRDB，HITSENSER，JP1，OpenTP1，uCosminexus は，株式会社 日立製作所の商標または登録商標です。

Active Directory 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。

AIX は，世界の多くの国で登録された International Business Machines Corporation の商標です。

Excel 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。

IBM は，世界の多くの国で登録された International Business Machines Corporation の商標です。

Internet Explorer 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。

Itanium は，Intel Corporation またはその子会社の商標です。

Linux は，Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft 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。

MQSeries は，世界の多くの国で登録された International Business Machines Corporation の商標です。

MVS は，世界の多くの国で登録された International Business Machines Corporation の商標です。

Oracle および Java は，オラクルおよびその関連会社の登録商標です。

Outlook 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。

Pentium は，Intel Corporation またはその子会社の商標です。

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

SAP，および本文書に記載されたその他の SAP 製品，サービス，ならびにそれぞれのロゴは，ドイツおよびその他の国々における SAP SE の商標または登録商標です。

SQL Server 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 の商標です。

Windows 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。

Windows Server 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。

すべての SPARC 商標は，米国 SPARC International, Inc. のライセンスを受けて使用している同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す。SPARC 商標がついた製品は，米国 Sun Microsystems, Inc. が開発したアーキテクチャに基づくものです。

その他記載の会社名，製品名などは，それぞれの会社の商標もしくは登録商標です。

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project.

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

This product includes software developed by the University of California, Berkeley and its contributors.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, and Bell Communications Research, Inc (Bellcore).

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The original software is available from ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/

This product includes software developed by Ralf S. Engelschall <rse@engelschall.com> for use in the mod_ssl project (http://www.modssl.org/).

1. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)

2. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)

3. This product includes software written by Tim Hudson (tjh@cryptsoft.com)

4. 本製品には OpenSSL Toolkit ソフトウェアを OpenSSL License および Original SSLeay License に従い使用しています。OpenSSL License および Original SSLeay License は以下のとおりです。

LICENSE ISSUES

==========

The OpenSSL toolkit stays under a double license, i.e. both the conditions of

the OpenSSL License and the original SSLeay license apply to the toolkit.

See below for the actual license texts. Actually both licenses are BSD-style
Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.
OpenSSL License

/*
==========
* Copyright (c) 1998-2018 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.

*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
*
==========
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/
Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.

*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the rouines from the library
* being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

This product includes software developed by the Java Apache Project for use in the Apache JServ servlet engine project (http://java.apache.org/).

This product includes software developed by IAIK of Graz University of Technology.

This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (http://relaxngcc.sf.net/).

This product includes software developed by Andy Clark.

Java is a registered trademark of Oracle and/or its affiliates.

■ マイクロソフト製品の表記について

このマニュアルでは，マイクロソフト製品の名称を次のように表記しています。

表記		製品名
Excel		Microsoft(R) Excel
		Microsoft(R) Office Excel
Exchange Server		Microsoft(R) Exchange 2000 Enterprise Server
		Microsoft(R) Exchange 2000 Server
		Microsoft(R) Exchange Server
IE	Internet Explorer	Microsoft(R) Internet Explorer(R)
		Windows(R) Internet Explorer(R)
Microsoft SQL Server		Microsoft(R) SQL Server
		Microsoft(R) SQL Server Enterprise Edition
MSMQ		Microsoft(R) Message Queue Server
Outlook	Outlook 2010	Microsoft(R) Office Outlook(R) 2010
	Outlook 2013	Microsoft(R) Office Outlook(R) 2013
	Outlook 2016	Microsoft(R) Office Outlook(R) 2016
	Outlook 2019	Microsoft(R) Office Outlook(R) 2019
Outlook Express		Microsoft(R) Outlook(R) Express
Windows 10		Windows(R) 10 Enterprise
		Windows(R) 10 Pro
		Windows(R) 10 Home
Windows 7		Microsoft(R) Windows(R) 7 Enterprise
		Microsoft(R) Windows(R) 7 Professional
		Microsoft(R) Windows(R) 7 Ultimate

表記		製品名
Windows 8		Windows(R) 8 Enterprise
		Windows(R) 8 Pro
Windows 8.1		Windows(R) 8.1 Enterprise
		Windows(R) 8.1 Pro
Windows Server 2012	Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Datacenter
		Microsoft(R) Windows Server(R) 2012 Standard
	Windows Server 2012 R2	Microsoft(R) Windows Server(R) 2012 R2 Datacenter
		Microsoft(R) Windows Server(R) 2012 R2 Standard
Windows Server 2016		Microsoft(R) Windows Server(R) 2016 Datacenter
		Microsoft(R) Windows Server(R) 2016 Standard
Windows Server 2019		Microsoft(R) Windows Server(R) 2019 Datacenter
		Microsoft(R) Windows Server(R) 2019 Standard
Windows Server 2022		Microsoft(R) Windows Server(R) 2022 Datacenter
		Microsoft(R) Windows Server(R) 2022 Standard

Windows Server 2022，Windows Server 2019，Windows Server 2016，Windows 10，Windows 8.1，Windows 8，Windows Server 2012，および Windows 7 を総称して Windows と表記することがあります。

■ 発行

2022 年 1 月 3021-3-D23-30

■ 著作権

Copyright (C) 2019, 2022, Hitachi, Ltd.
Copyright (C) 2019, 2022, Hitachi Solutions, Ltd.

変更内容

変更内容については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド」の変更内容を参照してください。

はじめに

このマニュアルは，JP1/Automatic Job Management System 3（以降，JP1/AJS3 と省略します）の業務設計について説明したものです。JP1/AJS3 の機能を解説した，マニュアル「JP1/Automatic Job Management System 3 導入ガイド」の内容を踏まえた上でご利用ください。

なお，このマニュアルをお読みになる際の前提情報については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド」のはじめにの説明を参照してください。

■ マニュアルの構成

このマニュアルは，次に示す章から構成されています。

なお，このマニュアルは各 OS（Operating System）に共通のマニュアルです。OS ごとに差異がある場合は，本文中でそのつど内容を書き分けています。

第 1 章　業務設計の概要

JP1/AJS3 で業務を自動化するための設計の流れや検討項目などの概要について説明しています。

第 2 章　ジョブの定義内容と実行順序の検討

JP1/AJS3 で業務の自動化を実現するために必要な，ジョブやジョブネットの構築ための検討項目について説明しています。

第 3 章　運用カレンダーと実行スケジュールの検討

JP1/AJS3 運用上のカレンダーや実行スケジュールを立てるための検討項目について説明しています。

第 4 章　実行登録方法の検討

JP1/AJS3 にどのような方法でジョブネットを実行登録するかについて説明しています。

第 5 章　監視方法の検討

JP1/AJS - View を使ってジョブネットを監視するための検討項目について説明しています。

第 6 章　アクセス権の検討

ジョブネットへのアクセス権の検討や，JP1 ユーザーと OS ユーザーの対応づけの検討について説明しています。

第 7 章　業務設計時の注意事項

業務設計での注意事項をまとめて説明しています。

第 8 章　定義内容の事前チェック

実際に JP1/AJS3 での運用に入る前の，定義内容チェックの流れやチェック項目，注意事項などについて説明しています。

目次

前書き　2
変更内容　13
はじめに　14

1　業務設計の概要　20
1.1　設計の流れ　21
1.2　業務の自動化　22
1.3　業務の自動化の流れ　23
1.4　業務の自動化のための検討項目　24

2　ジョブの定義内容と実行順序の検討　25
2.1　ジョブの構築のための検討　26
2.1.1　自動化する業務を洗い出す　26
2.1.2　実行ファイルを設計する　26
2.1.3　ジョブの種類を検討する　29
2.1.4　ジョブの定義内容を検討する　33
2.2　ジョブネットの構築のための検討　37
2.2.1　ジョブの実行順序を検討する　37
2.2.2　ジョブの実行に失敗したときの対処方法を検討する　45
2.2.3　ジョブネットの定義内容を検討する　46
2.2.4　ジョブネットコネクタを使用したルートジョブネットの実行順序制御　48
2.2.5　待ち合わせ条件を使用したユニットの実行順序制御　67
2.2.6　マクロ変数の使用を検討する　121
2.3　業務の自動化検討時のヒント　131
2.3.1　負荷を分散させた処理　131
2.3.2　業務の自動化を実現する JP1/AJS3 の機能　139
2.4　さまざまなジョブネット定義　142
2.4.1　ファイルを指定して処理を実行する（標準ジョブを使ったジョブネットの定義例）　142
2.4.2　複数の条件のうち一つが成立したら処理を実行する（OR ジョブを使ったジョブネットの定義例）143
2.4.3　先行ジョブの結果でそのあとの処理を動的に変える（判定ジョブを使ったジョブネットの定義例）144
2.4.4　事象の発生を契機に処理を実行する（イベントジョブを使ったジョブネットの定義例）　151
2.4.5　先行ジョブの終了や事象の発生を契機に JP1 イベントを送信する（JP1 イベント送信ジョブを使ったジョブネットの定義例）　168
2.4.6　ジョブが異常終了したときに特定の処理を実行する（リカバリーユニットを使ったジョブネットの定義例）　170

2.4.7 ルートジョブネットの実行順序を制御する（ジョブネットコネクタを使ったジョブネットの定義例） 173
2.4.8 異なるジョブネットにあるユニット同士の実行順序を制御する（待ち合わせ条件を使ったジョブネットの定義例） 174
2.4.9 動的に変わる情報を後続ユニットに引き継ぐ（引き継ぎ情報設定ジョブを使ったジョブネットの定義例） 177
2.4.10 クラウド環境でジョブを実行する（フレキシブルジョブを使ったジョブネットの定義例） 188
2.4.11 一つのジョブを複数の実行エージェントで一斉に実行する（フレキシブルジョブを使ったジョブネットの定義例） 191
2.4.12 Web 上の業務システムと連携する（HTTP 接続ジョブを使ったジョブネットの定義例） 193
2.4.13 ジョブの異常終了時に自動でリトライする 198

3 運用カレンダーと実行スケジュールの検討 215

3.1 カレンダーやスケジュールの検討の流れ 216
3.2 JP1/AJS3 運用上のカレンダーの検討 217
3.3 ジョブネットの実行スケジュールの検討 218
3.4 起動条件の検討 219
3.4.1 起動条件の検討内容 219
3.5 さまざまなスケジュール設定 222
3.5.1 2 日にまたがる業務のスケジュールを立てる 222
3.5.2 実行開始日時を複数設定する 225
3.5.3 ジョブネットの一部のジョブにほかのジョブと異なるスケジュールを設定する 226
3.5.4 1 日のうちに同じジョブネットを何回か定期的に実行する（サイクルジョブ定義） 232
3.5.5 算出したスケジュールを基に，実行予定日を前後にずらす（起算スケジュール） 233
3.5.6 月の上旬，中旬，下旬を意識したスケジュールを設定する 236
3.5.7 業務ごとにカレンダーを定義する 237

4 実行登録方法の検討 243

4.1 ジョブネットの実行登録方法 244
4.1.1 計画実行登録 244
4.1.2 確定実行登録 245
4.1.3 即時実行登録 246

5 監視方法の検討 247

5.1 JP1/AJS3 - View での監視方法の検討 248
5.1.1 サマリー監視画面での監視対象を検討する 248
5.1.2 ジョブネットの遅延監視について検討する 248
5.1.3 ジョブの実行所要時間による終了遅延監視 249
5.2 Web GUI での監視方法の検討 251
5.2.1 ［ダッシュボード］画面での監視対象を検討する 251
5.3 API を利用したユーザーアプリケーションでの監視方法の検討 252

6 アクセス権の検討 253

6.1 ジョブネットへのアクセス権を検討する流れ 254
6.2 アクセス権を設定する範囲についての検討 255
6.3 登録するユーザーの検討 257
6.4 設定するアクセス権限の検討 259
6.4.1 アクセス権限についての検討項目 259
6.5 対応づけるユーザーの検討 284
6.5.1 対応づけるユーザーの検討項目 284

7 業務設計時の注意事項 289

7.1 ルートジョブネットの実行登録数に関する注意事項 290
7.2 保存世代数と性能との関係 292
7.3 PC ジョブ使用時の注意事項 294
7.4 UNIX ジョブ使用時の注意事項 297
7.5 リカバリーユニット使用時の注意事項 300
7.6 イベントジョブ使用時の注意事項 302
7.6.1 JP1 イベント受信監視ジョブの注意事項 304
7.6.2 ファイル監視ジョブの注意事項 310
7.6.3 メール受信監視ジョブの注意事項 330
7.6.4 ログファイル監視ジョブの注意事項 331
7.6.5 Windows イベントログ監視ジョブの注意事項 334
7.6.6 実行間隔制御ジョブの注意事項 336
7.6.7 引き継ぎ情報定義時の注意事項 336
7.6.8 イベントジョブを実行したままサービスを再起動する場合の注意事項 338
7.6.9 JP1/AJS3 が発行するイベントやメッセージを監視する 340
7.7 アクションジョブ使用時の注意事項 347
7.7.1 JP1 イベント送信ジョブの注意事項 347
7.7.2 メール送信ジョブの注意事項 348
7.7.3 JP1/Cm2 状態通知ジョブの注意事項 349
7.7.4 ローカル電源制御ジョブおよびリモート電源制御ジョブの注意事項 349
7.8 フレキシブルジョブ使用時の注意事項 352
7.9 HTTP 接続ジョブ使用時の注意事項 358
7.10 ジョブ定義についての注意事項 361
7.10.1 標準出力ファイル・標準エラー出力ファイルの注意事項 361
7.10.2 ジョブの実行優先順位に関する注意事項 366
7.10.3 ジョブの終了コードを確認する 370
7.11 スケジューラーサービスの起動性能について 375

8　定義内容の事前チェック　376
8.1　本番運用前の JP1/AJS3 定義内容のチェック　377
8.1.1　定義内容の事前チェック機能の概要と注意事項　378

付録　386
付録 A　イベントジョブおよびアクションジョブ実行時の戻り値一覧　387
付録 B　イベントジョブで引き継ぐ情報　394
付録 C　HTTP 接続ジョブで使用するファイル　399
付録 C.1　接続設定ファイル　399
付録 C.2　送信情報ファイル　405
付録 C.3　送信情報ファイル（URL パラメーター）　406
付録 C.4　送信情報ファイル（メッセージボディ）　407
付録 C.5　ステータス格納ファイル　407
付録 C.6　受信ヘッダー格納ファイル　408
付録 C.7　受信ボディ格納ファイル　409
付録 D　各バージョンの変更内容　411
付録 E　このマニュアルの参考情報　412
付録 F　用語解説　413

索引　414

1 業務設計の概要

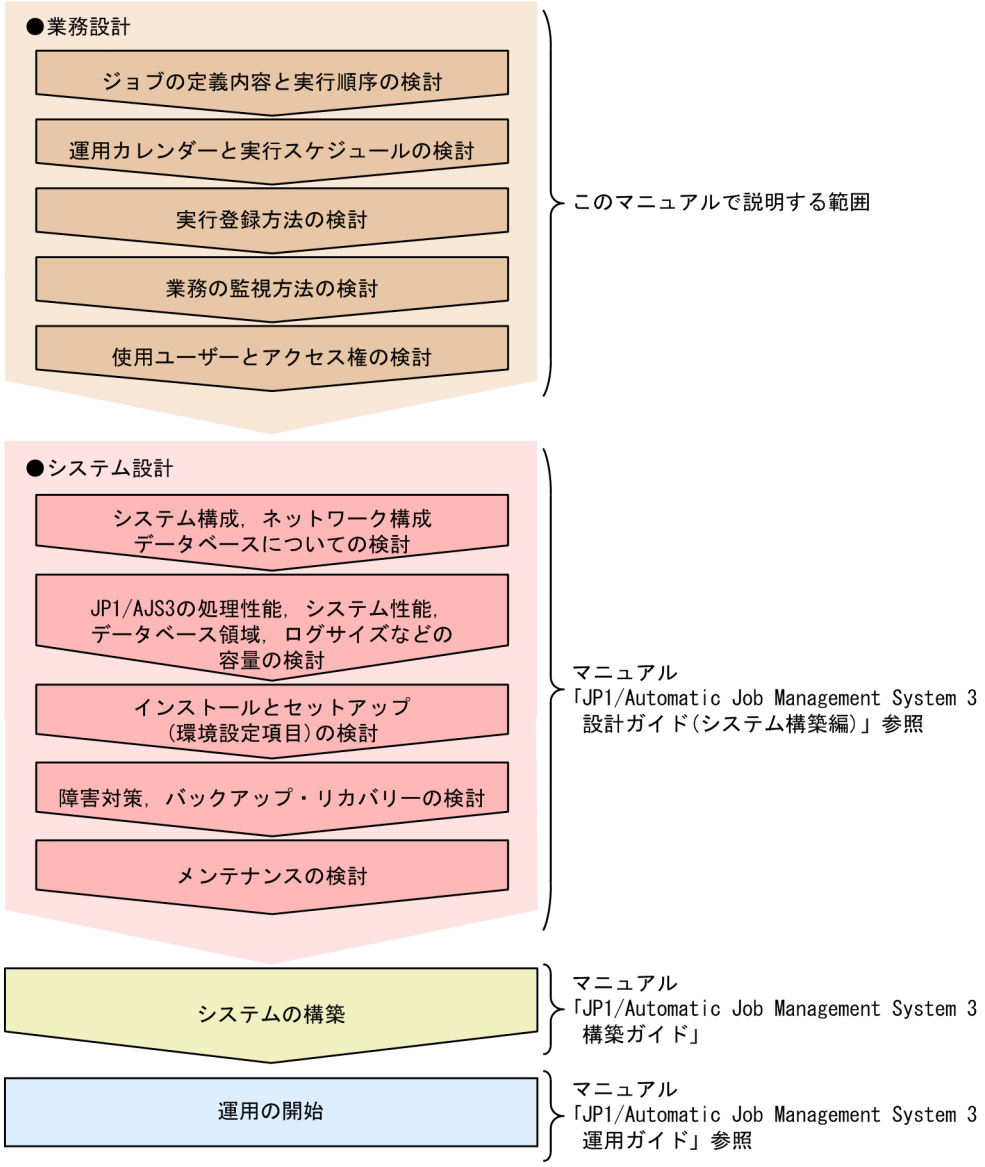
この章では，JP1/AJS3 で業務を自動化するための設計の概要を説明します。

1.1　設計の流れ

JP1/AJS3 導入時の設計は，JP1/AJS3 で業務を自動化するための業務設計と，JP1/AJS3 を効率良く導入・運用するためのシステム設計に大別されます。

JP1/AJS3 導入時の基本的な設計作業の流れを，次の図に示します。

図 1-1　JP1/AJS3 の設計作業の流れ



注　状況によっては，順序が前後します。

このマニュアルでは，業務を自動化するための検討項目について説明しています。

システムに関する設計については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編）」を，JP1/AJS3 運用上のノウハウなどについては，マニュアル「JP1/Automatic Job Management System 3 運用ガイド」を参照してください。

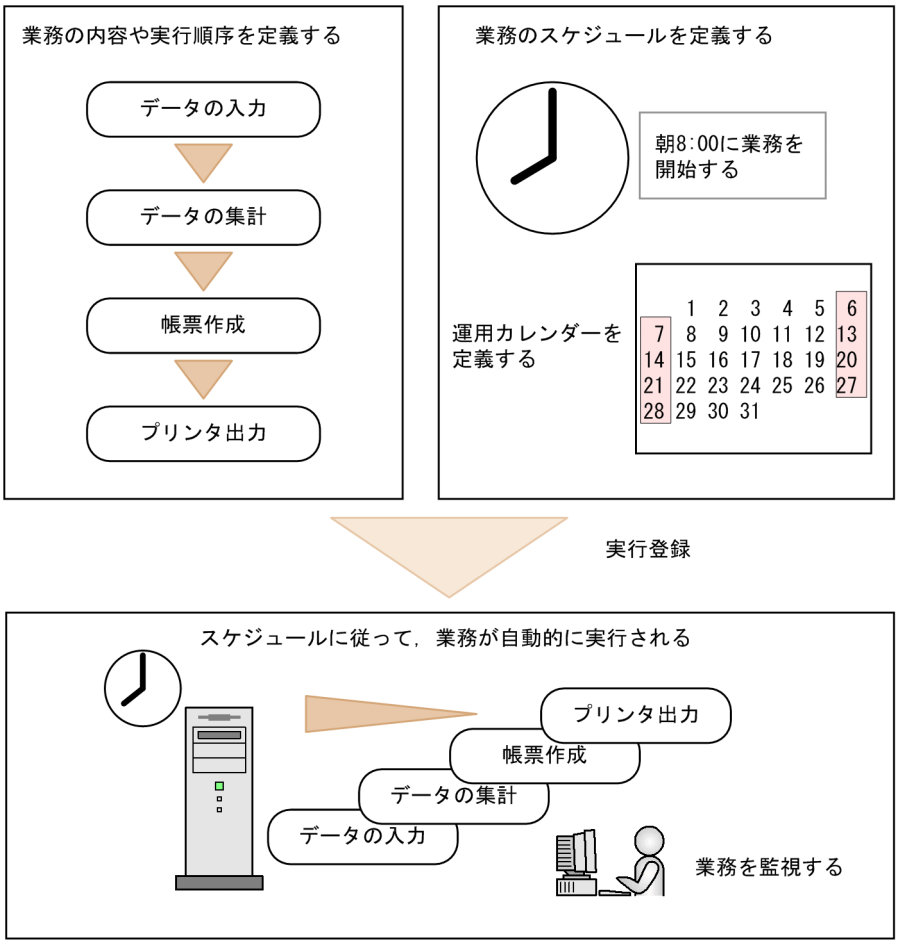
1.2　業務の自動化

JP1/AJS3 を使って業務を自動化するには，次のことを定義する必要があります。

- 業務の内容や実行順序
- 業務の運用カレンダーや実行スケジュール，業務の契機となる事象など

業務の自動化の概要を，次の図に示します。

図 1-2　業務の自動化の概要



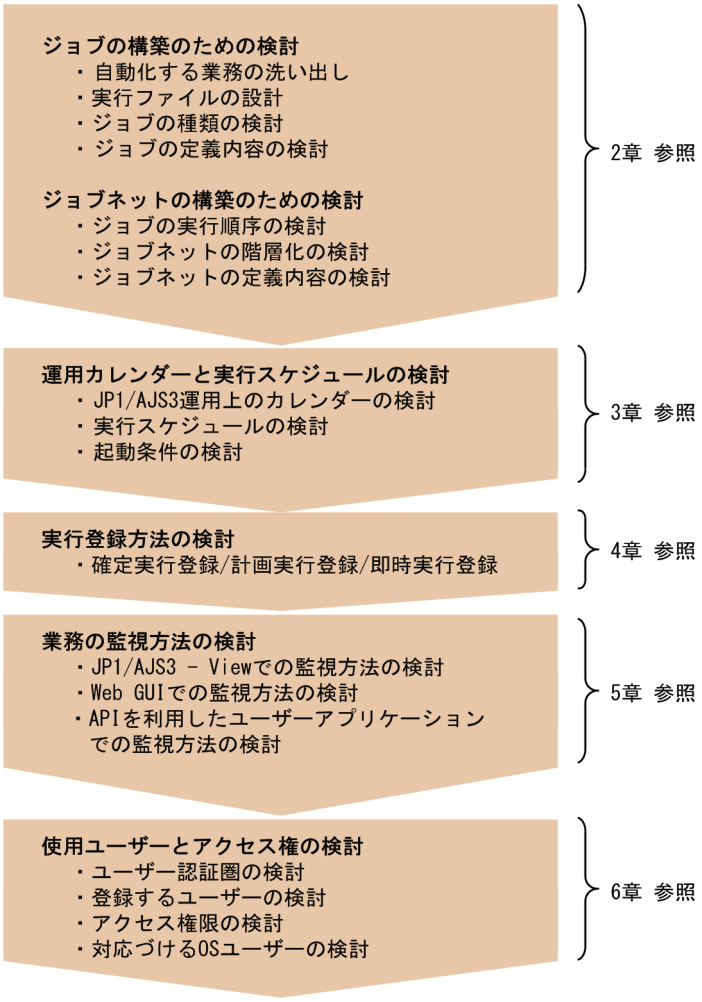
JP1/AJS3 で実際に業務を定義するに当たって，これらについて十分に検討する必要があります。

1.3　業務の自動化の流れ

JP1/AJS3 では，コマンドやアプリケーションプログラム，シェルスクリプトなどが自動化する業務を構成する一つ一つの処理単位となります。この一つ一つの処理を**ジョブ**として定義し，それらのジョブに実行順序を付けて**ジョブネット**という一つの業務単位にまとめます。作成したジョブネットには**スケジュールルール**を定義し，実行日時が自動的に決まるようにします。

JP1/AJS3 での，業務の自動化の流れを次の図に示します。

図 1-3　業務の自動化の流れ



1.4　業務の自動化のための検討項目

業務の自動化のための検討項目と記載個所の一覧を次の表に示します。

表 1-1　業務の自動化のための検討項目と記載個所

項番	検討項目	記載個所
1	ジョブの構築のための検討	「2.1 ジョブの構築のための検討」
2	ジョブネットの構築のための検討	「2.2 ジョブネットの構築のための検討」
3	運用カレンダーと実行スケジュールの検討	「3. 運用カレンダーと実行スケジュールの検討」
4	実行登録方法の検討	「4. 実行登録方法の検討」
5	業務の監視方法の検討	「5. 監視方法の検討」
6	使用ユーザーとアクセス権の検討	「6. アクセス権の検討」
7	業務設計時の注意事項	「7. 業務設計時の注意事項」

JP1/AJS3 の機能については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド」を参照してください。

2 ジョブの定義内容と実行順序の検討

JP1/AJS3 では，自動化したい業務をジョブやジョブネットとして定義し，スケジュールや処理を開始する条件などを設定します。

この章では，JP1/AJS3 で業務の自動化を実現するために必要な，ジョブやジョブネットの構築ための検討項目について説明します。

2.1　ジョブの構築のための検討

自動化する業務の一つ一つの細かい処理は，ジョブと呼ばれるユニットで定義します。ジョブの実体は，コマンドやアプリケーションプログラム，シェルスクリプトなどによる処理であり，作業の最小単位となります。業務を自動化するために必要な処理を洗い出し，それらを処理する実行プログラムの設計などについて検討します。

2.1.1　自動化する業務を洗い出す

JP1/AJS3 で自動化する業務を詳細に洗い出します。JP1/AJS3 では，日次，月次などのように決まった周期で決まった処理を実行する定型業務ばかりでなく，先行する処理の結果によってそのあとの処理が動的に変わる業務なども自動化できます。また，実行日時が不定期な業務でも，ファイルの更新やイベントの受信など，さまざまな事象を契機に自動的に実行できます。

JP1/AJS3 で自動化する業務を検討するためには，JP1/AJS3 の機能を理解する必要があります。JP1/AJS3 の機能については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド」を参照してください。

また，JP1/AJS3 で業務の自動化を実現するための代表的な機能を，業務の自動化検討時のヒントとして紹介しています。業務の自動化を実現する JP1/AJS3 の機能の紹介については，「2.3.2 業務の自動化を実現する JP1/AJS3 の機能」を参照してください。

2.1.2　実行ファイルを設計する

JP1/AJS3 では，バッチファイルやシェルスクリプトなどの**実行ファイル**によって業務を処理します。業務を処理する実行ファイルの設計について検討します。

実行ファイルの設計に当たって検討する際のポイントを次に示します。

(1)　JP1/AJS3 で使用できる形式とする

JP1/AJS3 で使用できない形式の実行ファイルでの処理は，自動化の対象外です。JP1/AJS3 で使用できる実行ファイルの種類を次の表に示します。

表 2–1　JP1/AJS3 で使用できる実行ファイルの種類

実行するホスト	使用できる実行ファイル
Windows ホスト	• .exe ファイル • .com ファイル • .cmd ファイル • バッチファイル（.bat） • JP1/Script で作成したスクリプトファイル（.spt）

実行するホスト	使用できる実行ファイル
Windows ホスト	• アプリケーションに関連づけられているファイルタイプ（拡張子）を持つデータファイル
UNIX ホスト	• シェルスクリプト • 実行ファイル

(2) 一つの実行ファイルには一つの命令を定義する

一つの実行ファイルには，一つの命令を定義することを推奨します。

処理を分けておくことで，どの処理まで正常に終了しているのか，どの処理がうまくいかなかったのかがわかります。また，業務が途中で異常終了した場合にどの処理で異常が起きたのかがわかれば，処理を再実行する場合などに異常終了した処理からあとの処理を再実行したり，異常終了した処理だけを再実行したりできます。

(3) 画面やメッセージによる応答が発生しない処理にする

JP1/AJS3 で実行する実行ファイルは，バックグラウンドで実行されます。そのため，画面やメッセージが表示されて応答待ちの状態にならない処理にしてください。ただし，例えばメッセージダイアログボックスが表示されても，自動的に［はい］ボタンを選択して進むような処理であれば，JP1/AJS3 でも使用できます。

(4) リターンコードを出力する処理にする

JP1/AJS3 では，処理が正常に終了したのか失敗したのかを，実行ファイルのリターンコードによって判定します。したがって，処理結果に応じたリターンコードが出力されるように実行ファイルを作成してください。

バッチファイルによる処理の場合

バッチファイルによる処理の場合，jp1exec コマンドやjp1exit コマンドを使用して，バッチファイル内の実行プログラムの処理結果に応じたリターンコードが出力されるようにしてください。また，コマンドが異常終了した場合に原因を分析できるように，要因となったリターンコードがそのまま出力されるようにしてください。

jp1exit コマンドやjp1exec コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド」を参照してください。

シェルスクリプトによる処理の場合

シェルスクリプトによる処理の場合，次に示す例を参考にして，処理結果に応じたリターンコードが出力されるようにしてください。

```
   :
RC=$?
exit△$RC
```

また，出力されるリターンコードのどの範囲を警告終了，異常終了とするか，というしきい値を決めておいてください。

(5) 実行時間が短い処理にする

一つの処理の実行時間が長くなり過ぎると，正常に動作しているのか，トラブルなどによって停止しているのか，わからなくなるおそれがあります。一つの処理の実行時間は，最大で 2 時間程度を目安としてください。

(6) 命名規則を設ける

実行ファイルや，実行ファイルを格納するフォルダには，命名規則を決めておくことを推奨します。JP1/AJS3 の定義で使用する業務や処理などの名称とあわせて決めておくと，業務全体が管理しやすくなります。業務名，処理名，処理サイクル，実行場所などを意味する識別子を決めておくと便利です。

JP1/AJS3 で使用する業務や処理の名称には，半角英数字，全角文字，および次の記号を使用できます。

```
! # $ % + @ -（ハイフン） .（ピリオド） _（アンダーバー）
```

ただし，記号は UNIX のシェルなどのコマンドインタープリターで特別な意味を持つ場合があり，業務や処理の名称に記号を使うと誤動作することがあります。そのため，記号はできるだけ使用しないでください。

なお，次の文字および記号は使用しないでください。

- （業務や処理の名称の先頭で）「.（ピリオド）」および「@」
- 機種依存文字

(7) 処理を実行したいホストに実行ファイルを保存する

実行ファイル（コマンドやバッチファイルなど）は，処理を実行するホストに保存してください。

(8) その他のポイント

- 実行ファイルが使用する入出力ファイルやコマンドに指定するパスは，フォルダ名を含めた完全パスで入力してください。
- 異常終了時の原因を調査しやすくするために，echo コマンドなどを使用して，任意のメッセージを標準エラー出力ファイルに出力させるようにしてください。原因，対処方法，業務名，再実行のポイントなどを表示させておくと便利です。出力されたメッセージは，JP1/AJS3 - View の［実行結果詳細］ダイアログボックスで参照できます。［実行結果詳細］ダイアログボックスについては，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12.3.39 ［実行結果詳細］ダイアログボックス」を参照してください。
- 環境変数を使用する場合，原則として実行ファイルに環境変数を定義してください。

- シェルスクリプトの場合，使用するシェルをファイルの先頭で宣言してください。使用するシェルについては，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編）2.5.3 ジョブ実行時の OS ユーザー環境の検討」も参照してください。

2.1.3　ジョブの種類を検討する

ジョブにはさまざまな種類があり，処理の形態に合わせて定義するジョブを選べます。ジョブの種類には，次のものがあります。

- 標準ジョブ
- OR ジョブ
- 判定ジョブ
- イベントジョブ
- アクションジョブ
- カスタムジョブ
- 引き継ぎ情報設定ジョブ
- HTTP 接続ジョブ

それぞれのジョブの特性について，次に説明します。

(1)　標準ジョブ

実行ファイルを指定して処理を実行させるジョブです。それぞれのジョブの内容と指定できる実行ファイルを次の表に示します。

表 2-2　標準ジョブの内容と指定できる実行ファイル

項番	ジョブの種類	ジョブの内容	指定できる実行ファイル
1	UNIX ジョブ	UNIX ホストで処理を実行させる。	• 実行ファイル • シェルスクリプト
2	PC ジョブ	Windows ホストで処理を実行させる。	• .exe ファイル • .com ファイル • .cmd ファイル • .bat ファイル • .spt ファイル※（JP1/Script で作成したスクリプトファイル） • アプリケーションに関連づけられているファイルタイプ（拡張子）を持つデータファイル
3	QUEUE ジョブ	特定のキューにジョブを送って処理を実行させる。	• 実行ファイル • シェルスクリプト

項番	ジョブの種類	ジョブの内容	指定できる実行ファイル
3	QUEUE ジョブ	JP1/NQSEXEC や JP1/OJE など，他システムと連携する場合に使用する。	• .exe ファイル • .com ファイル • .cmd ファイル • .bat ファイル • .spt ファイル※（JP1/Script で作成したスクリプトファイル） • アプリケーションに関連づけられているファイルタイプ（拡張子）を持つデータファイル
4	フレキシブルジョブ	• マネージャーホストが直接管理していないエージェントホストで，処理を実行させる。 • 一つのジョブを複数の実行エージェントに一斉に配信する。 • クラウド環境で処理を実行させる。ロードバランサーにも対応している。	• 実行ファイル • シェルスクリプト • .exe ファイル • .com ファイル • .cmd ファイル • .bat ファイル

注※

.spt ファイルを指定した場合，実行先ホスト側にも JP1/Script がインストールされていないと実行されません。

補足事項

ホスト名や IP アドレスが固定されていて，マネージャーホストでエージェントホストを管理できる場合は，フレキシブルジョブではなく，PC ジョブや UNIX ジョブの使用を推奨します。

(2) OR ジョブ

事象の発生を監視するジョブ（イベントジョブ）を先行ジョブとして複数定義しておき，それらが監視する事象が一つでも発生した場合に後続ジョブを実行させるジョブです。OR ジョブの先行ジョブには，イベントジョブを定義します。

詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.1.1(1)(b) OR ジョブ」を参照してください。

(3) 判定ジョブ

実行する条件に合致しているか，していないかを判定するジョブです。

判定ジョブによる判定条件には，次の三つがあります。

- 先行ジョブの終了コードによる判定
- ファイルの有無による判定
- 変数比較による判定

詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.1.1(1)(c) 判定ジョブ」を参照してください。

(4)　イベントジョブ

事象（イベント）の発生を監視するジョブです。ジョブフローやジョブネットの起動条件などにイベントジョブを定義することによって，事象の発生を契機にジョブやジョブネットを実行させることができます。

それぞれのジョブの内容を次の表に示します。

表 2-3　イベントジョブの種類

項番	イベントジョブ名	ジョブの内容
1	JP1 イベント受信監視ジョブ	JP1/Base から特定のイベントを受け取ったときにイベントジョブが終了する。
2	ファイル監視ジョブ	特定のファイルが作成，削除，または更新されたときにイベントジョブが終了する。
3	メール受信監視ジョブ	特定のメールを受信したときにイベントジョブが終了する。
4	ログファイル監視ジョブ	JP1/Base のログファイルトラップ機能と連携し，指定したログファイルに，特定の情報が書き込まれたときにイベントジョブが終了する。
5	Windows イベントログ監視ジョブ	JP1/Base のイベントログトラップ機能と連携し，Windows イベントログファイルに特定の情報が書き込まれたときにイベントジョブが終了する。
6	実行間隔制御ジョブ	指定した時間が経過したらイベントジョブが終了する。
7	メッセージキュー受信監視ジョブ※	TP1/Message Queue または MQSeries から特定のメッセージを受信したときにイベントジョブが終了する。
8	MSMQ 受信監視ジョブ	MSMQ から特定のメッセージを受信したときにイベントジョブが終了する。

注

メール受信監視ジョブ，メッセージキュー受信監視ジョブ，および MSMQ 受信監視ジョブは，JP1/AJS3 とそれぞれのプログラムとの連携が必要です。プログラムとの連携については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド」を参照してください。

注※

Linux は対象外です。

イベントジョブの動作は，JP1 ユーザーの権限およびジョブに定義されている権限（所有者，JP1 資源グループ，および実行ユーザー種別）には依存しません。Windows の場合は，JP1/AJS3 サービスのアカウント権限に依存するため，JP1/AJS3 サービスにあらかじめ権限を設定しておく必要があります。

(5) アクションジョブ

特定の処理を実行するジョブです。イベントジョブと組み合わせることによって，事象の発生を契機に JP1 イベントやメールを送信したり，状態を通知したりなどの処理（アクション）を実行させたりできます。

それぞれのジョブの内容を次の表に示します。

表 2-4 アクションジョブの種類

項番	アクションジョブ名	ジョブの内容
1	JP1 イベント送信ジョブ	JP1 イベントを JP1/Base のイベントサービスに登録する。
2	メール送信ジョブ	メールを送信する。
3	JP1/Cm2 状態通知ジョブ	JP1/Cm2/NNM または HP NNM に状態を通知する。
4	ローカル電源制御ジョブ	JP1/Power Monitor と連携して，ローカル電源制御ジョブを実行したホストをシャットダウンする。
5	リモート電源制御ジョブ	JP1/Power Monitor と連携して，ネットワーク上のホストの電源を投入したり，システムを終了したりする。
6	メッセージキュー送信ジョブ※	TP1/Message Queue または MQSeries のメッセージを送信する。
7	MSMQ 送信ジョブ	MSMQ のメッセージを送信する。

注

メール送信ジョブ，メッセージキュー送信ジョブ，MSMQ 送信ジョブ，JP1/Cm2 状態通知ジョブ，ローカル電源制御ジョブ，およびリモート電源制御ジョブは，JP1/AJS3 とそれぞれのプログラムとの連携が必要です。プログラムとの連携については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド」を参照してください。

注※

Linux は対象外です。

(6) カスタムジョブ

JP1/AJS3 以外のプログラムが JP1/AJS3 と連携して処理を実行するジョブです。

詳細については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド　4. カスタムジョブ」を参照してください。

(7) 引き継ぎ情報設定ジョブ

先行ジョブが出力した標準出力ファイルから必要な情報を切り出して，後続ジョブに引き継ぐジョブです。動的に変化する情報を標準出力ファイルに出力すれば，引き継ぎ情報設定ジョブでその情報を切り出し，後続ジョブで使用することができます。

詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.1.1(1)(g) 引き継ぎ情報設定ジョブ」を参照してください。

(8)　HTTP 接続ジョブ

HTTP 接続ジョブは，HTTP プロトコルでリクエストの送信およびレスポンスの受信ができるジョブです。クラウド上や Web サーバ上で公開されている Web API（例えば REST API）を，JP1/AJS3 で呼び出すことができます。HTTP 接続ジョブを使用すると，JP1/AJS3 で構築された業務システムと Web 上の業務システムとを連携させることができます。

詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.1.1(1)(h) HTTP 接続ジョブ」を参照してください。

2.1.4　ジョブの定義内容を検討する

ジョブに定義する内容について検討します。

ジョブに定義する詳細な項目については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12. ウィンドウとダイアログボックス」にある各ジョブの詳細定義ダイアログボックスの説明を参照してください。

(1)　実行エージェント

ジョブの実行先となる実行エージェント名を指定します。実行エージェント制限を使用している場合は，実行エージェントプロファイルの一覧から選択できます。実行エージェントは，ジョブの実行先となるエージェントホストの論理的な名称です。マネージャーに定義した実行エージェント情報に従って，ジョブに指定した実行エージェントとエージェントホストの物理的なホスト名がマッピングされ，ジョブが配信されます。

ジョブ実行環境が実行エージェントグループによって負荷分散させる構成の場合は，実行エージェントグループを指定すると，グループ化された実行エージェントの優先順位に従ってジョブを配信します。

実行エージェントを指定できるジョブは次のものです。

- PC ジョブ
- UNIX ジョブ
- イベントジョブ※
- アクションジョブ
- カスタムジョブ
- HTTP 接続ジョブ

注※

イベントジョブは，実行エージェントグループでの運用に対応していません。詳細については，「7.6 イベントジョブ使用時の注意事項」を参照してください。

キューレスジョブの場合は，実行ホスト名を指定します。指定する実行ホスト名を実行エージェントプロファイルに設定しておくと，キューレスジョブの場合も実行エージェント制限を使用することができます。

(2) 実行優先順位

JP1/AJS3 では，実行ファイルを実行したときに起動するプロセスの優先順位を設定できます。処理を優先させ，できるだけ早く終了させたい実行ファイルがある場合など，ジョブに実行優先順位を設定する必要があるかどうかを検討します。

ジョブの実行優先順位を指定しない場合，JP1/AJS3 から実行するジョブの実行優先順位は低く設定されます。これは，JP1/AJS3 から実行したジョブがループするなど，一部のジョブの沈み込みによって JP1/AJS3 全体の沈み込みを防止するためです。ただし，ジョブの実行優先順位が低いと CPU 割り当ての優先順位が低くなります。このため，実行優先順位の高いプロセスに処理が集中すると，JP1/AJS3 から実行したジョブが CPU の割り当てを長時間待つことになり，ジョブ終了までに時間が掛かったり，ジョブプロセスが停止した状態が続いたりする事象が発生します。そのジョブプロセスが資源にロックを掛けた状態で CPU の割り当てを待つようなプロセスであった場合，同じ資源の解放待ちをしている他プロセスの実行にも影響を与えることになります。

このような事象は，1 プロセッサのシステム構成や実行優先順位の高い複数のプロセスが集中的に処理している場合など，CPU 使用率が比較的高くなりやすいシステム構成で顕著に発生しやすくなります。システム環境や運用に応じてジョブの実行優先順位が高くなるように変更する必要がないかどうかを検討してください。

(3) 終了判定

処理の終了状態を判定するための方法を検討します。標準ジョブおよび HTTP 接続ジョブでは，実行結果の終了判定ができます。終了判定には，次の五つの方法があります。

- 常に正常終了とする
- 常に異常終了とする
- 指定したファイルが実行先のエージェントホスト上にあれば正常終了とする
- 指定したファイルがジョブの実行開始から終了までの間に更新されれば正常終了とする
  フレキシブルジョブの場合，指定したファイルが宛先エージェント上でのジョブプロセスの実行開始から終了までの間に更新されれば正常終了とする
- しきい値（警告しきい値，異常しきい値）を設定し，ジョブ終了時の戻り値によって判定する

しきい値による判定では，ジョブ終了時の戻り値と設定したしきい値の大小関係で正常終了，警告終了，異常終了を判定します。終了時の戻り値が警告しきい値以下であれば正常終了，警告しきい値より大きく異常しきい値以下であれば警告終了，異常しきい値より大きければ異常終了とします。

なお，戻り値は符号なしの整数値として判定されます。例えば，「-1」は Windows では「4,294,967,295」，UNIX では「255」として扱われます。

(4)　異常終了時リトライ

ジョブに定義した実行ファイルが異常終了した場合に，ジョブを「異常検出終了」状態にしないで自動的にリトライさせるかどうかを検討します。実行ファイルの一時的なエラーであれば，自動的にリトライさせることでジョブのエラーを回復し，運用を継続できます。詳細については，「2.4.13 ジョブの異常終了時に自動でリトライする」を参照してください。

(5)　転送ファイル

標準ジョブの場合，実行ファイルが実行先のエージェントホスト上にないときに，マネージャーホストからエージェントホストにファイルを転送して実行させることもできます。また，jpqjobsub コマンドを使用してサブミットジョブを実行する場合は，jpqjobsub コマンドを実行したホストから，実行先のエージェントホストにファイルを転送して実行することができます。なお，標準ジョブ，サブミットジョブのどちらも，転送できるファイルは，テキスト形式のファイルだけです。

なお，フレキシブルジョブはファイルを転送できません。

(6)　実行先サービス

キューレスジョブを使用する場合は，ジョブの実行先サービスとして［キューレス］を指定します。デフォルトでは［標準］が設定されています。

キューレスジョブを使用すると，［標準］を設定しているジョブと比べて処理性能が高く，一定の時間により多くのジョブを実行できます。キューレスジョブについては，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　10.5 キューレスジョブ」を参照してください。

ただし，キューレスジョブでは，実行エージェントや実行エージェントグループなどが使用できないため，通常は実行先サービスを［標準］のまま使用してください。

(7)　打ち切り時間

ジョブを打ち切る時間を設定し，ジョブの実行が開始されてからの経過時間によってジョブの実行を打ち切ることができます。何らかの要因で処理が終了しない場合を想定し，打ち切り時間について検討してください。ジョブの実行を打ち切ることで，原因を調査したり，システム管理者に異常を通知する処理を実行させたり，異常終了した場合だけ特定の処理を実行させたりできます。

(8)　終了遅延監視

ジョブの実行所要時間を設定して，ジョブの実行が開始されてからの経過時間によって終了遅延を監視できます。詳細については，「5. 監視方法の検討」を参照してください。

(9) JP1 資源グループ

ジョブに対するユーザーのアクセス権限を検討します。詳細については，「6. アクセス権の検討」を参照してください。

(10) 注意事項

- ジョブに定義するコメントの末尾には半角空白文字を使用しないでください。コメントの末尾に連続して設定した任意の数の半角空白文字はすべて無効になります。
- ジョブの定義内容に機種依存文字を使用しないでください。機種依存文字を使用した場合，文字化けするおそれがあります。

2.2　ジョブネットの構築のための検討

複数のジョブに実行順序を付けて一つの業務としてまとめ，ジョブネットに定義します。ジョブネットを構築するために，ジョブの実行順序や JP1/AJS3 でジョブを管理するための階層化などについて検討します。

2.2.1　ジョブの実行順序を検討する

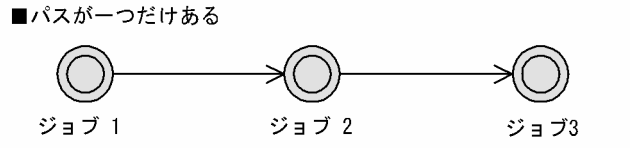
ジョブの構築について検討できたら，業務ごとにジョブの実行順序（ジョブフロー）について検討します。処理の流れや階層をチャートにしておくと，ジョブやジョブネットを定義する際に便利です。

ジョブフローの作成例を次に示します。

(1)　処理の経路が一つだけの例

ジョブ 1，ジョブ 2，ジョブ 3 という三つのジョブに順序性があり，処理の経路が一とおりだけある場合のジョブフローの作成例を次の図に示します。

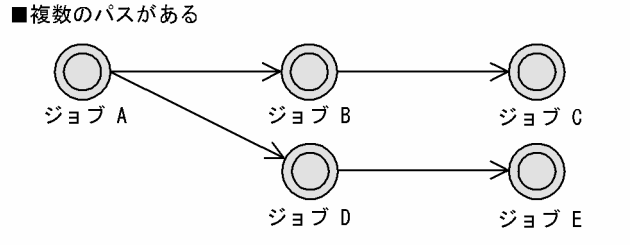
図 2–1　パスが一つだけのジョブフロー



(2)　処理の経路が複数ある例

処理の経路が複数ある場合のジョブフローの作成例を，次の図に示します。

図 2–2　パスが複数ある場合のジョブフロー



この場合，ジョブ A が実行されると，「ジョブ A－ジョブ B－ジョブ C」というパスと「ジョブ A－ジョブ D－ジョブ E」というパスの二つに処理が分岐されることになります。

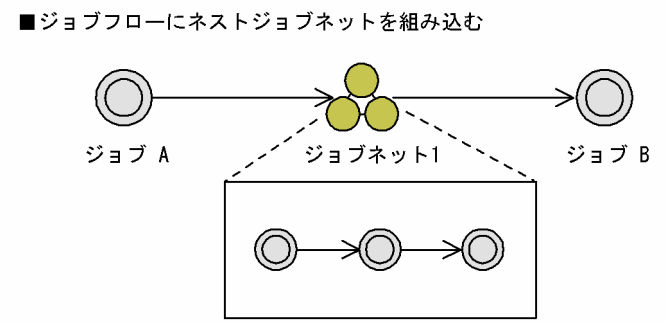
(3)　ネストジョブネットを使った例

ジョブフローにジョブネットを組み込むこともできます。ネストジョブネットの使用例を次に示します。

(a)　ジョブネットを入れ子にする

ジョブフローに別のジョブネットを組み込む場合の例を，次の図に示します。

図 2-3　ネストジョブネットの使用例



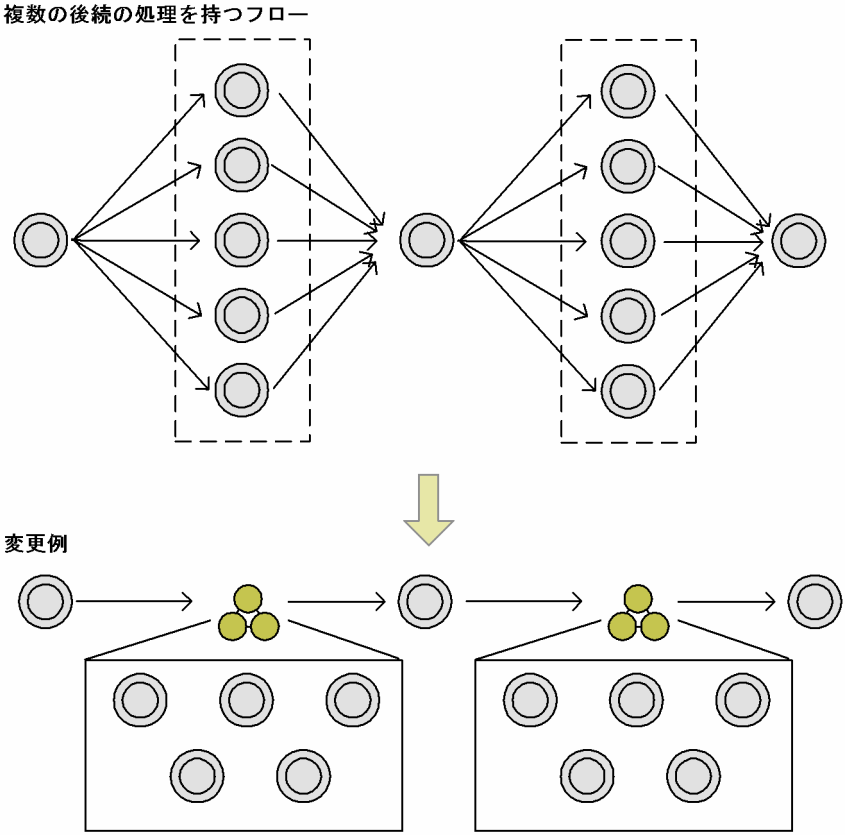
この場合，ジョブ A の実行が終了したら，ジョブネット 1 に定義されているジョブが実行され，ジョブネット 1 の処理が終了したらジョブ B が実行されます。

(b)　複数のジョブをまとめる

複数の後続ジョブを持つフローは推奨しません。複数のジョブを一つのジョブネットとしてまとめることで，管理しやすくできます。

複数のジョブをネストジョブネットとしてまとめた場合の例を，次の図に示します。

図 2-4　複数のジョブをネストジョブネットとしてまとめた場合の例

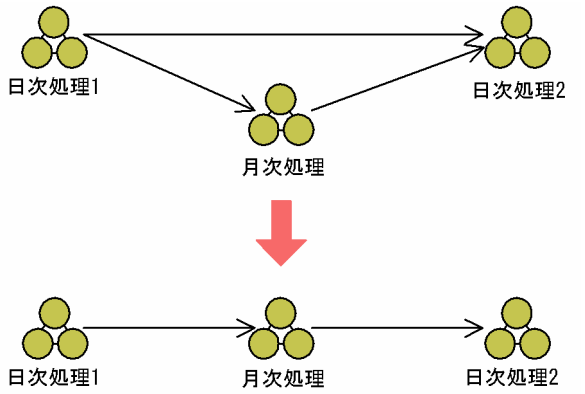


(c)　二つの経路を一つにまとめる

ネストジョブネットを使って，二つの経路を一つにまとめることができます。

「日次処理 1－日次処理 2」と，「日次処理 1－月次処理－日次処理 2」という二つのパスを一つのフローにまとめる例を，次の図に示します。

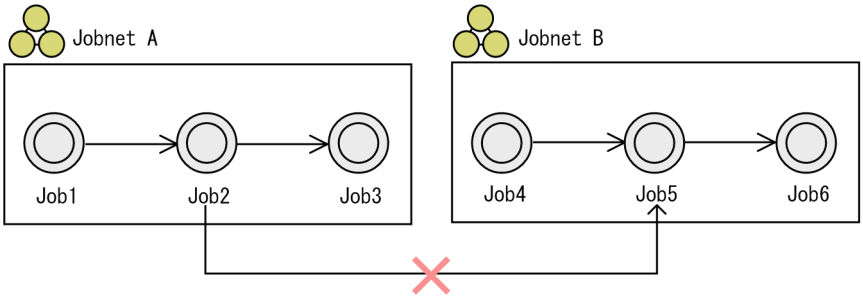
図 2-5　ジョブネットのフロー化



「日次処理 1」および「日次処理 2」は毎日実行され，「月次処理」は月に一度だけ実行される場合，JP1/AJS3 ではその日に実行予定のないジョブネットをスキップして実行するため，一つのパスにまとめることができます。

(4)　異なるジョブネット間のジョブ同士を順序づけたい場合

JP1/AJS3 では，異なるジョブネット間のジョブ同士を関連線で順序づけることはできません。



異なるジョブネット間のジョブ同士を関連線で順序づけることはできない。

異なるジョブネット間のジョブ同士は，次の方法で順序づけることができます。

- ジョブネットを分割する
- ジョブネットを統合する
- ジョブネットコネクタを使用する
- 待ち合わせ条件を使用する

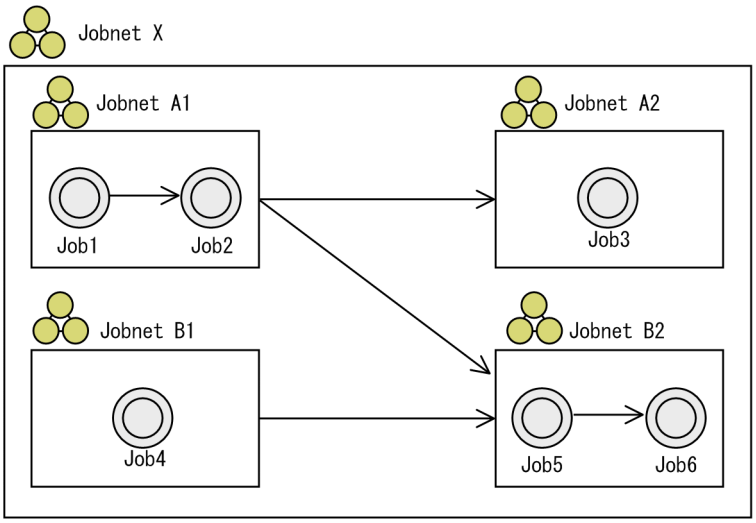
それぞれの場合について，次に説明します。

(a)　ジョブネットを分割する

ジョブネットを分割する場合の例を，次の図に示します。

図 2-6　ジョブネットを分割する例

■ジョブネットを分割する

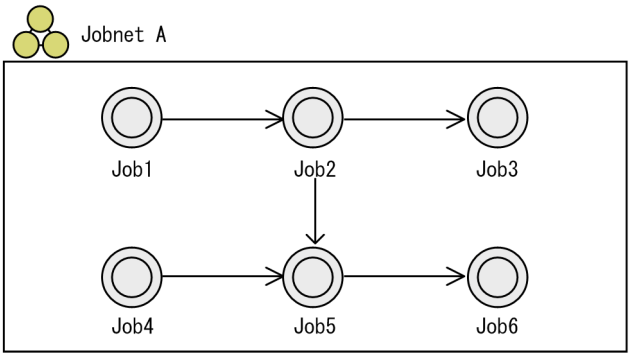


(b)　ジョブネットを統合する

ジョブネットを統合する場合の例を，次の図に示します。

図 2-7　ジョブネットを統合する例

■ジョブネットを統合する

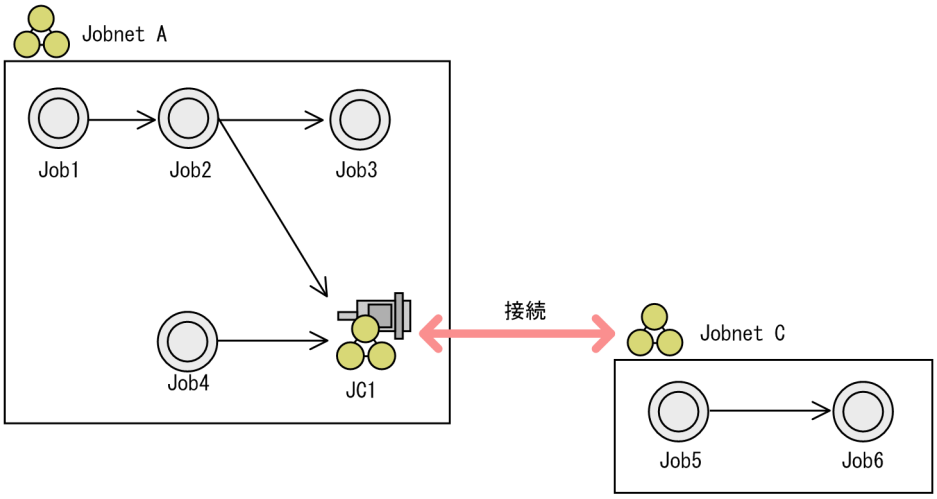


(c)　ジョブネットコネクタを使用する

ジョブネットコネクタを使用すると，異なるルートジョブネット間の実行順序を制御できます。ジョブネットコネクタを使用する場合の例を，次の図に示します。

図 2-8　ジョブネットコネクタを使用する例

■ジョブネットコネクタを使用する



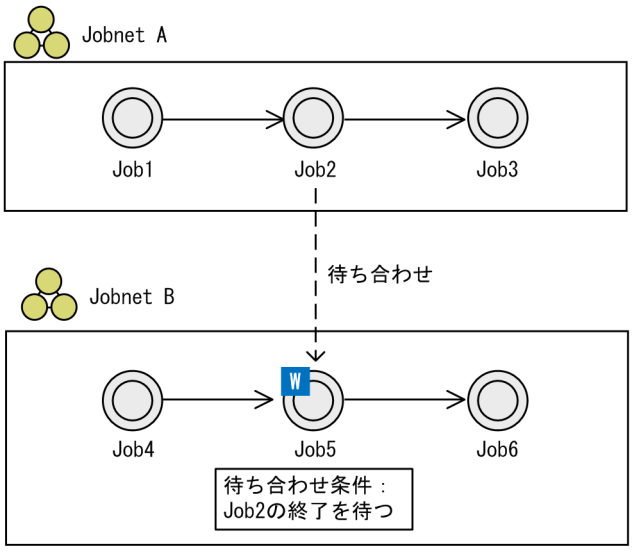
ジョブネットコネクタの詳細については，「2.2.4 ジョブネットコネクタを使用したルートジョブネットの実行順序制御」を参照してください。

(d)　待ち合わせ条件を使用する

待ち合わせ条件を使用すると，異なるジョブネットの配下にあるユニット同士の実行順序を制御できます。待ち合わせ条件を使用する場合の例を，次の図に示します。

図 2-9　待ち合わせ条件を使用する例

■待ち合わせ条件を使用する



待ち合わせ条件の詳細については，「2.2.5 待ち合わせ条件を使用したユニットの実行順序制御」を参照してください。

(5)　ジョブネットの階層化の考え方

ジョブネットを階層化する目的またはメリットとして，次のことが考えられます。

- ジョブやジョブネットが監視しやすくなります。
- ジョブやジョブネットを変更する際，場所が特定しやすくなります。
- 業務単位にアクセス権（JP1 資源グループ）を割り当てることで，ほかの業務グループの担当者からのアクセスを制限できます。
- 適度に階層化することで，ジョブやジョブネットの起動性能が劣化しにくくなります。

補足事項

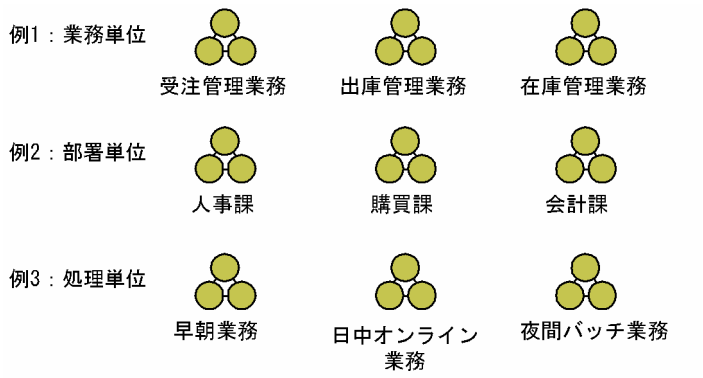
- ジョブネット数と階層化に関する注意事項については，「7.1 ルートジョブネットの実行登録数に関する注意事項」を参照してください。
- 階層を深くし過ぎると，監視などの操作をする上で運用しにくくなります。そのため，階層化は 2～3 階層程度にすることを推奨します。

推奨するジョブネットの階層化の方法を次に示します。

1. **最上位のジョブネットを作成する。**

   次の図のように，特定の単位ごとに最上位となるジョブネットを作成します。

   図 2-10　最上位のジョブネットの作成例

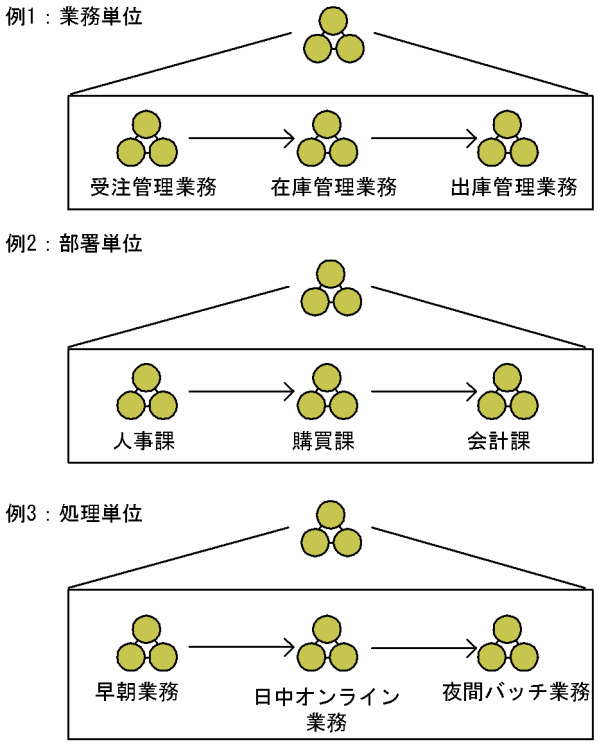


   相互に順序性がない場合は，そのまま最上位のジョブネットとして管理します。

2. **特定の単位ごとに作成したジョブネットに順序性がある場合は，一つのジョブネットにまとめ，階層化する。**

   それぞれのジョブネットが順序性を持つ場合の階層化の例を次に示します。

図 2–11　それぞれのジョブネットが順序性を持つ場合の階層化の例

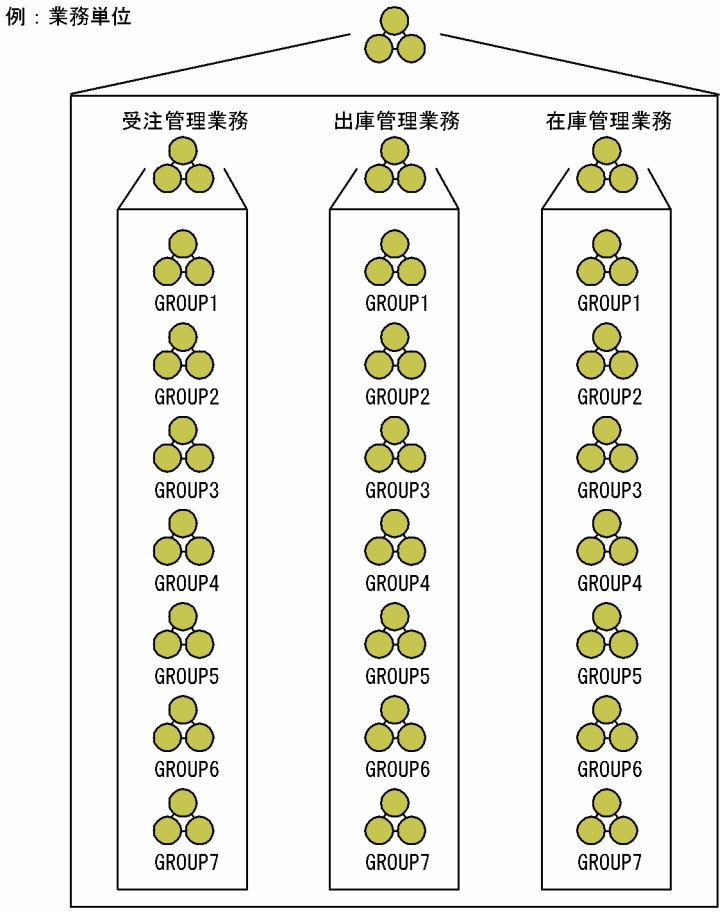


3. 処理サイクルごとにジョブネットを作成する。

業務単位に異なるサイクルで実行する必要がある場合は，処理サイクルごとにジョブネットをグルーピングします。

処理サイクルごとにジョブネットを作成する場合の例を次に示します。

図 2-12　処理サイクルごとのジョブネットの作成例



処理サイクルに作成するジョブネットには，次の表に示すようなルールを決めておきます。また，このルールを「コメント」として定義しておくことを推奨します。

表 2-5　処理サイクルごとにジョブネットを分けた場合のルール

処理サイクルごとのジョブネット	ルール
GROUP1	日次，週次，月次，半期，年次の業務が相互に関連を持ったグループ
GROUP2	単独で実行される「日次業務」のグループ
GROUP3	単独で実行される「週次業務」のグループ
GROUP4	単独で実行される「月次業務」のグループ
GROUP5	単独で実行される「半期業務」のグループ
GROUP6	単独で実行される「年次業務」のグループ
GROUP7	単独で実行される「不定期業務」のグループ

4. 最下位のジョブネットを作成する。

処理サイクルごとに分けたジョブネットの下位に，さらに次のような名称を付けたジョブネットを作成します。ジョブやジョブネットをコマンドで操作する場合や，JP1/IM の自動アクション機能で正規表現を使用する場合を考慮に入れて，半角英数字で名称を付けることを推奨します。

「xxxxxxDN」
「xxxxxxWN」
「xxxxxxMN」
「xxxxxxHN」
「xxxxxxYN」
「xxxxxxRN」

（凡例）

D：1 日 1 回のサイクルで実行するジョブネット
W：週に 1 回のサイクルで実行するジョブネット
M：月に 1 回のサイクルで実行するジョブネット
H：半期に 1 回のサイクルで実行するジョブネット
Y：年に 1 回のサイクルで実行するジョブネット
R：不定期に実行するジョブネット
N：ジョブネットを意味する

5. ジョブを作成する。

最後に，次のような名称を付けたジョブを作成します。

「xxxxxxDJ」
「xxxxxxWJ」
「xxxxxxMJ」
「xxxxxxHJ」
「xxxxxxYJ」

（凡例）

D：1 日 1 回のサイクルで実行するジョブ
W：週に 1 回のサイクルで実行するジョブ
M：月に 1 回のサイクルで実行するジョブ
H：半期に 1 回のサイクルで実行するジョブ
Y：年に 1 回のサイクルで実行するジョブ
J：ジョブを意味する

2.2.2　ジョブの実行に失敗したときの対処方法を検討する

ジョブの実行順序について検討したら，ジョブの実行に失敗したときの対処方法について検討します。

検討する内容は次の二つです。

- 失敗したジョブのあとに自動的に実行させる処理の方法

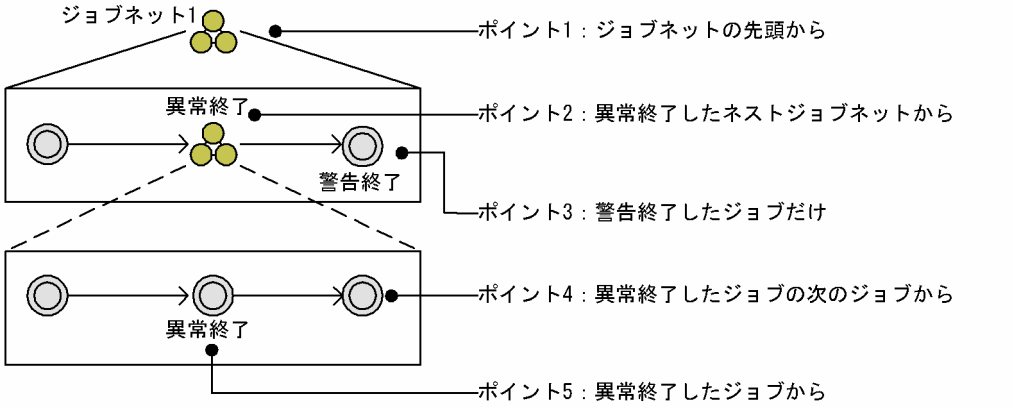
- 同じジョブを再実行する場合の方法

JP1/AJS3 では，実行ファイルのリターンコードで処理結果を判定します。そのため，例えば異常終了を意味するリターンコードが出力されたときに，システム管理者に異常を知らせるメールを送信するなどの処理を自動的に実行させることができます。

なお，同じジョブを再度実行させる場合は，一部のジョブだけ実行させるのか，最初からすべてのジョブを実行させるのかについて検討してください。

同じジョブを実行させる場合の再実行ポイントを次の図に示します。

図 2-13　業務の再実行ポイント



ジョブネットを再実行する場合は，最初からすべてのジョブを実行させるパターンでも，一部のジョブだけ実行させるパターンでも，一時的に保留状態にしてから再実行できます。保留にしておくと，保留解除するまで実行されないため，実行予定を一時的に変更したり，中止したりできます。

一部のジョブだけを再実行させる場合は，異常終了したジョブを再実行させるのか，異常終了したジョブの次のジョブから再実行させるのかなどについても検討してください。

2.2.3　ジョブネットの定義内容を検討する

ジョブネットに定義する内容について検討します。

ジョブネットに定義する詳細な項目については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12. ウィンドウとダイアログボックス」にあるルートジョブネットの詳細定義ダイアログボックスの説明を参照してください。

(1)　実行エージェント

ジョブネットの実行先となる実行エージェント名を指定します。実行エージェント制限を使用している場合は，実行エージェントプロファイルの一覧から選択できます。実行エージェントは，ジョブやジョブネットの実行先となるエージェントホストの論理的な名称です。マネージャーに定義した実行エージェント情

報に従って，ジョブやジョブネットに指定した実行エージェントとエージェントホストの物理的なホスト名がマッピングされ，ジョブが配信されます。

ジョブ実行環境が実行エージェントグループによって負荷分散させる構成の場合は，実行エージェントグループを指定すると，グループ化された実行エージェントの優先順位に従ってジョブを配信します。

(2) 多重起動とスケジューリング方式

周期的に実行させるジョブネットについて，次の実行の予定開始時刻までに処理が終了しなかったときに，処理を多重に実行させるかどうかについて検討します。詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.3.3 多重起動とスケジューリング方式」を参照してください。

(3) 保存世代数

何回分のジョブネットの実行結果ログを保存するかについて検討します。詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.2.3 ジョブネットの保存世代数の管理」を参照してください。

(4) 打ち切り時間

ジョブネットの開始予定日時になっても実行が開始されない場合，どのくらい経過した時点で実行開始待ちの状態を打ち切るかについて検討します。実行開始待ちの状態を打ち切ったジョブネットは，「繰り越し未実行」状態になります。詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.1.1(2) ジョブネット」の打ち切り時間についての説明を参照してください。

保留状態のジョブネットが打ち切り時間を経過した場合も，「繰り越し未実行」状態になります。また，打ち切り時間の直前に一時変更操作をすると，一時変更中に打ち切り時間を超過して「繰り越し未実行」状態になることがあります。この場合，必要に応じてジョブネットの再実行を検討してください。「繰り越し未実行」状態にしたくない場合は，次のように余裕を持った打ち切り時間を設定することも検討してください。

打ち切り時間を「1 日」で検討していた場合

　「2 日」または「無制限」

打ち切り時間を「2 日」で検討していた場合

　「無制限」

打ち切り時間の定義を「システム設定に従う」にする場合は，環境設定パラメーターEXECDEFER に上記に相当する値を設定してください。

(5) ジョブネット監視

ジョブネットの実行所要時間を設定して，ジョブネットの実行が開始されてからの経過時間によって終了遅延を監視できます。詳細については，「5. 監視方法の検討」を参照してください。

(6) 実行順序制御

ジョブを管理する階層の最上位のジョブネット同士を順序づけできます。詳細については，「2.2.4 ジョブネットコネクタを使用したルートジョブネットの実行順序制御」を参照してください。

(7) JP1 資源グループ

ジョブネットに対するユーザーのアクセス権限を検討します。詳細については，「6. アクセス権の検討」を参照してください。

(8) 注意事項

- ジョブネットに定義するコメントの末尾には半角空白文字を使用しないでください。コメントの末尾に連続して設定した任意の数の半角空白文字はすべて無効になります。
- ジョブネットの定義内容に機種依存文字を使用しないでください。機種依存文字を使用した場合，文字化けするおそれがあります。

2.2.4 ジョブネットコネクタを使用したルートジョブネットの実行順序制御

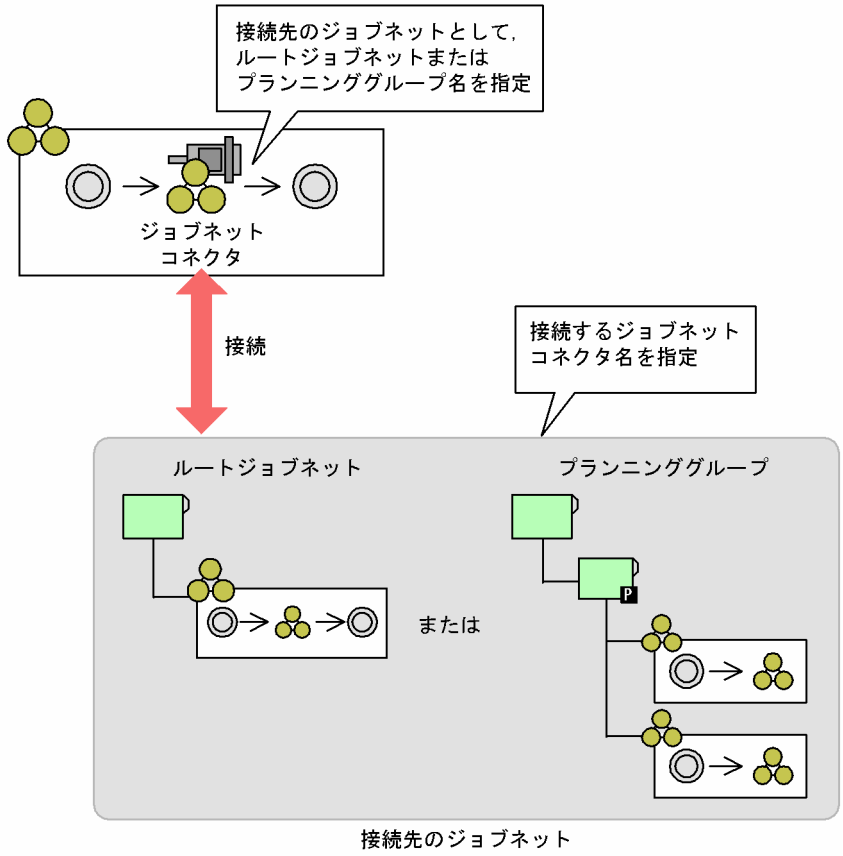
ルートジョブネットは，ジョブやネストジョブネットのようにユニットを並べて実行づけすることはできませんが，**ジョブネットコネクタ**というユニットを使用することで，ルートジョブネット同士の実行順序を制御できます。

(1) ジョブネットコネクタを使ったルートジョブネット実行順序制御の概要

ジョブネットコネクタは，実行順序を制御するルートジョブネットと接続関係を持つことで，ルートジョブネットの終了を待ち合わせたり，ルートジョブネットの開始を同期させたりして，実行のタイミングや順序を制御できます。

ジョブネットコネクタと実行順序を制御するルートジョブネットの関係を，次の図に示します。

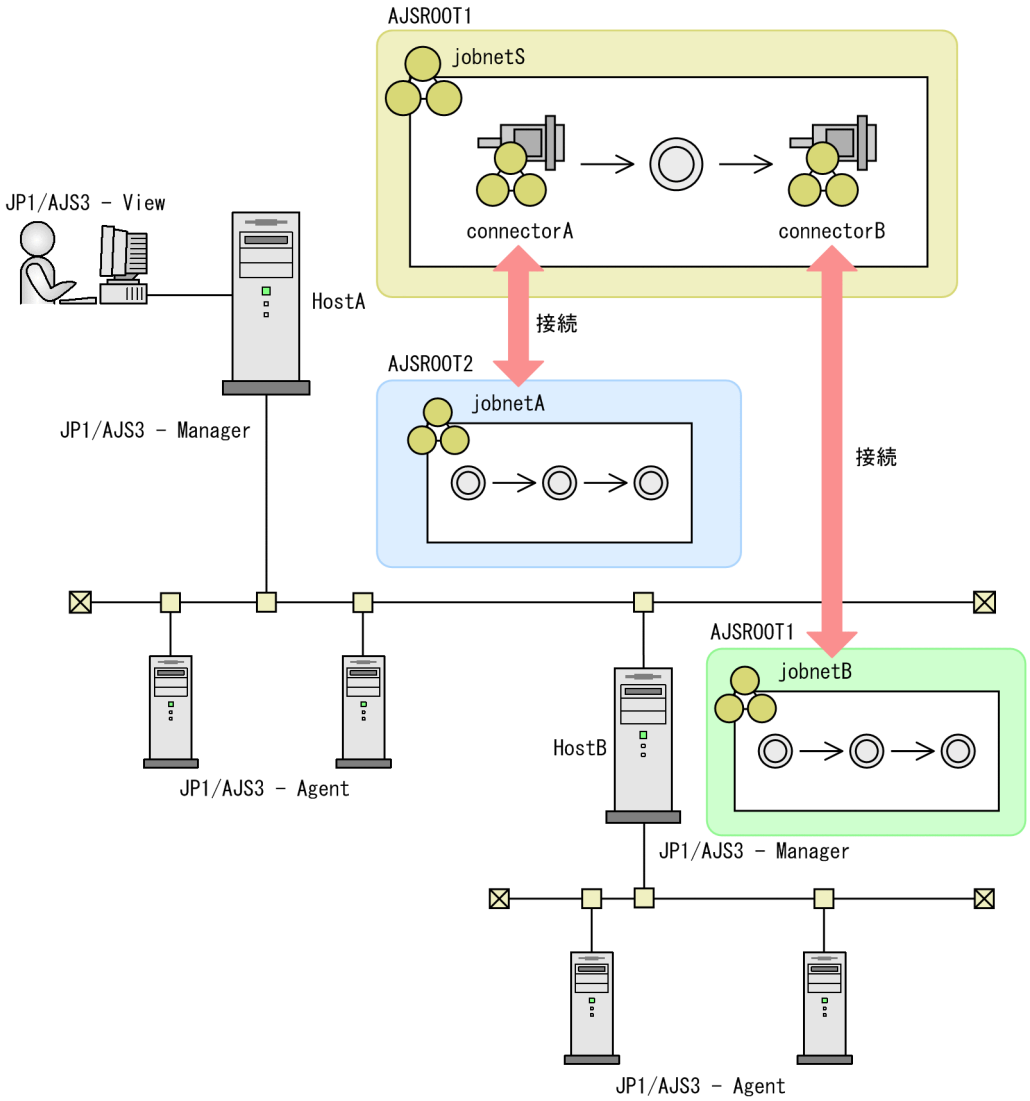
図 2-14　ジョブネットコネクタと実行順序を制御するルートジョブネットの関係



ジョブネットコネクタは，ジョブネット配下にユニットの一つとして定義できます。

ジョブネットコネクタを使って実行順序を制御できる対象は，ルートジョブネットまたはプランニンググループ直下のルートジョブネットです。異なるスケジューラーサービス間でもルートジョブネットの実行順序を制御できます。異なるスケジューラーサービス間のルートジョブネットの実行順序制御について，次の図に示します。

図 2-15　異なるスケジューラーサービス間のルートジョブネットの実行順序制御



なお，ジョブネットコネクタと接続関係を持つルートジョブネットを**接続先のジョブネット**といいます。

ジョブネットコネクタを使ったルートジョブネットの実行順序制御は，次の流れで行います。

1. **ジョブネットを作成し，ジョブネットコネクタを定義する。**

2. **ジョブネットコネクタと実行順序制御の対象との接続関係を定義する。**

   ジョブネットコネクタ側に，接続先のジョブネット名を指定します。実行順序制御の対象がルートジョブネットの場合はルートジョブネット名を，プランニンググループ直下のルートジョブネットの場合はプランニンググループ名を指定します。

   接続先のジョブネット側には，対応するジョブネットコネクタ名を指定します。

   補足事項

   JP1/AJS3 - View のメニューコマンド［ジョブネットコネクタとして記憶］［ジョブネットコネクタの自動生成］を使用すると，定義を手入力する手間が省けます。ただし，異なるスケジューラーサービス間で実行順序制御をする場合は使用できません。この場合は手入力で定義してください。

3. それぞれのジョブネットを実行登録する。

ジョブネットおよびジョブネットコネクタの定義方法については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　5. ジョブネットの定義」を参照してください。ジョブネットの実行登録方法については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　7. ジョブネットの実行」を参照してください。

注意事項

- JP1/AJS - View および接続先の JP1/AJS - Manager が 08-10 より前のバージョンでは，ジョブネットコネクタを使用できません。
- JP1/AJS - View および接続先の JP1/AJS - Manager のバージョンが 08-10 のジョブネットコネクタでは，異なるスケジューラーサービス間のルートジョブネットの実行順序を制御できません。異なるスケジューラーサービス間でジョブネットコネクタを使用できるのは，08-50 以降の JP1/AJS - View および JP1/AJS - Manager です。
- ジョブネットコネクタは，ルートジョブネットおよびネストジョブネット以外のユニット配下に定義できません。
- ジョブネットコネクタの接続先のジョブネットとして，ルートジョブネットおよびプランニンググループ以外のユニット種別を指定できません。
- 接続先のジョブネット配下には，ジョブネットコネクタを定義しないでください。接続先のジョブネット配下にジョブネットコネクタを定義した場合，ジョブネットの実行登録時にエラーになります。
- ジョブネットコネクタを定義するジョブネットおよび接続先のジョブネットには，起動条件を設定しないでください。ジョブネットコネクタを定義したジョブネットおよび接続先のジョブネットに起動条件を設定した場合，ジョブネットの実行登録時にエラーになります。
- ジョブネットコネクタは，従属ユニットおよびリカバリーユニットとしては定義できません。
- ジョブネットコネクタは，従属ジョブネットおよびリカバリージョブネット配下に定義できますが，接続関係が複雑になるため，これらの配下には定義しないことを推奨します。
- サスペンド中は，ジョブネットコネクタを新規作成および削除できません。
- スケジューラーサービス間および同一ホスト内での実行順序制御機能（ジョブネットコネクタ）は，NAT を経由して使用できません。

(2)　ジョブネットコネクタと接続先のジョブネットの接続ルール

ジョブネットコネクタが定義されたジョブネットと接続先のジョブネットは，それぞれの実行登録後，同じ実行日の世代同士が接続関係を結びます。接続関係を結んだ世代間でルートジョブネットの終了を待ち合わせたり，ルートジョブネットの開始を同期させたりできます。

(a)　接続ルール

ジョブネットコネクタが定義されたジョブネットと接続先のジョブネットの世代について，次の条件が重なる場合に接続関係が成立します。

- 実行日が同じ
- 別の世代との接続関係が確定していない世代※

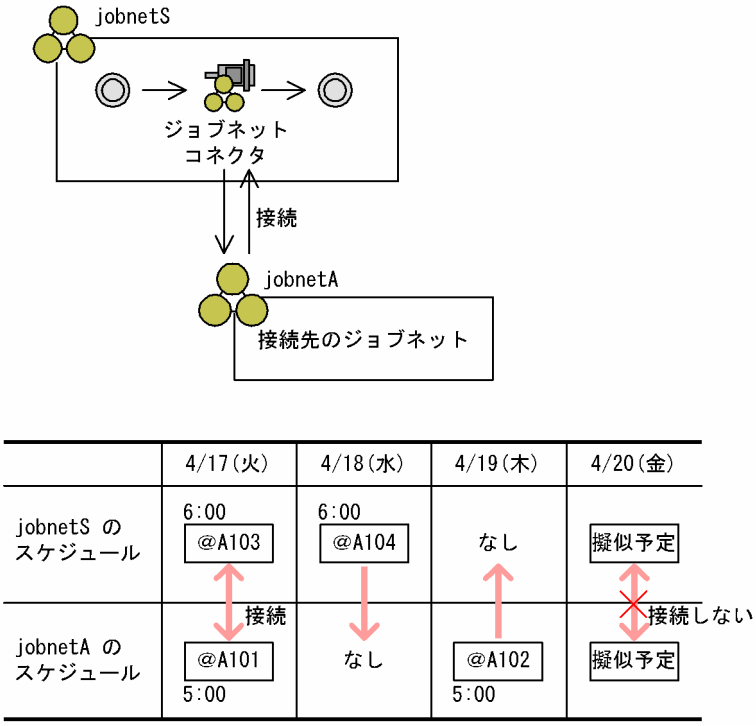
注※

ジョブネットコネクタが定義されたジョブネットと接続先のジョブネットとの接続関係は，どちらか一方がまだ実行を開始していないときには確定していません。両方が実行を開始したとき，または繰り越し未実行など実行しないで終了したときに確定します。

実行予定が 1 日に複数回ある場合は，これらの条件を満たす世代のうち開始予定日時が最も早い世代同士で接続関係が成立します。ただし，擬似予定の段階では接続関係は成立しません。

ジョブネットコネクタを定義したルートジョブネット（jobnetS）と接続先のジョブネット（jobnetA）を実行登録した場合の接続例を，次の図に示します。

図 2-16　ジョブネットコネクタと接続先のジョブネットの接続例



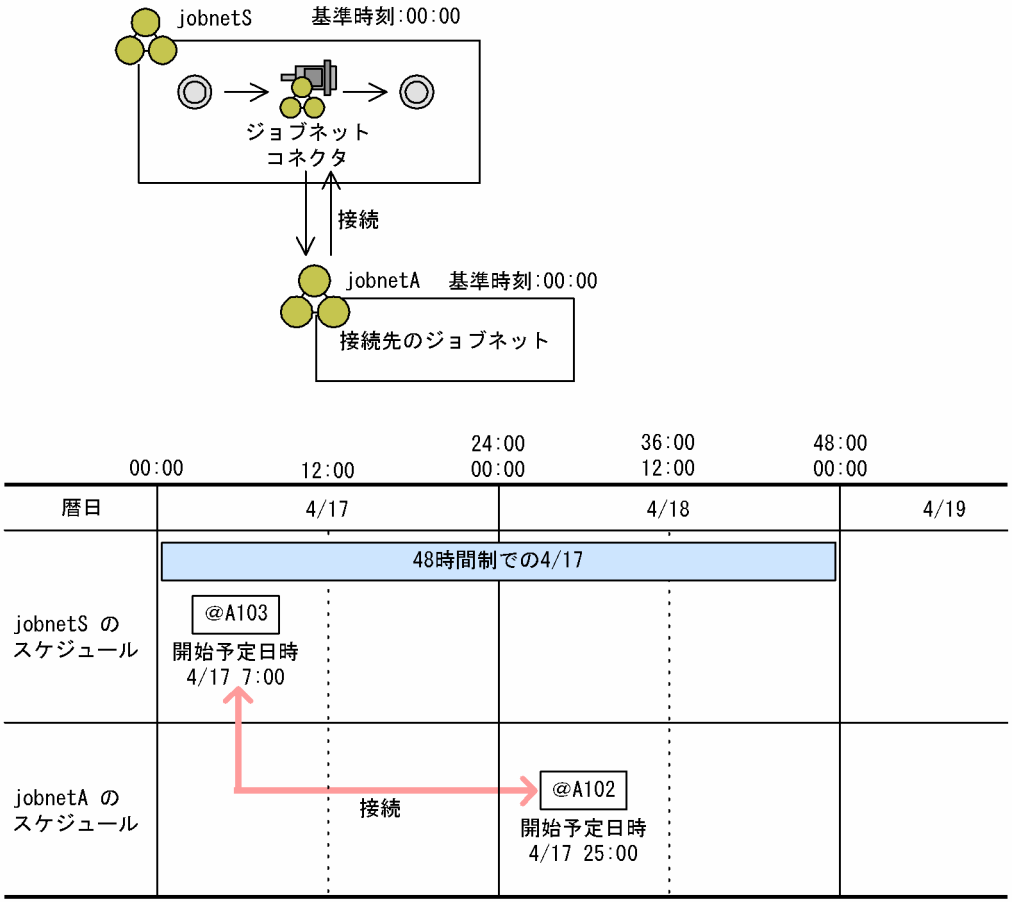
jobnetS の世代「@A103」と jobnetA の世代「@A101」は，同一実行日の世代であるため接続関係が成立します。jobnetS の世代「@A104」および jobnetA の世代「@A102」は，ともに同一実行日に対応する世代がないため接続されません。4/20（金）には jobnetS および jobnetA ともに擬似予定がありますが，擬似予定の段階では接続関係は成立しません。

(b)　48 時間制の場合

48 時間制を採用している場合でも，同一実行日の世代であれば接続関係が成立します。

48 時間制の場合の接続例を，次の図に示します。

図 2-17　ジョブネットコネクタと接続先のジョブネットの接続例（48 時間制の場合）



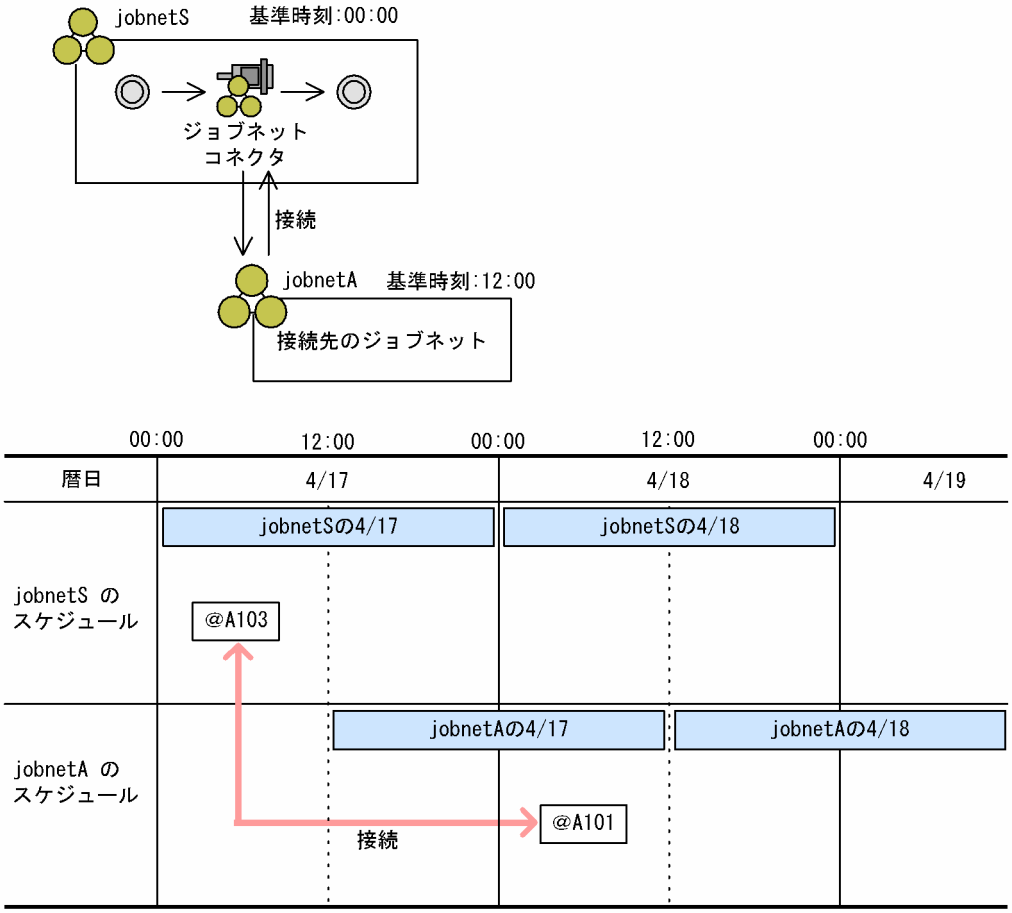
jobnetS の世代「@A103」と jobnetA の世代「@A102」は，暦日では実行日が異なりますが，48 時間制の場合は同一実行日となるため，接続関係が成立します。

(c)　基準時刻が異なる場合

ジョブネットコネクタを定義したルートジョブネットと接続先のジョブネットの基準時刻が異なる場合でも，同一実行日の世代であれば接続関係が成立します。

ジョブネットコネクタを定義したルートジョブネットと接続先のジョブネットの基準時刻が異なる場合の接続例を，次の図に示します。

図 2-18　ジョブネットコネクタと接続先のジョブネットの接続例（基準時刻が異なる場合）



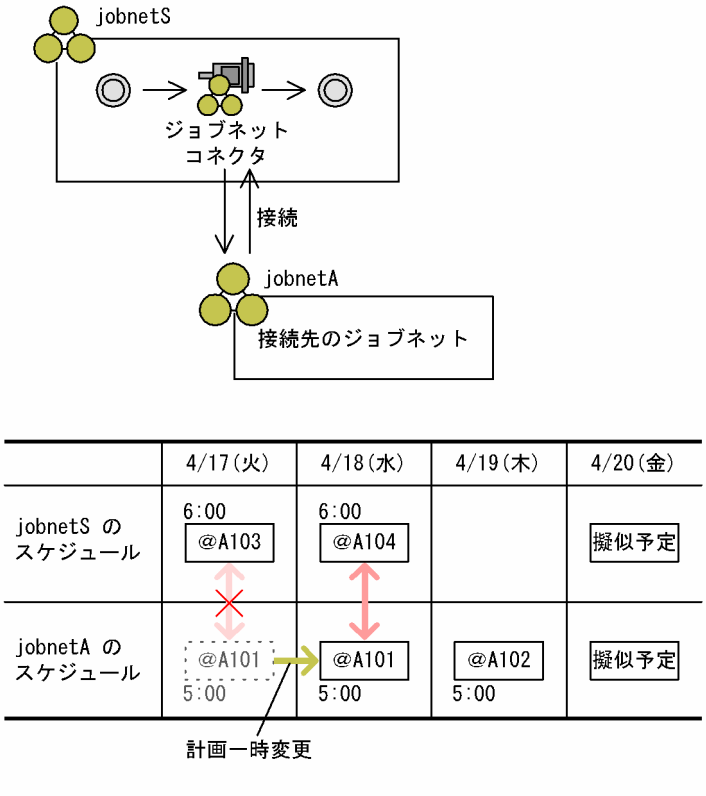
jobnetS の世代「@A103」と jobnetA の世代「@A101」は，暦日では実行日が異なりますが，それぞれの基準時刻で見るとどちらも同じ 4/17 であるため，接続関係が成立します。ただし，基準時刻が異なるルートジョブネットの実行順序を制御しようとした場合，この例のように暦日では実行日が異なる世代が接続されることがあるため，ジョブネットコネクタと接続先のジョブネットでは，基準時刻を合わせて運用してください。

(d)　世代の実行日を変更した場合

世代が接続されても，ジョブネットコネクタを定義したルートジョブネット，または接続先のジョブネットの世代が実行される前に，どちらかの世代を登録解除または実行中止した場合，接続関係が成立しなくなります。

また，計画一時変更やスケジュール定義変更によって世代の実行日が変わった場合は，接続関係も変わります。実行予定日を変更した場合の接続関係を，次の図に示します。

図 2-19　実行予定日を変更した場合の接続関係



このように，同一実行日で接続関係にある jobnetS の世代「@A103」と jobnetA の世代「@A101」を，計画一時変更によって「@A101」の実行予定を翌日の 4/18 にずらした場合，「@A103」との接続関係はなくなり，4/18 にある世代「@A104」と接続されます。

(e)　実行予定が 1 日に複数回ある場合

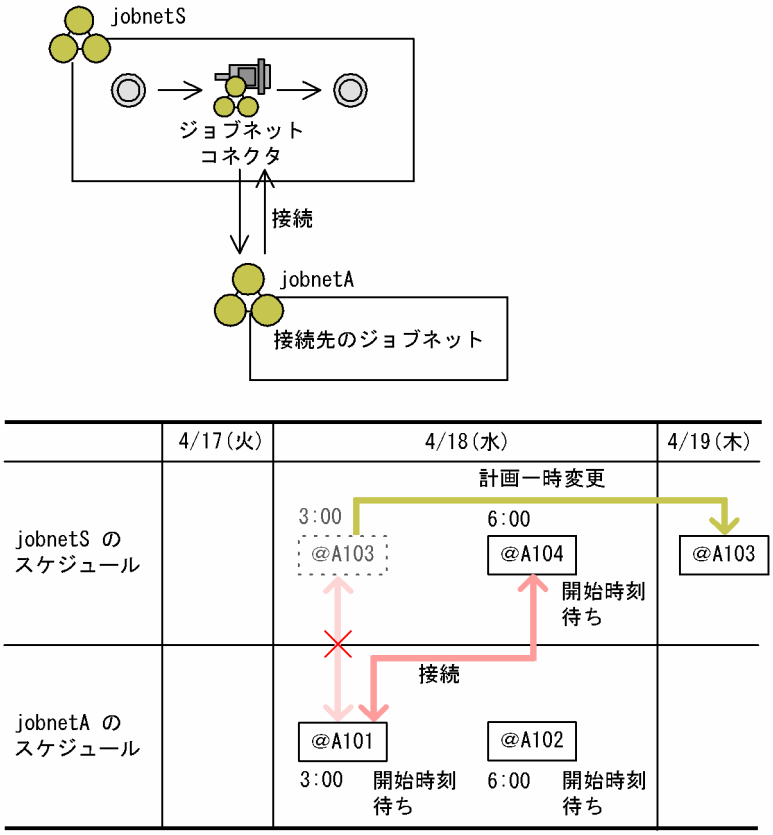
1 日に実行予定が複数回ある場合でも，同一実行日の世代であれば接続関係が成立します。計画一時変更や実行中止などによって実行予定が変更されたときなどに接続関係が複雑になるため，ジョブネットコネクタを定義したジョブネットおよび接続先のジョブネットの実行は 1 日 1 回とすることを推奨します。

注意事項

実行予定が 1 日に複数回あり，かつ実行順序制御方式を「同期」に設定している場合，ジョブネットコネクタを定義したジョブネットの実行が遅延することで，接続先ジョブネットの次回実行予定の世代が生成されないことがあります。この場合，遅延後の接続関係が複雑になってしまうため，ジョブネットコネクタを定義したジョブネットの実行は 1 日 1 回とすることを推奨します。詳細については，「(3) (b) 同期実行」を参照してください。

1 日に実行予定が複数回ある場合に実行予定を計画一時変更した例を，次の図に示します。

図 2-20　1 日に実行予定が複数回ある場合に実行予定を計画一時変更した例

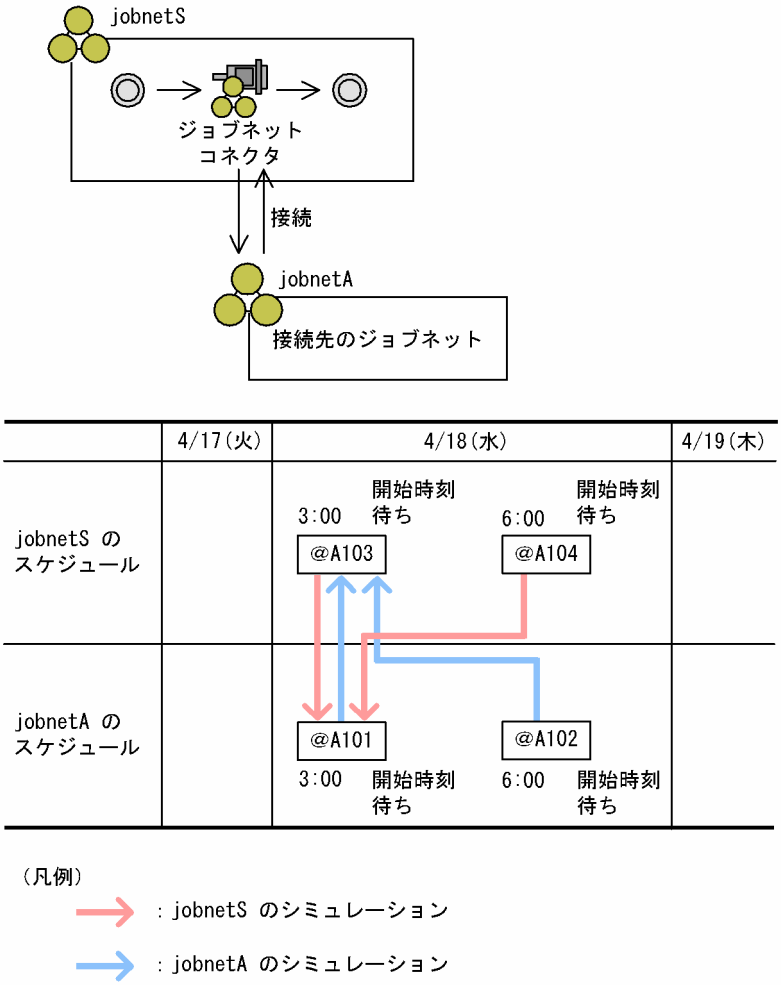


1 日に実行予定が複数回ある場合は，同一実行日の世代のうち，開始予定日時の早い順に接続関係が成立するため，jobnetS の世代「@A103」は jobnetA の世代「@A101」と接続関係を結びます。ここで，jobnetS の世代「@A103」の実行予定を 4/19（木）に計画一時変更した場合，jobnetS の世代「@A103」と jobnetA の世代「@A101」の接続関係はなくなり，jobnetA の世代「@A101」は jobnetS の世代「@A104」と接続関係を結びます。

また，1 日に実行予定が複数回ある場合に JP1/AJS3 - View や`ajsshow` コマンドで接続関係をシミュレーションすると，まだ実行されていない世代に複数の世代が接続するように見えます。

1 日に実行予定が複数回ある場合に接続関係をシミュレーションした例を，次の図に示します。なお，すべての世代が開始時刻待ちの状態であるとします。

図 2-21　1 日に実行予定が複数回ある場合に接続関係をシミュレーションした例



まず，jobnetS の世代「@A103」は，同一実行日で jobnetA の世代のうち開始予定日時が最も早い「@A101」と接続関係がシミュレーションされます。ただし，jobnetS の世代「@A103」と jobnetA の世代「@A101」との接続関係は確定していないため，jobnetS の世代「@A104」も jobnetA の世代「@A101」と接続関係がシミュレーションされます。同様に，jobnetA の世代「@A101」は jobnetS の世代「@A103」と，jobnetA の世代「@A102」も jobnetS の世代「@A103」と接続関係がシミュレーションされます。

(f)　タイムゾーンが異なる場合

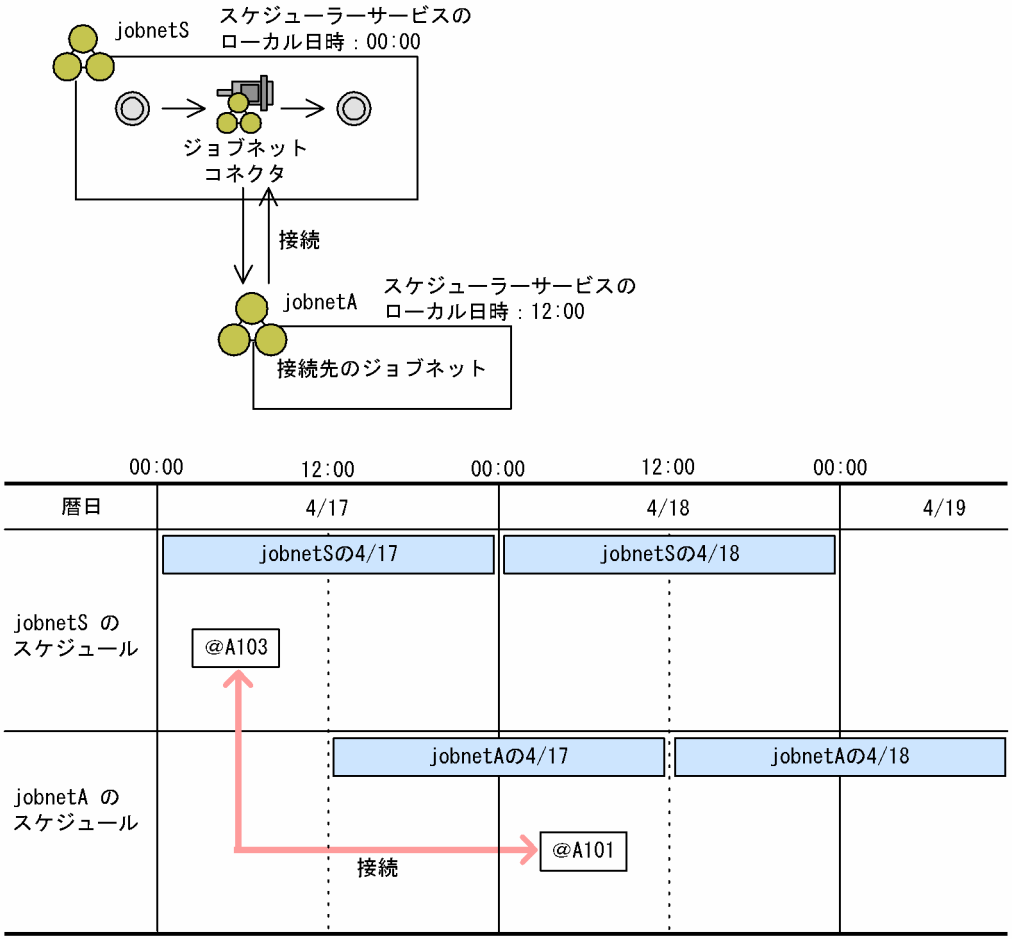
ジョブネットコネクタを定義したルートジョブネットと接続先のジョブネットのタイムゾーンが異なる場合でも，実行日が同一であれば世代間の接続関係が成立します。ただし，運用が困難になるため，ジョブネットコネクタと接続先のジョブネットのタイムゾーンを統一して運用してください。

(g)　スケジューラーサービスのローカル日時が異なる場合

ジョブネットコネクタを定義したルートジョブネットと接続先のジョブネットの，スケジューラーサービスのローカル日時が異なる場合でも，同一実行日の世代であれば接続関係が成立します。

ジョブネットコネクタを定義したルートジョブネットと接続先のジョブネットの，スケジューラーサービスのローカル日時が異なる場合の接続例を，次の図に示します。

図 2-22　ジョブネットコネクタと接続先のジョブネットの接続例（スケジューラーサービスのローカル日時が異なる場合）



jobnetS の世代「@A103」と jobnetA の世代「@A101」は，暦日では実行日が異なりますが，それぞれのスケジューラーサービスのローカル日時はどちらも同じ 4/17 であるため，接続関係が成立します。ただし，スケジューラーサービスのローカル日時が異なるルートジョブネットの実行順序を制御しようとした場合，この例のように暦日では実行日が異なる世代が接続されることがあるため，ジョブネットコネクタと接続先のジョブネットでは，スケジューラーサービスのローカル日時を合わせて運用してください。

(3)　ルートジョブネットの実行順序制御方式

ルートジョブネットの実行開始をジョブネットコネクタと同期して実行させるか，非同期実行させるかをルートジョブネットの実行順序制御方式といいます。実行順序制御方式は，接続先のジョブネット側に設定します。デフォルトは「非同期」です。

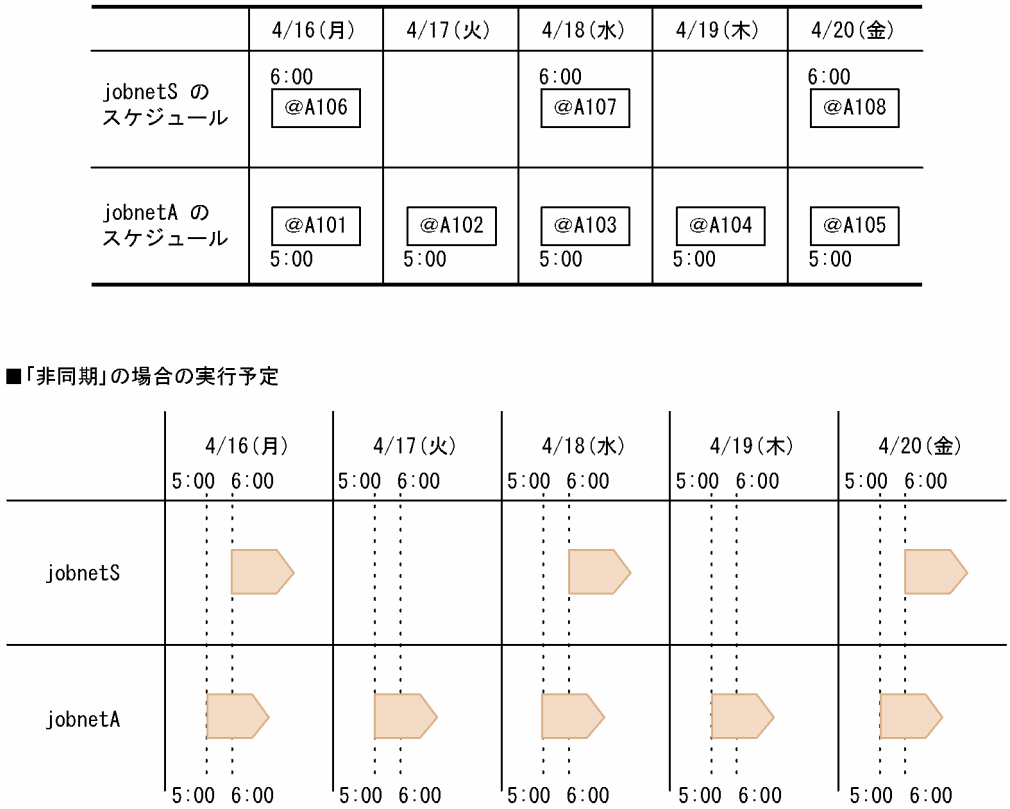
それぞれを設定した場合のルートジョブネット実行開始時の動作について説明します。

(a)　非同期実行

実行順序制御方式を「非同期」と設定した場合，接続先のジョブネットは接続関係の有無やジョブネットコネクタの状態にかかわらず，自身のスケジュールで実行を開始します。

ジョブネットコネクタが定義されているジョブネットを「jobnetS」，接続先のジョブネットを「jobnetA」とした，実行順序制御方式が「非同期」の場合の動作を，次の図に示します。

図 2-23　実行順序制御方式が「非同期」の場合の動作



注意事項

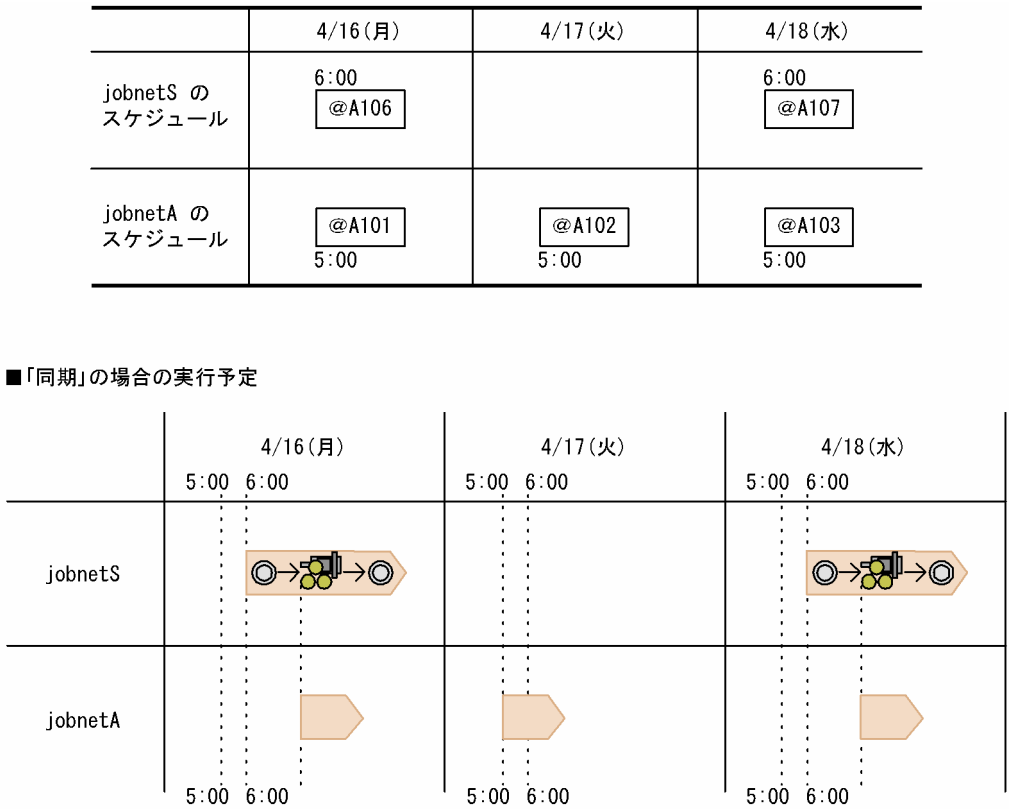
スケジューリング方式に多重スケジュールを設定していても，前回の実行予定が実行を開始していない場合は，次回実行予定が開始予定時刻になっても実行を開始しません。ジョブネットコネクタを定義したルートジョブネットおよび接続先のジョブネットで，それぞれ多重スケジュールを設定している場合は，次回実行予定の開始予定時刻になる前に，前回の実行予定が実行を開始するように運用してください。

(b)　同期実行

実行順序制御方式を「同期」と設定した場合，接続関係が成立している世代はジョブネットコネクタが実行開始するまで開始を待ち合わせます。同一実行日に対応する世代がない場合は，自身のスケジュールで実行を開始します。

ジョブネットコネクタが定義されているジョブネットを「jobnetS」，接続先のジョブネットを「jobnetA」とした，実行順序制御方式が「同期」の場合の動作を，次の図に示します。

図 2-24　実行順序制御方式が「同期」の場合の動作



この場合，jobnetS 自身の開始予定時刻は「6:00」，jobnetA の開始予定時刻は「5:00」ですが，4/16（月），4/18（水）の世代は接続関係が成立しているため，jobnetA は 5:00 には開始しないで，ジョブネットコネクタの実行と同期して開始します。接続関係を持たない 4/17（火）の世代は，自身のスケジュールに従って 5:00 に実行を開始します。

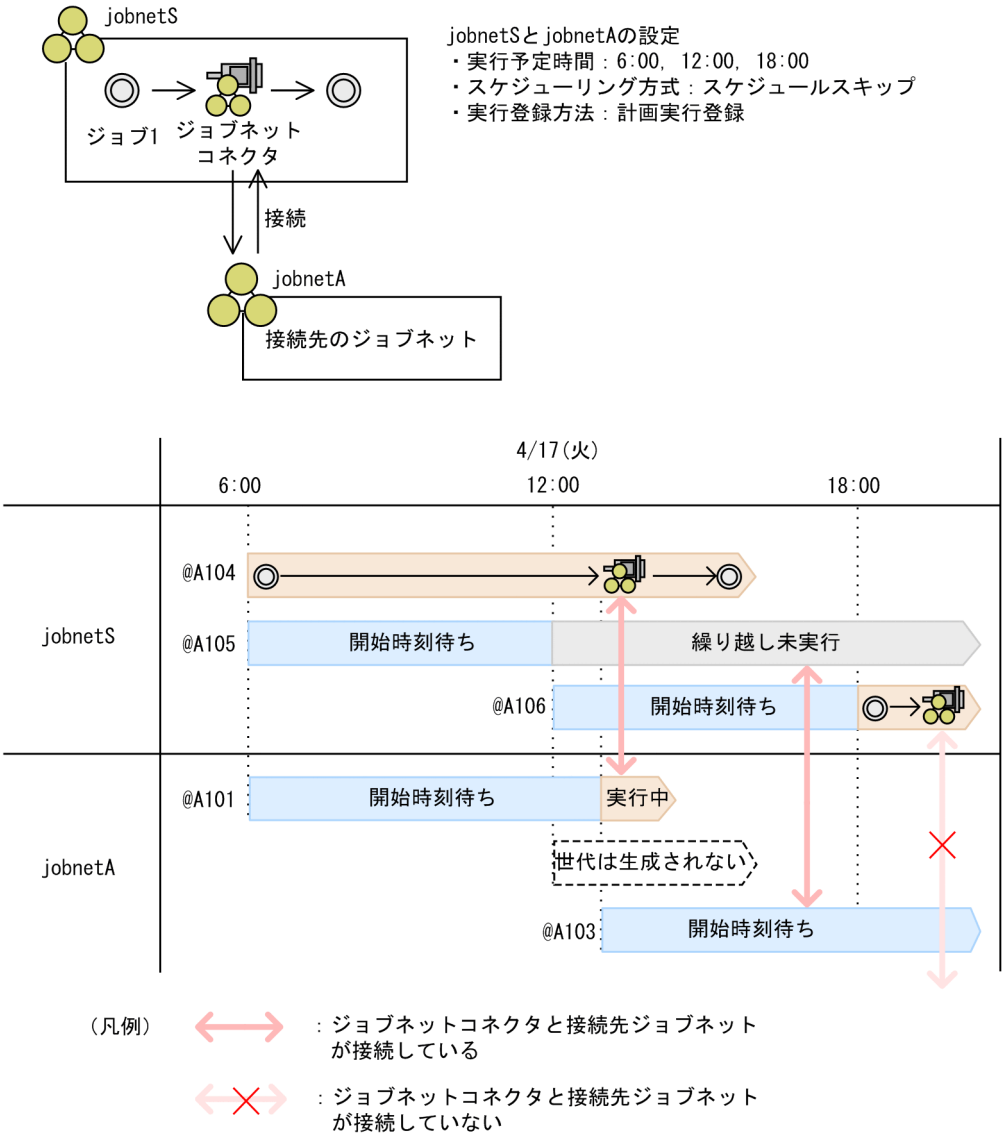
同期実行の設定は，一度実行したあとに再実行する場合には適用されません。再実行時にも同期させる場合は，実行順序制御方式の一時変更機能で一時変更してください。

注意事項

- スケジューリング方式に多重スケジュールを設定していても，前回の実行予定が実行を開始していない場合は，次回実行予定が開始予定時刻になっても実行を開始しません。ジョブネットコネクタを定義したルートジョブネットおよび接続先のジョブネットで，それぞれ多重スケジュールを設定している場合は，次回実行予定の開始予定時刻になる前に，前回の実行予定が実行を開始するように運用してください。
- 実行順序制御方式を「同期」に設定して 1 日に複数回実行予定がある場合，ジョブネットコネクタを定義したジョブネットの実行が遅延することで，接続先ジョブネットの次回実行予定の世代が生成されないことがあります。この場合，遅延後の接続関係が複雑になってしまうため，ジョブネットコネクタを定義したジョブネットの実行は 1 日 1 回とすることを推奨します。

  実行順序制御方式を「同期」に設定して 1 日に複数回実行する例を次の図に示します。

図 2-25　実行順序制御方式を「同期」に設定して 1 日に複数回実行する例



この例では，ジョブネットコネクタを定義したルートジョブネット（jobnetS）および接続先のジョブネット（jobnetA）の実行開始予定を，6:00，12:00，18:00 としています。また，スケジューリング方式にはスケジュールスキップを設定しています。実行登録方法は計画実行登録としています。

jobnetS の 6:00 に開始した世代「@A104」で，ジョブネットコネクタの先行ジョブ「ジョブ 1」が jobnetS の次の実行予定時間 12:00 を過ぎても終了しなかった場合，jobnetS および jobnetA は次のように動作します。

1. jobnetS の 12:00 実行予定の世代「@A105」が「繰り越し未実行」状態になる。

   jobnetS にはスケジュールスキップを設定しているため，実行予定時間が前の実行分の処理と重なっている世代は「繰り越し未実行」状態になり実行がスキップされます。

   また，jobnetA の 12:00 実行予定の世代は，世代「@A101」の実行が開始していないため，擬似予定のまま世代が生成されません。

2. ジョブ 1 の実行終了後，jobnetS の世代「@A104」のジョブネットコネクタが実行される。

同期実行を設定しているjobnetAの世代「@A101」も実行されます。同時に，jobnetAの次回実行予定である18:00実行予定の世代が「開始時刻待ち」状態で生成されます（実行ID「@A103」）。すると，「繰り越し未実行」状態のjobnetSの世代「@A105」とjobnetAの世代「@A103」との接続関係が成立します。jobnetAの世代「@A103」は「繰り越し未実行」状態のジョブネットと接続しているため，「開始時刻待ち」状態のまま実行されません。ジョブネットコネクタおよび接続先ジョブネットの状態遷移については，「(4) ジョブネットコネクタおよび接続先のジョブネットの状態遷移」を参照してください。

3. 時刻が18:00になると，jobnetSの18:00実行予定の世代「@A106」が実行される。

   jobnetSの世代「@A106」には接続する世代が存在しません。そのため，jobnetAに接続先の世代が生成されるかどうかを待つために，jobnetSの世代「@A106」はjobnetAの世代「@A103」が終了するまで「実行中」状態のまま待ち続けます。

   なお，実行順序制御方式を一時変更機能で「非同期」にすると，jobnetSの世代「@A106」は接続先の世代の有無に関係なく実行されます。実行順序制御方式の一時変更機能については，「(c) 実行順序制御方式の一時変更機能」を参照してください。

なお，時刻が18:00になる前であれば，「繰り越し未実行」状態になったjobnetSの世代「@A105」を再実行することで，接続関係が成立しているjobnetAの世代「@A103」も実行されます。jobnetAの世代「@A103」の終了後，新しく世代が生成される場合は，jobnetSの世代「@A106」と接続関係が成立してjobnetSの世代「@A106」が実行されます。新しく世代が生成されない場合は，jobnetSの世代「@A106」のジョブネットコネクタはジョブネットと接続できないため，「計画未実行」状態になりジョブネットの実行が終了します。

(c) 実行順序制御方式の一時変更機能

再起動時に接続先のジョブネットをジョブネットコネクタと同期して実行させる場合や，ある世代に対して設定を一時的に変更する場合には，実行順序制御方式の一時変更機能を使用できます。

実行順序制御方式の一時変更機能は，次のウィンドウの［操作］－［実行順序制御方式の一時変更］で実行できます。

- ［デイリースケジュール（階層表示）］ウィンドウ
- ［マンスリースケジュール］ウィンドウ
- ［ジョブネットモニタ］ウィンドウ

補足事項

擬似予定に対しては，実行順序制御方式を一時変更できません。

(4) ジョブネットコネクタおよび接続先のジョブネットの状態遷移

ジョブネットコネクタおよび接続先のジョブネットの状態遷移について説明します。なお，ジョブネットコネクタおよび接続先のジョブネットが持つ状態については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　6.1 ジョブネット・ジョブ・ジョブネットコネクタの状態」を参照してください。

(a)　ジョブネットコネクタの状態遷移

ジョブネットコネクタを定義したジョブネットを実行登録すると，ジョブネットコネクタを定義したルートジョブネットが実行開始します。ジョブネットコネクタの状態は先行ユニットがすべて終了した時点で，接続する世代の状態に応じて遷移します。

ジョブネットコネクタの状態遷移を次の表に示します。

表 2-6　ジョブネットコネクタの状態遷移

接続先のジョブネットの状態	ジョブネットコネクタの状態
未登録	異常検出実行中
（世代なし）※1	計画未実行
未計画	計画未実行
開始時刻待ち	実行中
保留中	実行中
繰り越し未実行	異常検出実行中※2
正常終了	正常終了
警告検出終了	警告検出終了
異常検出終了	異常検出実行中※2
中断	異常検出実行中※2
強制終了	異常検出実行中※2
閉塞	異常検出実行中
順序不正	異常検出実行中※2
実行中	実行中
警告検出実行中	警告検出実行中
異常検出実行中	異常検出実行中

注※1

実行登録済みで，ジョブネットコネクタと接続する世代がない場合です。接続ルールについては，「(2) ジョブネットコネクタと接続先のジョブネットの接続ルール」を参照してください。

注※2

異常検出実行中の状態で，接続先のジョブネットのうち，終了状態の世代が保存世代数管理や登録解除で削除されると，ジョブネットコネクタは異常検出終了になります。

なお，次のようにジョブネットコネクタの定義が不正だった場合は，すぐに異常検出終了になります。

- 接続先のジョブネットを指定していない

- 接続先のジョブネットに指定したユニットがない
- 接続先のジョブネットに指定したユニットがルートジョブネットまたはプランニンググループでない
- 接続先のジョブネットとして指定したユニットの定義に，実行順序制御を「する」と設定していない
- 接続先のジョブネットとして指定したユニットの定義で，ジョブネットコネクタ名の設定に誤りがある
- 接続先のジョブネットに指定したユニットと接続範囲の指定が一致していない
- 接続範囲に「別サービス」を指定している場合で，次に示す不正な指定をしている
  - 接続ホスト名に指定したホストに接続できない
  - 接続サービス名に指定したスケジューラーサービスがない
  - 接続先のジョブネットに指定したユニットの接続ホスト名に指定したホストが，このジョブネットコネクタのあるホストと異なる
  - 接続先のジョブネットに指定したユニットの接続サービス名に指定したスケジューラーサービスが，このジョブネットコネクタのあるスケジューラーサービスと異なる

(b)　接続先のジョブネットの状態遷移

接続先のジョブネットを実行登録すると，接続する世代の状態に応じて，実行順序制御方式（同期・非同期）ごとに実行開始時の動作が変わります。

接続先のジョブネットの状態遷移を次の表に示します。

表 2-7　接続先のジョブネットの状態遷移

ジョブネットコネクタの状態	接続先のジョブネットの状態	
	同期	非同期
未登録	開始時刻待ち	実行中
（世代なし）※1	実行中	実行中
未計画	開始時刻待ち	実行中
先行終了待ち	開始時刻待ち	実行中
未実行終了	開始時刻待ち※2	実行中
計画未実行	実行中	実行中
実行中	実行中	実行中
正常終了	開始時刻待ち※2	実行中
警告検出終了	開始時刻待ち※2	実行中
異常検出終了	開始時刻待ち※2	実行中
強制終了	開始時刻待ち※2	実行中
閉塞	開始時刻待ち※2	実行中

ジョブネットコネクタの状態	接続先のジョブネットの状態	
	同期	非同期
終了状態不明	開始時刻待ち※2	実行中

注※1

実行登録済みで，接続先のジョブネットと接続する世代がない場合です。接続ルールについては，「(2) ジョブネットコネクタと接続先のジョブネットの接続ルール」を参照してください。

注※2

接続先のジョブネットの実行開始時に，状態変更操作などですでにジョブネットコネクタが終了状態の場合，接続先のジョブネットはジョブネットコネクタが実行中になるまで「開始時刻待ち」の状態になります。この場合，ジョブネットコネクタを再実行するか，または接続先のジョブネットの実行順序制御方式を「非同期」に一時変更します。

なお，次のように接続先のジョブネットの定義が不正だった場合は，すぐに異常検出終了になります。ただし，接続範囲に「別サービス」を指定していて，実行順序制御方式が「非同期」の場合は，定義不正であっても異常検出終了にはならないで，実行を開始します。そのため，実行登録する前に定義内容の事前チェック機能を使用して定義を確認することを推奨します。

- 実行順序制御を「する」と設定していて，ジョブネットコネクタ名を指定していない
- ジョブネットコネクタ名に指定したユニットがない
- ジョブネットコネクタ名に指定したユニットがジョブネットコネクタでない
- ジョブネットコネクタ名に指定したジョブネットコネクタに別の接続先が設定されている
- ジョブネットコネクタ名に指定したユニットと接続範囲の指定が一致していない
- 接続範囲に「別サービス」を指定している場合で，次に示す不正な指定をしている
  - 接続ホスト名に指定したホストに接続できない
  - 接続サービス名に指定したスケジューラーサービスがない
  - ジョブネットコネクタ名に指定したユニットの接続ホスト名に指定したホストが，この接続先のジョブネットのあるホストと異なる
  - ジョブネットコネクタ名に指定したユニットの接続サービス名に指定したスケジューラーサービスが，この接続先のジョブネットのあるスケジューラーサービスと異なる

(c) 異なるスケジューラーサービス間の通信処理に失敗した場合の状態

異なるスケジューラーサービス間のルートジョブネットの実行順序制御では，次のときに通信処理を行います。

- ジョブネットコネクタが実行開始するとき
- 接続先のジョブネットが実行開始するとき
- 接続先のジョブネットが状態遷移するとき

このときに通信エラーが発生した場合の，ジョブネットコネクタおよび接続先のジョブネットの状態について次に示します。なお，エラーが発生した場合の対処については，マニュアル「JP1/Automatic Job Management System 3 トラブルシューティング　2.6.5 ジョブネットコネクタのトラブルへの対処」を参照してください。

■ ジョブネットコネクタの実行開始時に通信エラーとなった場合

ジョブネットコネクタが実行開始するときに，接続先のジョブネットと世代の接続を行います。世代の接続については，「(2) ジョブネットコネクタと接続先のジョブネットの接続ルール」を参照してください。

この，世代の接続の際に通信エラーが発生した場合，ジョブネットコネクタは「異常検出終了」となります。接続先のジョブネットの状態は遷移しません。

■ 接続先のジョブネットの実行開始時に通信エラーとなった場合

接続先のジョブネットの実行順序制御方式が「同期」の場合，実行開始するときにジョブネットコネクタと世代の接続を行います。世代の接続については，「(2) ジョブネットコネクタと接続先のジョブネットの接続ルール」を参照してください。

この，世代の接続の際に通信エラーが発生した場合，接続先のジョブネットは「異常検出終了」となります。ジョブネットコネクタの状態は遷移しません。

■ 接続先のジョブネットの状態遷移時に通信エラーとなった場合

接続先のジョブネットが状態遷移しても，ジョブネットコネクタが接続先のジョブネットの状態を確認できないため，ジョブネットコネクタの状態は遷移しません。

(5)　ジョブネットコネクタ・接続先のジョブネットの再実行

ジョブネットコネクタおよび接続先のジョブネットの再実行について説明します。

(a)　ジョブネットコネクタの再実行

ジョブネットコネクタは，ジョブと同様に終了状態の場合だけ再実行できます。ただし，ジョブネットコネクタは保留属性を持たないため，再実行時に保留状態にできません。

(b)　接続先のジョブネットの再実行

接続先のジョブネットは，通常のジョブネットと同様に再実行できます。ただし，実行順序制御方式を「同期」と設定していて，一度実行した場合，実行順序制御方式は「非同期」となり再実行時にジョブネットコネクタと同期しません。再実行時にもジョブネットコネクタと同期して実行させる場合は，実行順序制御方式の一時変更機能で「同期」に変更してください。

(c)　接続関係が途切れた場合の再実行時の動作

登録解除や実行中止，保存世代数管理などによって，一度接続関係が成立した世代のどちらか片方が削除された場合，接続関係が途切れた状態になり，もう一方の世代を再実行できません。どちらか一方の世代が削除された場合に再実行したときの動作とその対処方法を次の表に示します。

表 2-8　接続関係が途切れた場合の再実行時の動作と対処方法

世代	再実行時の動作	対処方法
ジョブネットコネクタ側の世代が残っている場合	ジョブネットコネクタを再実行すると，ジョブネットコネクタが「異常検出終了」※となる。	ジョブネットコネクタの次のユニットから再実行する。
接続先のジョブネットの世代が残っている場合	実行順序制御方式を「同期」にして接続先のジョブネットを再実行すると，接続先のジョブネットが「異常検出終了」※となる。	実行順序制御方式を「非同期」に一時変更して再実行する。

注※

終了したあとに再実行した場合は異常検出終了となりますが，実行状態で再実行した場合は，異常検出終了にならないで再実行できます。

(6)　ジョブネットコネクタのポーリング

異なるスケジューラーサービス間のルートジョブネットの実行順序を制御する場合，ジョブネットコネクタおよび実行順序制御方式に「同期」を設定した接続先のジョブネットは，次の間ポーリングし，定期的に接続相手となるユニットの状態を確認します。

- ジョブネットコネクタが実行開始してから終了するまで
- 接続先のジョブネットが開始時刻に到達してから実行開始するまで

スケジューラーサービスの停止や通信障害によって，ポーリングによる接続相手の状態取得ができない場合，ジョブネットコネクタ，または接続先のジョブネットは異常検出終了します。

2.2.5　待ち合わせ条件を使用したユニットの実行順序制御

異なるジョブネット間のユニットの実行順序を制御するには，**待ち合わせ条件**を使用します。待ち合わせ条件を設定したユニットは，待ち合わせ条件で指定したユニットの実行終了を待ち合わせてから実行を開始します。待ち合わせ条件を使用することで，異なるジョブネット間のユニットの実行順序を制御できます。

待ち合わせ条件を使用する場合，環境設定パラメーターPREWAITUSE で待ち合わせ条件を使用できるように設定しておく必要があります。また，待ち合わせできるユニット種別を拡張する場合は，必要に応じて環境設定パラメーターPREWAITEXTEND を設定します。

(1)　待ち合わせ条件を使用したユニットの実行順序制御の概要

待ち合わせ条件を使用したユニットの実行順序制御の概要と，待ち合わせ条件を使用してユニットの実行順序を制御する流れについて説明します。

(a) 待ち合わせ条件付きユニットと待ち合わせ対象ユニット

待ち合わせ条件を設定するユニットの待ち合わせ条件には，実行終了を待ち合わせるユニット名を指定します。ここでいう「実行終了」に該当する状態は，起動条件を使用するかどうか※によって異なります。

起動条件を使用しない※ユニットの場合

「実行終了」とは，次の状態に遷移することをいいます。

- 正常終了
- 警告検出終了
- 計画未実行

起動条件を使用する※ユニットの場合

「実行終了」とは，デフォルトの設定では待ち合わせ対象の監視世代およびそのすべての実行世代が次の状態に遷移することをいいます。

監視世代

- 監視正常終了

実行世代

- 正常終了
- 警告検出終了

待ち合わせ対象ユニットに起動条件を使用する場合，「実行終了」に該当する状態を変更できます。詳細については，「(5)(c) 待ち合わせ対象ユニットに起動条件を使用する場合の，待ち合わせ条件の設定」を参照してください。

注※

「起動条件を使用する」とは，ルートジョブネットが次の条件を満たしていることを指します。

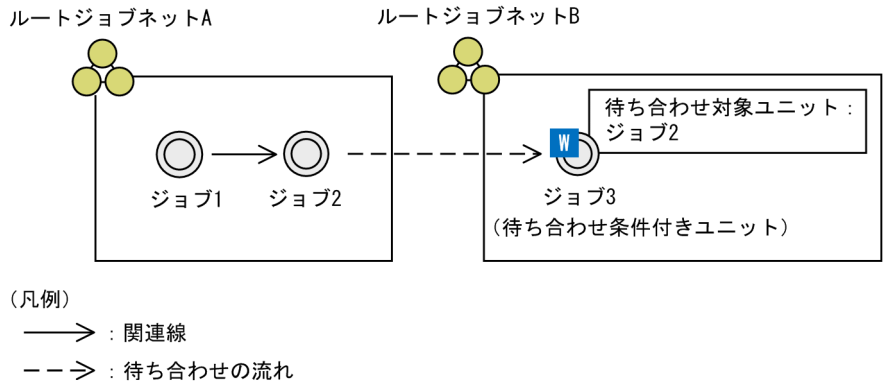
- 起動条件付きジョブネットである。
- 実行登録時に，起動条件を使用する設定で実行登録している。
  即時実行の場合は，実行登録時に設定します。
  計画実行または確定実行の場合は，スケジュールルールで設定します。
- ajsplan コマンドの-j オプションの設定で，起動条件を無効にしていない。

待ち合わせ条件で指定したユニットが実行終了すると，待ち合わせ条件を設定したユニットが実行を開始します。

待ち合わせ条件を設定したユニットのことを**待ち合わせ条件付きユニット**，待ち合わせ条件で指定したユニットのことを**待ち合わせ対象ユニット**と呼びます。

待ち合わせ条件による実行順序制御を，次の図に示します。

図 2-26　待ち合わせ条件による実行順序制御



この例では，ルートジョブネット B 配下のジョブ 3 に待ち合わせ条件を設定しています。待ち合わせ条件には，待ち合わせ対象ユニットとしてルートジョブネット A 配下のジョブ 2 を指定しています。ルートジョブネット A およびルートジョブネット B を実行すると，ジョブ 3 はジョブ 2 の実行終了を待ち合わせます。ジョブ 2 が実行終了すると，ジョブ 3 が実行を開始します。

待ち合わせ条件を使うと，さまざまなユニット間の実行順序を制御できます。待ち合わせ条件を設定できるユニット，および待ち合わせ対象ユニットとして指定できるユニットを，次の表に示します。

表 2-9　待ち合わせ条件を設定できるユニットおよび待ち合わせ対象ユニットとして指定できるユニット

項番	ユニット種別		待ち合わせ条件を設定できるユニット	待ち合わせ対象ユニットとして指定できるユニット
1	ジョブグループ	ジョブグループ自体	×	×
2		ジョブグループ直下のジョブ	○※1	×
3	ルートジョブネット	起動条件を使用しないルートジョブネット	○	○
4		起動条件を使用しないルートジョブネットの配下のユニット	○※2	○※2
5		起動条件を使用するルートジョブネット	○※3	○※3
6		起動条件を使用するルートジョブネットの配下のユニット	×※2	×※2
7	起動条件（.CONDITION）		×	×
8	ネストジョブネット		○	○
9	リモートジョブネット	リモートジョブネット自体	×	×

項番	ユニット種別		待ち合わせ条件を設定できるユニット	待ち合わせ対象ユニットとして指定できるユニット
10	リモートジョブネット	リモートジョブネットの配下のユニット	×	×
11	プランニンググループ	プランニンググループ自体	×	○
12		プランニンググループ直下のルートジョブネット	○	×
13		プランニンググループ直下のルートジョブネットの配下のユニット	○	○※4
14	標準ジョブ		○	○
15	ジョブネットコネクタ		○	○
16	OR ジョブ		×	×
17	判定ジョブ		×	×
18	イベントジョブ	ジョブネット配下のイベントジョブ	○	○
19		起動条件（.CONDITION）中のイベントジョブ	×	×
20	アクションジョブ		○	○
21	カスタムジョブ		○	○
22	引き継ぎ情報設定ジョブ		○	○
23	HTTP 接続ジョブ		○	○

（凡例）

○：待ち合わせ条件を設定できる，または待ち合わせ対象ユニットとして指定できる

×：待ち合わせ条件を設定できない，または待ち合わせ対象ユニットとして指定できない

注※1

待ち合わせ条件を設定することはできますが，実行登録できません。

注※2

起動条件付きジョブネットであっても，即時実行登録時のオプションやスケジュールルールの定義によって，起動条件を使用しないで実行するようにできます。そのため，起動条件付きジョブネットの配下のユニットであっても，実行するまでは起動条件が使用されるかどうか分からないため，待ち合わせ条件を定義することはできます。同様に，起動条件付きジョブネット配下のユニットであっても，待ち合わせ対象ユニットとして指定できます。しかし，実行時に起動条件を使用すると，実行エラーになります。

注※3

起動条件を使用するジョブネットに待ち合わせ条件を設定したり，待ち合わせ対象ユニットとして指定したりするには，次の条件を満たしている必要があります。

- JP1/AJS3 - Manager がバージョン 10-00 以降である。
- 環境設定パラメーターPREWAITEXTEND で「condition」を設定している。

注※4

プランニンググループ直下のルートジョブネット配下のユニットを待ち合わせ対象ユニットとして指定する場合，ルートジョブネット名を省略した形式で指定してください。例えば，プランニンググループ直下にあるルートジョブネット配下のジョブを待ち合わせ対象ユニットとして指定する場合は，「/プランニンググループ/ルートジョブネット/ジョブ」ではなく，「/プランニンググループ/ジョブ」と指定します。

なお，待ち合わせ条件は，同一スケジューラーサービス内のユニットの実行順序を制御します。そのため，待ち合わせ条件付きユニットと待ち合わせ対象ユニットは，同一のスケジューラーサービス配下にある必要があります。

注意事項

- 待ち合わせ条件を使用すると，ジョブフローが確認しにくくなり，業務の流れやジョブネットの階層が把握しづらくなります。ジョブフローを作成する際は，ジョブネットを入れ子にしたり，複数のジョブを一つにまとめたりするなど，できるだけ待ち合わせ条件を使用しないで，関連線を使用することを先に検討してください。
- 待ち合わせ条件を使用する場合は，複数の待ち合わせが同一時間帯に集中して開始しないように，ジョブネットの開始予定時刻をずらすなど，待ち合わせが開始する時間帯を分散させてください。

  同一時間帯に複数の待ち合わせ条件付きユニットが待ち合わせを開始する場合は，待ち合わせ対象ユニットが終了してから待ち合わせ条件付きユニットが実行開始するまでの所要時間について，次の点を考慮してジョブフローを設計してください。

  ・待ち合わせ条件付きユニットは，待ち合わせ対象ユニットの状態確認を行います。そのため，一つの待ち合わせ対象ユニットが終了してから待ち合わせ条件付きユニットが実行を開始するまで，時間が掛かります。その時間は，同一スケジューラーサービス内のほかの待ち合わせ条件付きユニットも含めた，同一時間帯に待ち合わせをしている待ち合わせ条件付きユニット数に比例します。

  ・一つのスケジューラーサービス内で 1 秒間に処理できる待ち合わせ条件付きユニットの数は，4 件から 20 件程度です。例えば，待ち合わせ条件付きユニット 100 個が同時に一つの待ち合わせ対象ユニットとの待ち合わせを開始した場合，待ち合わせ対象ユニットの実行が終了してから 100 個目の待ち合わせ条件付きユニットが実行を開始するまで，5 秒から 25 秒程度掛かります。

  ・待ち合わせ対象ユニットに起動条件付きジョブネットを指定している場合，待ち合わせ条件付きユニットが実行を開始するまでに掛かる時間は，起動条件付きジョブネットの実行世代数に依存します。例えば，待ち合わせ条件付きユニット 100 個が，実行世代が 10 世代ある起動条件付きジョブネット 100 個との待ち合わせを同時に開始した場合，起動条件付きジョブネットの実行が終了してから待ち合わせ条件付きユニットが実行開始するまでに掛かる時間は，5 秒から 20 秒程度です。起動条件付きジョブネットの実行世代が 100 世代ある場合だと，50 秒から 100 秒程度掛かります。

・待ち合わせ条件付きユニットが実行を開始するまでの所要時間は，マシン性能や，待ち合わせ対象ユニットと待ち合わせ条件付きユニットのジョブネット構成などによって変動します。

- 待ち合わせ条件は，JP1/AJS3 - View および JP1/AJS3 - Manager のバージョンが 09-50 以降の場合に使用できます。ただし，09-50 以降の JP1/AJS3 - Manager を使用していても，データベースが互換用 ISAM 構成の場合は，待ち合わせ条件は使用できません。

(b) 待ち合わせ状態

待ち合わせ対象ユニットの実行終了を待ち合わせ条件付きユニットが把握できているかどうかを，待ち合わせ状態として表示します。待ち合わせ状態には次の状態があります。

表 2-10 待ち合わせ状態一覧

項番	待ち合わせ状態	意味
1	未完了	待ち合わせ対象ユニットの実行終了の待ち合わせがまだ完了していません。
2	完了	待ち合わせ対象ユニットの実行終了の待ち合わせが完了しました。
3	未完了（手動）	待ち合わせ対象ユニットの実行終了の待ち合わせがまだ完了していません（待ち合わせ条件の有効化によって待ち合わせ状態が未完了に変更されています）。
4	完了（手動）	待ち合わせ対象ユニットの実行終了の待ち合わせが完了しました（待ち合わせ条件の無効化によって待ち合わせ状態が完了に変更されています）。
5	未完了（再実行）	待ち合わせ対象ユニットの実行終了の待ち合わせがまだ完了していません（待ち合わせ対象ユニットは，後続ユニットの再実行によって異常終了から終了状態に変更されています）。

待ち合わせ状態が「完了」または「完了（手動）」になることで，待ち合わせ条件が成立し，待ち合わせ条件付きユニットが実行を開始します。

待ち合わせ条件付きユニットの動作を設定することができます。詳細については，「2.2.5(5) 待ち合わせ条件付きユニットの動作の設定」を参照してください。

(c) 実行順序制御の流れ

待ち合わせ条件を使ったユニット間の実行順序制御は，次の流れで行います。

1. 待ち合わせ対象ユニットおよび待ち合わせ条件を設定するユニットを，それぞれ定義する。
   ユニットの詳細定義およびスケジュールを定義します。
2. 待ち合わせ条件を設定するユニットに待ち合わせ条件を設定する。
   待ち合わせ条件には，次の項目を設定します。
   - 待ち合わせ対象ユニット名
   - 待ち合わせ方法
   - 待ち合わせ対象ユニットに待ち合わせる世代がないときの動作

「待ち合わせ方法」および「待ち合わせ対象ユニットに待ち合わせる世代がないときの動作」の詳細については，「2.2.5(5) 待ち合わせ条件付きユニットの動作の設定」を参照してください。

待ち合わせ条件の設定方法については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　5.2.6 待ち合わせ条件を設定する」を参照してください。

3. 待ち合わせ条件付きユニットおよび待ち合わせ対象ユニットを管理するルートジョブネットを，それぞれ実行登録する。

   待ち合わせ条件付きユニットが，待ち合わせ対象ユニットの実行終了を待ち合わせます。

   ルートジョブネットの実行登録方法については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　7. ジョブネットの実行」を参照してください。

(2) 待ち合わせ条件付きユニットと待ち合わせ対象ユニットの待ち合わせルール

待ち合わせ条件付きユニットを実行登録すると，待ち合わせ条件付きユニットは，待ち合わせルールに従って待ち合わせ対象ユニットの実行終了を待ち合わせます。

待ち合わせルールは，次に示す場合で異なります。

- 待ち合わせ条件付きユニットと待ち合わせ対象ユニットが同一のルートジョブネット配下にある場合
- 待ち合わせ条件付きユニットと待ち合わせ対象ユニットが異なるルートジョブネット配下にある場合
- 待ち合わせ条件付きユニットまたは待ち合わせ対象ユニットが起動条件を使用する場合
- 待ち合わせ対象ユニットにプランニンググループを指定した場合

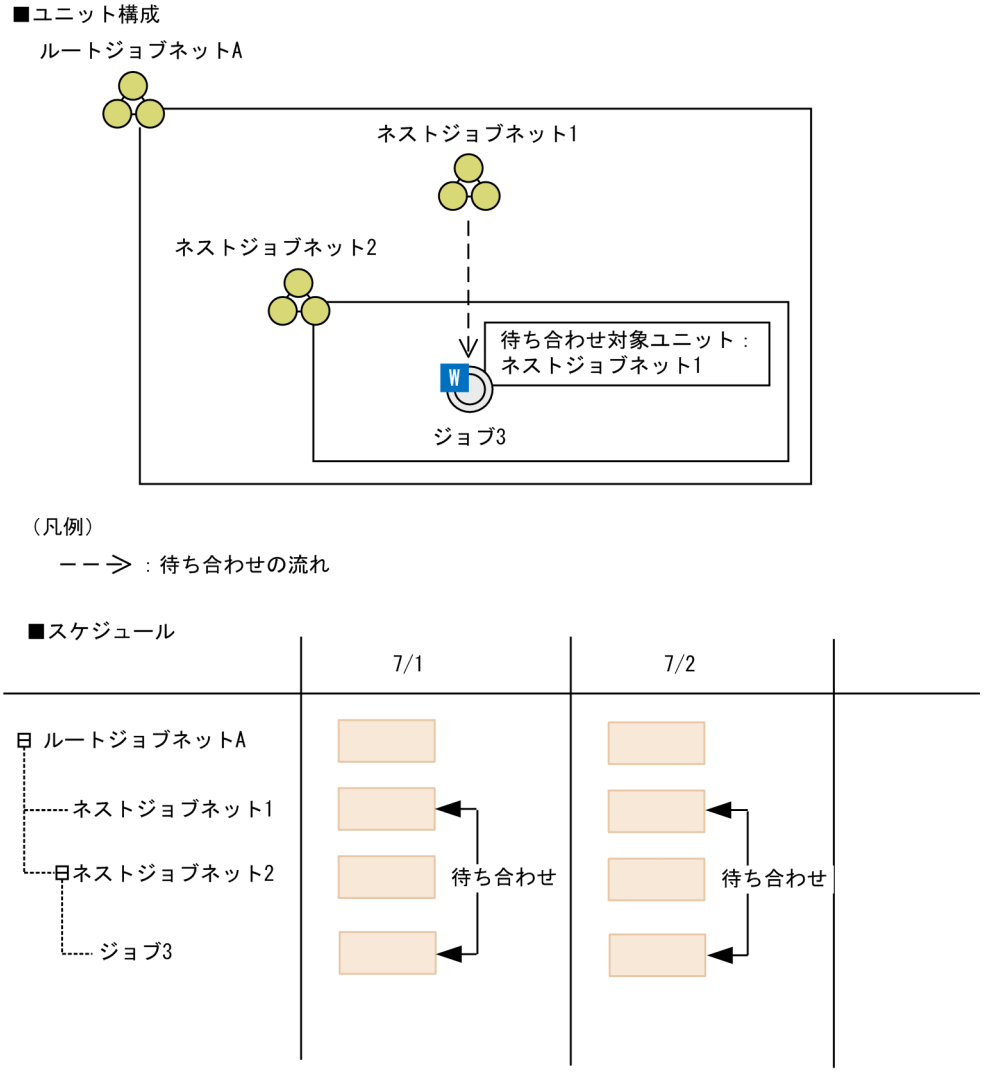
それぞれの場合について説明します。

(a) 待ち合わせ条件付きユニットと待ち合わせ対象ユニットが同一のルートジョブネット配下にある場合

待ち合わせ条件付きユニットと待ち合わせ対象ユニットが同一のルートジョブネット配下にある場合，各ユニットは同一の世代で実行されます。そのため，待ち合わせ条件付きユニットは，同じ世代で実行される待ち合わせ対象ユニットの実行終了を待ち合わせます。

待ち合わせ条件付きユニットと待ち合わせ対象ユニットが同一ルートジョブネット配下にある場合の例を，次の図に示します。

図 2-27　待ち合わせ条件付きユニットと待ち合わせ対象ユニットが同一ルートジョブネット配下にある場合の例



この例では，待ち合わせ条件付きユニットのジョブ 3 と待ち合わせ対象ユニットのネストジョブネット 1 が，どちらもルートジョブネット A の配下に定義されています。ルートジョブネット A を 1 日に 1 回実行するようにスケジュールして実行登録すると，7/1 と 7/2 にルートジョブネット A の実行予定世代が一つずつ生成されます。この場合，ジョブ 3 は，同一世代で実行するネストジョブネット 1 の実行終了を待ち合わせます。

(b)　待ち合わせ条件付きユニットと待ち合わせ対象ユニットが異なるルートジョブネット配下にある場合

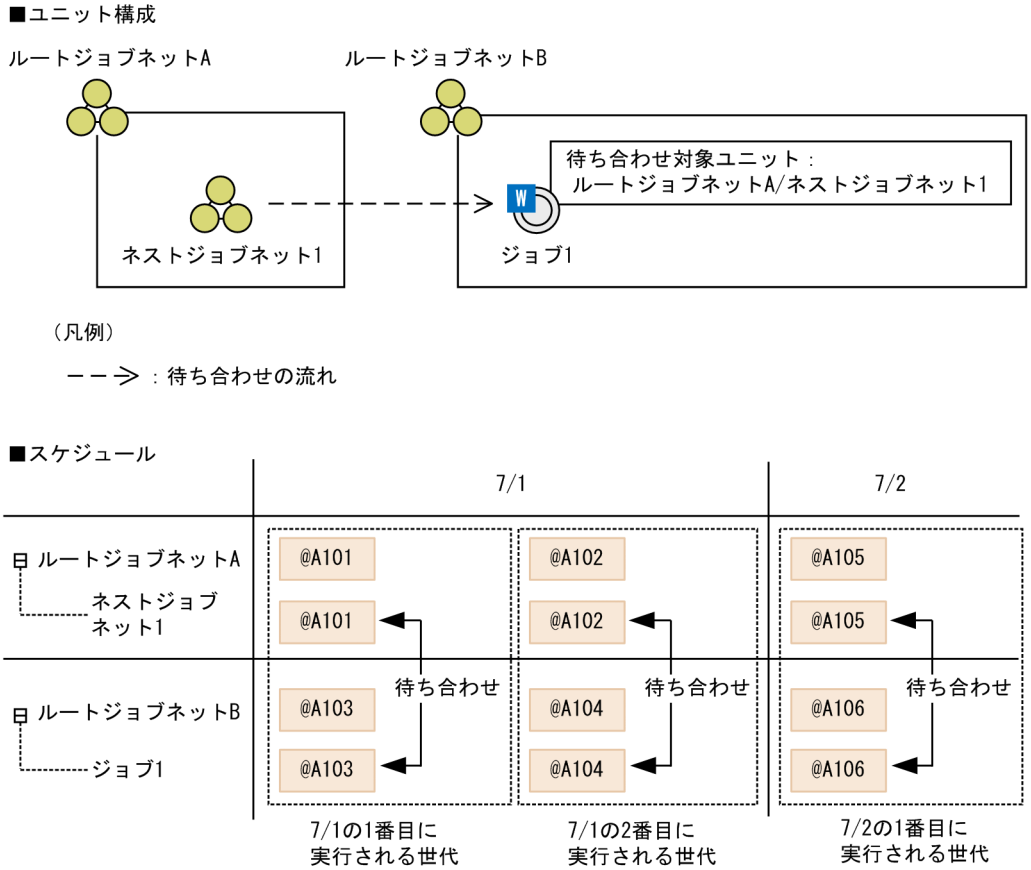
待ち合わせ条件付きユニットと待ち合わせ対象ユニットが異なるルートジョブネット配下にある場合，各ルートジョブネットを実行登録すると，それぞれのルートジョブネットに実行予定世代が生成されます。このとき，待ち合わせ条件付きユニットの実行予定世代は，次の条件を満たす待ち合わせ対象ユニットの実行予定世代の実行終了を待ち合わせます。

- 実行日が同じ

- 実行予定を開始予定時刻順に並べた場合に，開始予定の順番が等しい

待ち合わせ対象ユニットが待ち合わせ条件付きユニットと異なるルートジョブネット配下にある場合の例を，次の図に示します。

図 2-28　待ち合わせ対象ユニットが待ち合わせ条件付きユニットと異なるルートジョブネット配下にある場合の例



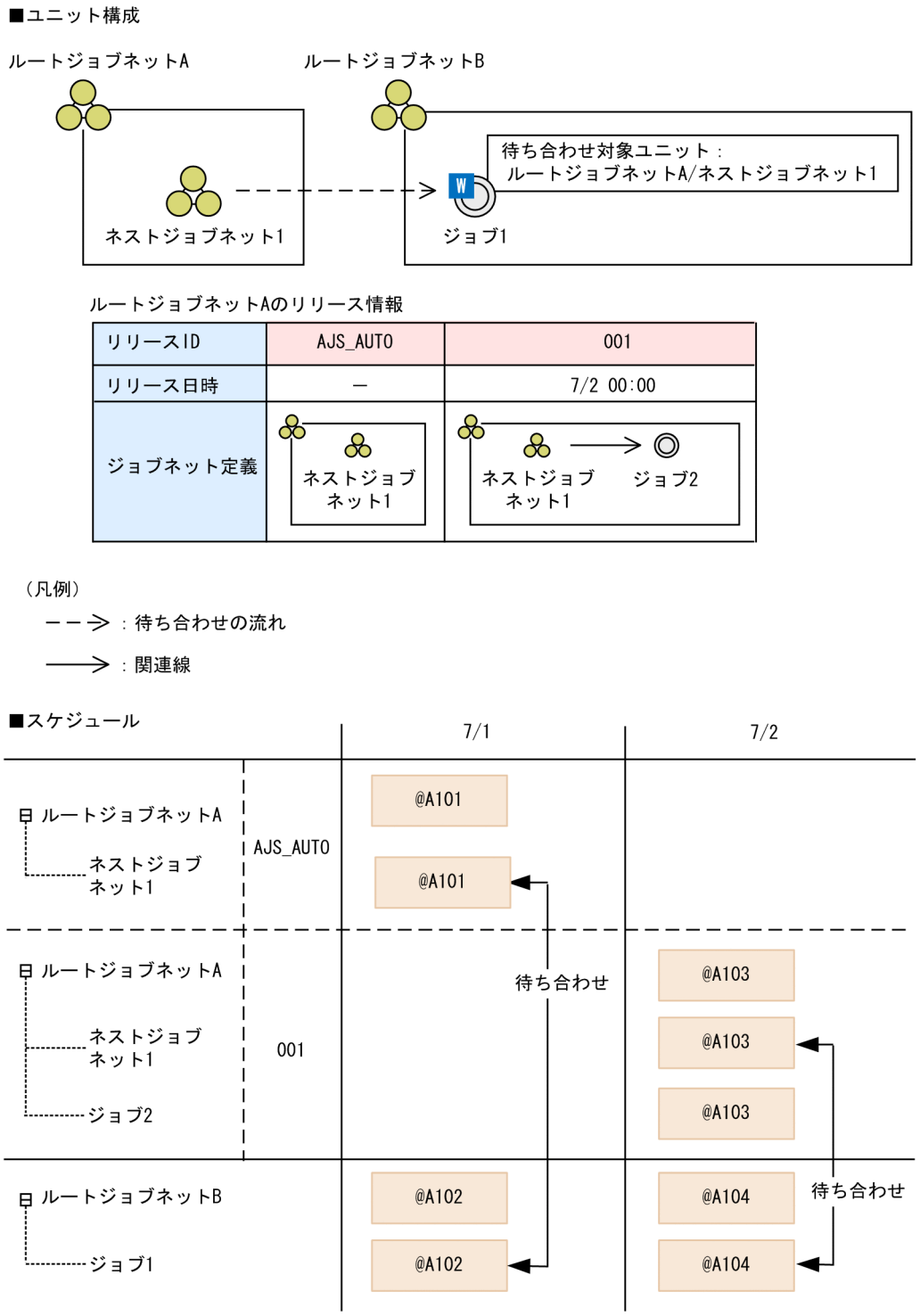
この例では，待ち合わせ条件付きユニットのジョブ 1 が，待ち合わせ対象ユニットのネストジョブネット 1 と，異なるルートジョブネットの配下に定義されています。ルートジョブネット A およびルートジョブネット B が 7/1 に 2 回実行するようにスケジュールして実行登録すると，7/1 のルートジョブネット A の実行予定として，実行 ID が@A101 の世代と@A102 の世代が生成されます。一方，7/1 のルートジョブネット B の実行予定として，実行 ID が@A103 の世代と@A104 の世代が生成されます。ジョブ 1 の世代@A103 および@A104 は，実行日が同じで，かつ開始予定時刻の順番が等しいネストジョブネット 1 の実行予定世代を待ち合わせます。そのため，ジョブ 1 で 1 番目に実行される@A103 は，ネストジョブネット 1 で 1 番目に実行される@A101 を待ち合わせます。同様に，ジョブ 1 で 2 番目に実行される@A104 は，ネストジョブネット 1 で 2 番目に実行される@A102 を待ち合わせます。7/2 に実行されるジョブ 1 の世代も，7/2 の開始予定時刻の順番が等しいネストジョブネット 1 の実行予定世代を待ち合わせます。

次に，JP1/AJS3 の機能と待ち合わせ条件を組み合わせて使用したときの例について説明します。

ジョブネットリリース機能を使用して待ち合わせ対象ユニットのジョブネット定義を切り替えるとき

ジョブネットリリース機能を使用して，待ち合わせ対象ユニットのジョブネット定義を運用中に切り替えるとき，待ち合わせ条件付きユニットが待ち合わせるジョブネット定義も，自動で切り替わります。待ち合わせ対象ユニットをジョブネットリリースする例を，次の図に示します。

図 2-29　待ち合わせ対象ユニットをジョブネットリリースする例



この例では，ルートジョブネット A がリリース登録されており，リリース日時になると，リリース ID「001」のジョブネット定義がリリースされます。リリース日時は 7/2 の 00:00 です。ルートジョブ

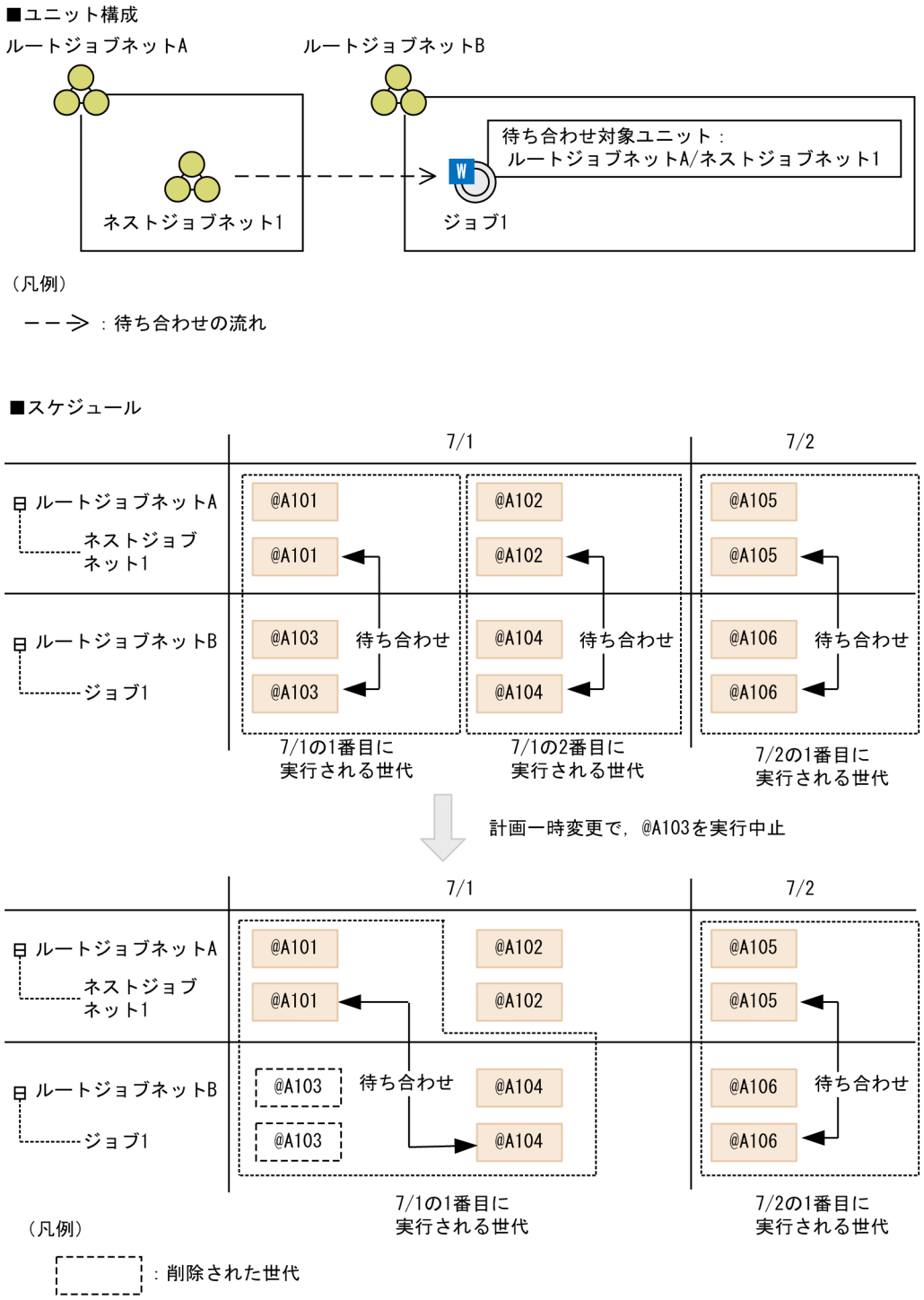
ネット A を実行登録すると，7/1 の実行予定世代@A101 はリリース ID「AJS_AUTO」のジョブネット定義で，7/2 の実行予定世代@A103 はリリース ID「001」のジョブネット定義で生成されます。この場合，ジョブ 1 は，7/1 まではリリース ID「AJS_AUTO」のネストジョブネット 1 の世代@A101 と待ち合わせ，7/2 以降はリリース ID「001」のネストジョブネット 1 の世代@A103 と待ち合わせます。

実行登録後に実行予定世代の開始予定の順番を変更するとき

実行登録後に計画一時変更などで実行予定世代の開始予定の順番を変更するとき，待ち合わせ条件付きユニットが待ち合わせる待ち合わせ対象ユニットの実行予定世代も，変更後の開始予定の順番に合わせて自動で変更されます。

計画一時変更で実行予定世代の開始予定の順番を変更する例を，次の図に示します。

図 2-30　計画一時変更で実行予定世代の開始予定の順番を変更する例



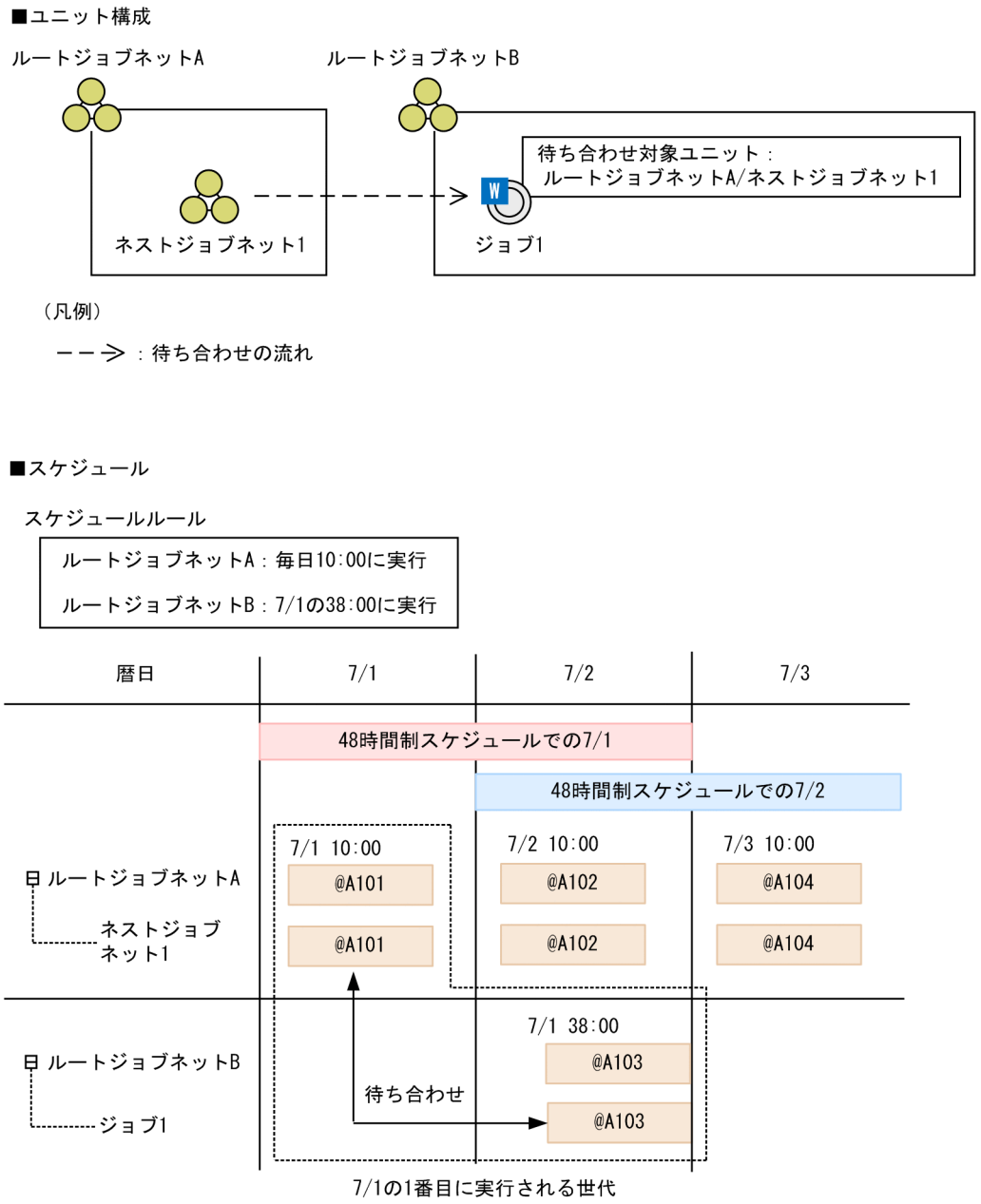
この例では，待ち合わせ対象ユニットのネストジョブネット 1 を管理するルートジョブネット A，および待ち合わせ条件付きユニットのジョブ 1 を管理するルートジョブネット B が，7/1 に 2 回実行するようにスケジュールされています。7/1 の 1 番目に実行されるルートジョブネット B の実行予定世代 @A103 を計画一時変更で実行中止すると，@A103 が削除され，7/1 の 1 番目に実行される世代は @A104 になります。そのため，ジョブ 1 の@A104 は，ネストジョブネット 1 の@A101 を待ち合わせます。

48 時間制スケジュールを採用しているとき

48 時間制スケジュールを採用しているときは，48 時間制スケジュールでの実行日に従って待ち合わせる世代が決定します。

48 時間制スケジュールを採用している例を，次の図に示します。

図 2-31　48 時間制スケジュールを採用している例



この例では，ルートジョブネット A およびルートジョブネット B のスケジューラーサービスに対して，48 時間制スケジュールを採用しています。待ち合わせ対象ユニットのネストジョブネット 1 を管理するルートジョブネット A のスケジュールルールは，毎日 10:00 に実行するように定義しています。また，待ち合わせ条件付きユニットのジョブ 1 を管理するルートジョブネット B のスケジュールルールは，7/1 の 38:00 に実行するように定義しています。この場合，7/1 の 38:00（暦日で 7/2 の 14:00）に実行されるジョブ 1 の実行予定世代@A103 は，7/1 の 1 番目に実行される世代になります。そのた

め，ジョブ 1 の@A103 の世代は，ネストジョブネット 1 の 7/1 の 1 番目に実行される世代である @A101 を待ち合わせます。

注意事項

- 基準時刻は，待ち合わせ条件付きユニットと待ち合わせ対象ユニットで別々に設定できます。しかし，それぞれのユニットで別々の設定をしていると，待ち合わせる世代の対応が複雑になり運用が困難になります。基準時刻の設定は，待ち合わせ条件付きユニットと待ち合わせ対象ユニットで合わせることを推奨します。
- UNIX の場合，タイムゾーンは，待ち合わせ条件付きユニットと待ち合わせ対象ユニットで別々に設定できます。しかし，それぞれのユニットで別々の設定をしていると，待ち合わせる世代の対応が複雑になり運用が困難になります。タイムゾーンの設定は，待ち合わせ条件付きユニットと待ち合わせ対象ユニットで合わせることを推奨します。
- 待ち合わせ対象ユニットの実行終了を安全に待ち合わせるために，待ち合わせ対象ユニットと待ち合わせ条件付きユニットの保存世代数は，少なくとも 1 日分の世代を保存できるよう設定してください。待ち合わせ対象ユニットの 1 日分の世代が保存できない場合，待ち合わせ条件付きユニットが実行を開始しないおそれがあります。また，待ち合わせ条件付きユニットの 1 日分の世代が保存できない場合，待ち合わせ条件付きユニットが意図しない待ち合わせ対象ユニットの世代を待ち合わせるおそれがあります。

(c) 待ち合わせ条件付きユニットまたは待ち合わせ対象ユニットが起動条件を使用する場合

■ 待ち合わせ対象ユニットが起動条件を使用する場合

待ち合わせ対象ユニットに指定した起動条件付きジョブネットが実行を開始すると，監視世代と実行世代の両方が生成されます。このとき，待ち合わせ条件付きユニットの実行予定世代は，待ち合わせ対象ユニットの監視世代と実行世代の両方が実行終了するのを待ち合わせます。

待ち合わせ条件付きユニットの実行予定世代は，次の条件を満たす待ち合わせ対象ユニットの世代の実行終了を待ち合わせます。

監視世代

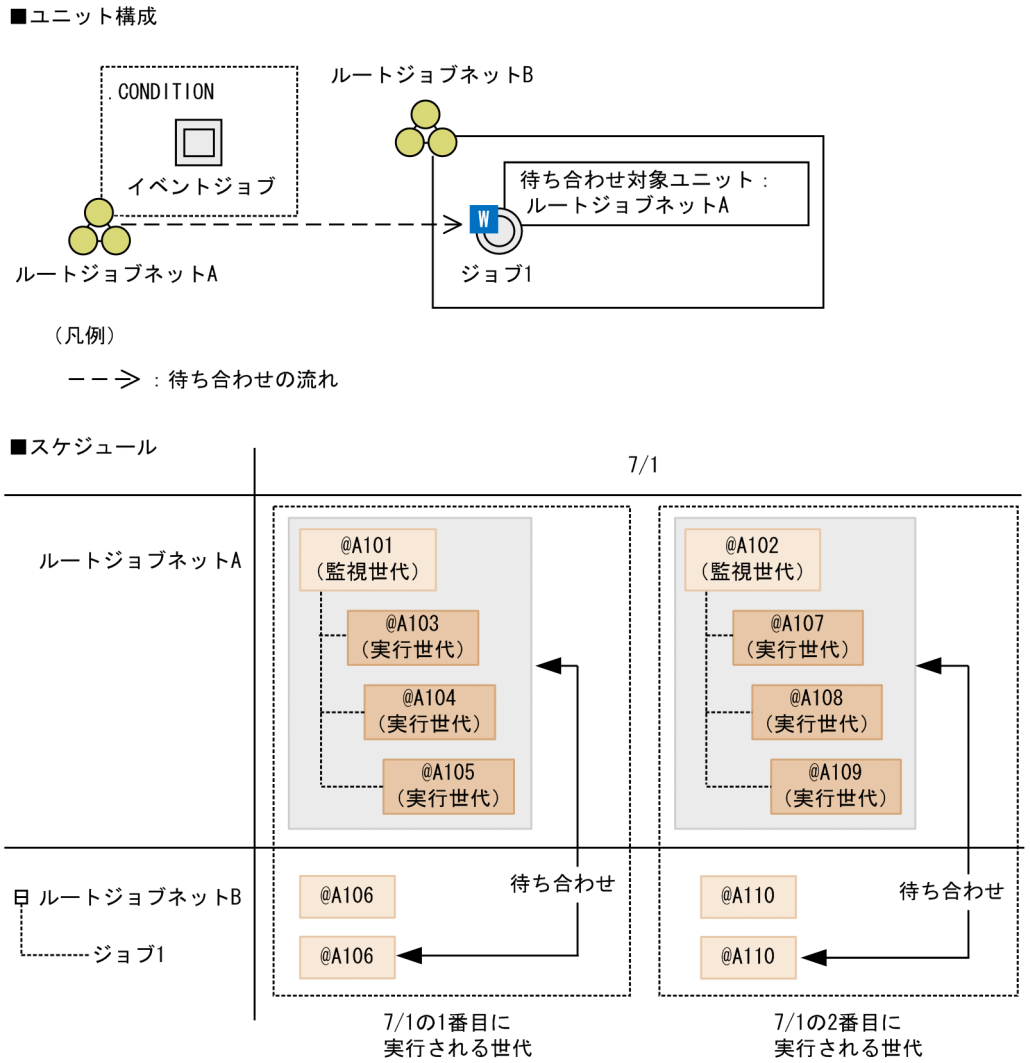
- 実行日が同じ
- 開始予定時刻順に監視世代を並べた場合に，開始予定の順番が等しい

実行世代

- 監視世代とともに生成されるすべての実行世代

待ち合わせ対象ユニットが起動条件付きジョブネットの例を，次の図に示します。

図 2-32　待ち合わせ対象ユニットが起動条件付きジョブネットの例



この例では，待ち合わせ対象ユニットとして起動条件付きジョブネットであるルートジョブネット A が定義されています。ルートジョブネット A の世代@A101 と@A102 は，それぞれ起動条件を使用する世代です。

ルートジョブネット A およびルートジョブネット B が 7/1 に 2 回実行するようにスケジュールして実行登録すると，7/1 のルートジョブネット A の実行予定として，実行 ID が@A101 の監視世代と@A103 の実行世代，および@A102 の監視世代と@A107 の実行世代が生成されます。

@A101 の監視世代が監視中の間に起動条件が成立し，実行世代@A103 が実行を開始すると，実行世代@A104 が生成されます。そのあと再度起動条件が成立すると，実行世代@A104 が実行を開始し，実行世代@A105 が生成されます。一方，7/1 のルートジョブネット B では，実行予定として実行 ID が@A106 および@A110 の世代が生成されます。ジョブ 1 の世代@A106 は，実行日が同じで，かつ開始予定の順番が等しいルートジョブネット A の監視世代および実行世代を待ち合わせます。そのため，ジョブ 1 の世代@A106 は，ルートジョブネット A の監視世代@A101 および実行世代@A103～@A105 がすべて実行終了するのを待ち合わせます。

同様に，ジョブ 1 で 2 番目に実行される世代@A110 は，ルートジョブネット A の監視世代@A102 および実行世代@A107～@A109 がすべて実行終了するのを待ち合わせます。

注意事項

- 起動条件付きジョブネットを待ち合わせ対象ユニットとして指定する場合，起動条件の有効範囲（回数または時間）には，明示的な数値を指定してください。起動条件の有効範囲が，回数・時間ともに無制限の場合，待ち合わせ条件付きユニットは待ち合わせを開始したときに異常終了します。これは，監視世代が実行終了しないで，待ち合わせ条件付きユニットが待ち続けてしまうのを防ぐためです。
- 起動条件付きジョブネットを待ち合わせ対象ユニットとして指定する場合，生成される実行世代の数を考慮して保存世代数を設定し，待ち合わせ中に保存世代数超過によって世代が削除されないようにしてください。詳細については，「(5)(c) 待ち合わせ対象ユニットに起動条件を使用する場合の，待ち合わせ条件の設定」を参照してください。
- 起動条件付きジョブネットを待ち合わせ対象ユニットとして指定する場合，起動条件を使用する世代と使用しない世代が混在しないようにしてください。例えば，起動条件付きジョブネットに複数のスケジュールルールを定義する場合は，すべてのスケジュールルールで起動条件を使用するか使用しないかを統一するようにしてください。起動条件を使用する世代と使用しない世代が混在していると，待ち合わせ条件が正しく動作しません。

■ 待ち合わせ条件付きユニットが起動条件を使用する場合

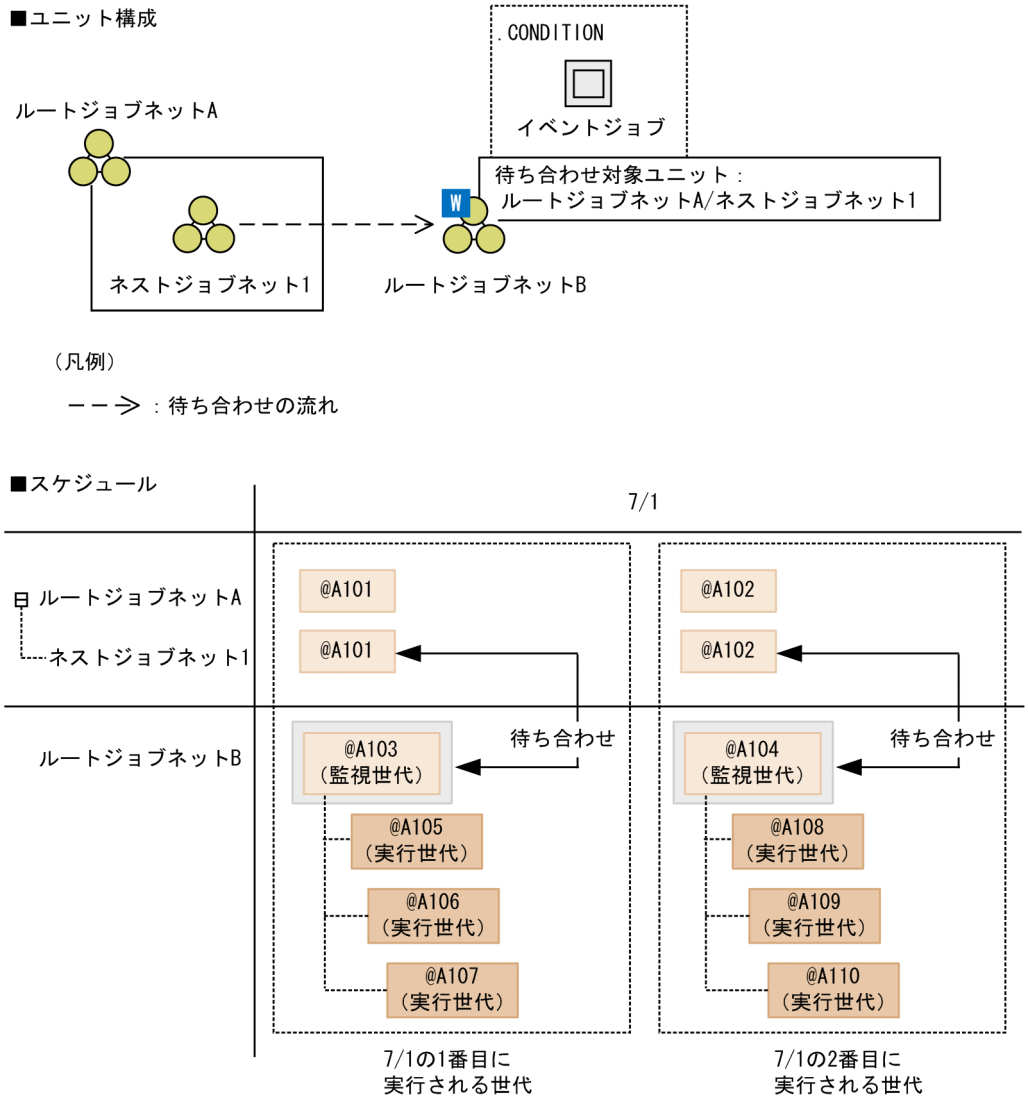
起動条件付きジョブネットが実行を開始すると，監視世代と実行世代の両方が生成されます。この場合，待ち合わせ条件付きユニットの監視世代が，待ち合わせ対象ユニットの実行終了を待ち合わせます。

待ち合わせ条件付きユニットの監視世代は，次の条件を満たす待ち合わせ対象ユニットの世代の実行終了を待ち合わせます。なお，待ち合わせ条件付きユニットの実行世代は，待ち合わせを行いません。実行世代は，起動条件が成立したときに実行を開始します。

- 実行日が同じ
- 開始予定時刻順に世代を並べた順番が，待ち合わせ条件付きユニットの監視世代を開始予定時刻順に並べた順番と等しい

待ち合わせ条件付きユニットが起動条件付きジョブネットの例を，次の図に示します。

図 2-33　待ち合わせ条件付きユニットが起動条件付きジョブネットの例



この例では，待ち合わせ条件付きユニットのルートジョブネット B が，起動条件付きジョブネットとして定義されています。

ルートジョブネット A およびルートジョブネット B が 7/1 に 2 回実行するようにスケジュールして実行登録すると，7/1 のルートジョブネット A の実行予定として，実行 ID が@A101 と@A102 の世代が生成されます。一方，7/1 のルートジョブネット B の実行予定として，実行 ID が@A103 の監視世代と@A105 の実行世代，および@A104 の監視世代と@A108 の実行世代が生成されます。監視世代@A103 は，実行日が同じで，かつ開始予定の順番が等しいネストジョブネット 1 の実行予定世代@A101 の実行終了を待ち合わせます。

世代@A101 が実行終了することで，待ち合わせ条件が成立します。待ち合わせ条件が成立すると，監視世代@A103 が起動条件の監視を開始します。そのあと，起動条件が成立して実行世代@A105 が実行を開始すると，実行世代@A106 が生成されます。再度起動条件が成立すると，実行世代@A106 が実行を開始し，実行世代@A107 が生成されます。

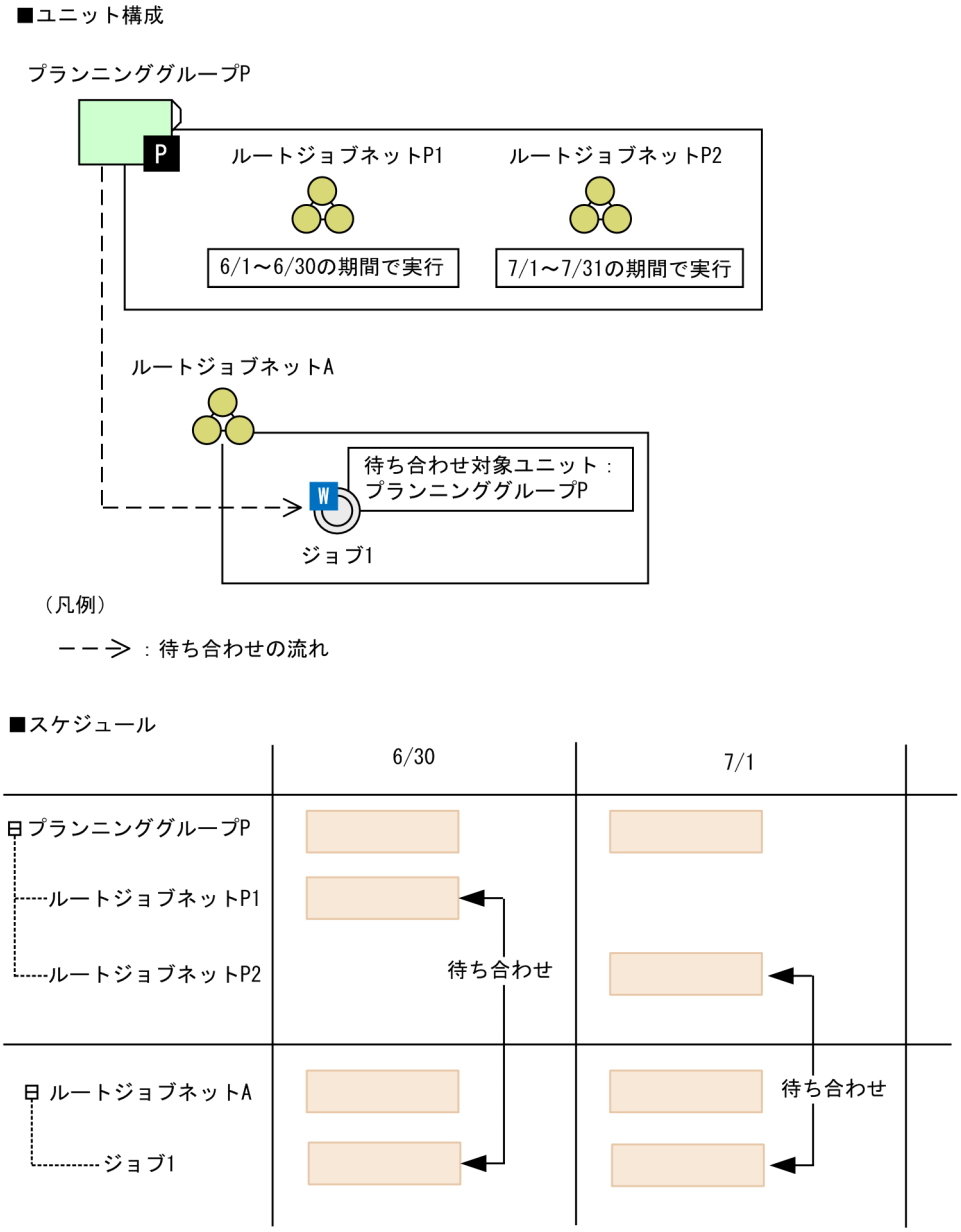
同様に，ルートジョブネット B の監視世代@A104 は，ネストジョブネット 1 で 2 番目に実行される世代@A102 の実行終了を待ち合わせます。

(d) 待ち合わせ対象ユニットにプランニンググループを指定した場合

待ち合わせ対象ユニットにプランニンググループを指定した場合，待ち合わせ条件付きユニットが待ち合わせるユニットは，プランニンググループ配下で実行しているルートジョブネットになります。プランニンググループ配下で実行しているルートジョブネットが切り替わると，待ち合わせ条件付きユニットが待ち合わせるユニットも自動で切り替わります。

待ち合わせ対象ユニットにプランニンググループを指定した場合の例を，次の図に示します。

図 2-34 待ち合わせ対象ユニットにプランニンググループを指定した場合の例



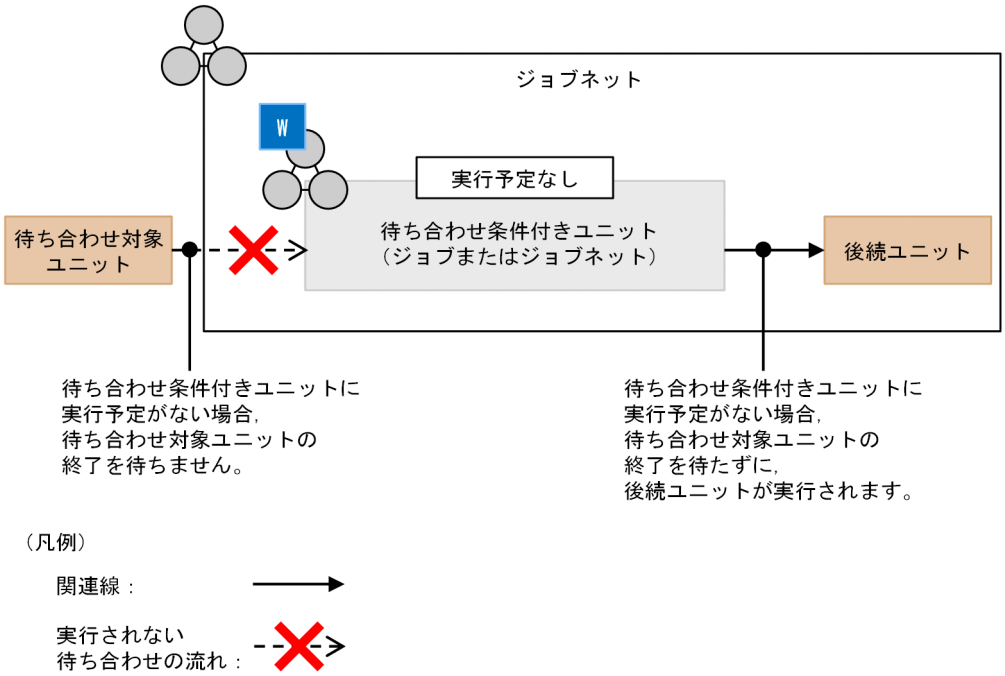
この例では，待ち合わせ対象ユニットのプランニンググループ P に，6/1～6/30 で実行するルートジョブネット P1 と，7/1～7/31 で実行するルートジョブネット P2 を定義しています。6/1～6/30 では，ジョブ 1 はルートジョブネット P1 を待ち合わせます。7/1 になってプランニンググループ配下で実行するルートジョブネットがルートジョブネット P2 に切り替わると，ジョブ 1 はルートジョブネット P2 を待ち合わせます。

(3)　待ち合わせ条件付きユニットに実行予定がない場合の動作

待ち合わせ条件付きユニットに実行予定がない場合，デフォルトの設定では待ち合わせは行われません。後続のユニットは，待ち合わせ対象ユニットの終了を待たずに実行されます。

待ち合わせ条件付きユニットに実行予定がない場合の動作を次の図に示します。

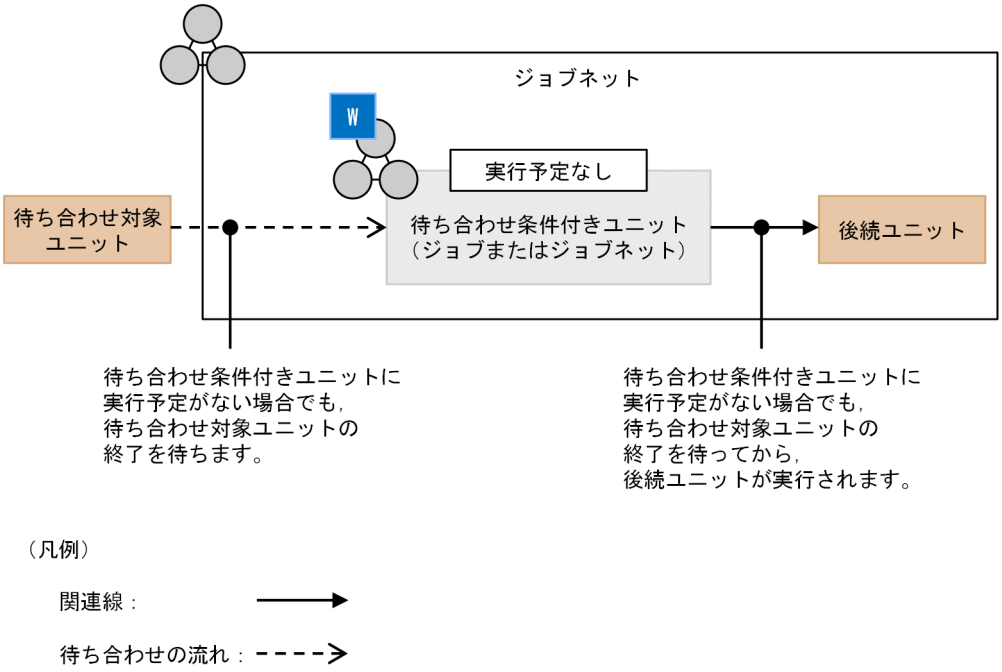
図 2-35　待ち合わせ条件付きユニットに実行予定がない場合の動作



この動作は，環境設定パラメーターPREWAITNOSCHUNITS で変更できます。環境設定パラメーターPREWAITNOSCHUNITS に「yes」を指定すると，待ち合わせ条件付きユニットに実行予定がない場合でも待ち合わせをします。環境設定パラメーターPREWAITNOSCHUNITS については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2(122) PREWAITNOSCHUNITS」を参照してください。

環境設定パラメーターPREWAITNOSCHUNITS に「yes」を指定した場合の動作を，次の図に示します。

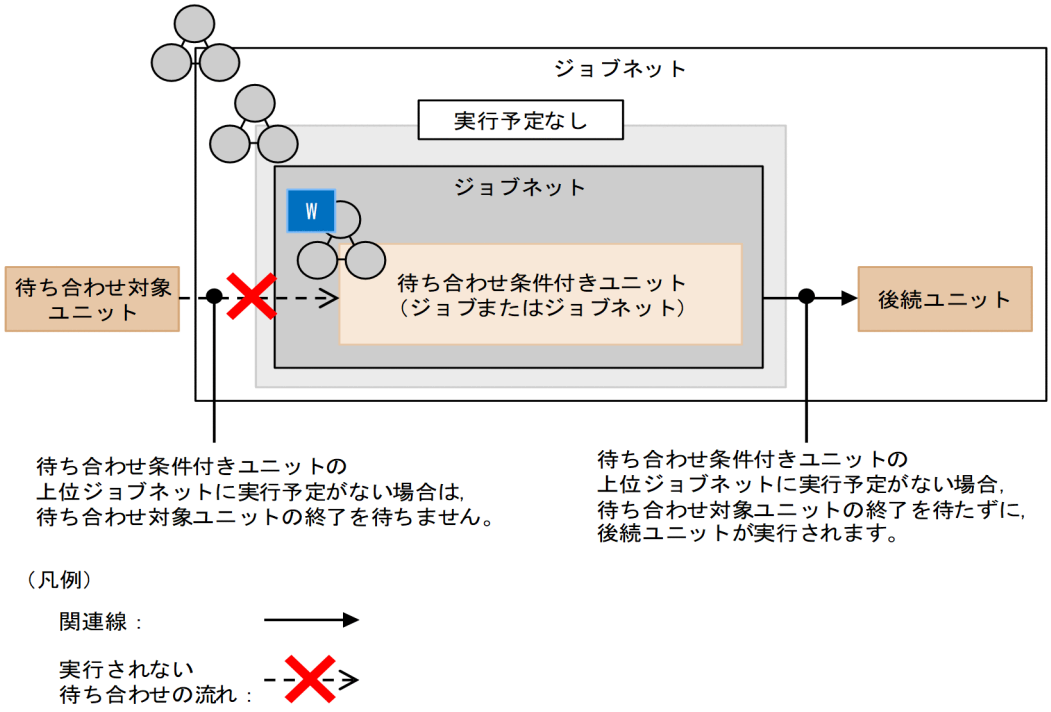
図 2-36　環境設定パラメーター PREWAITNOSCHUNITS に「yes」を指定した場合の動作



待ち合わせ条件付きユニットの上位ジョブネットに実行予定がない場合は，環境設定パラメーター PREWAITNOSCHUNITS に指定した値にかかわらず，待ち合わせは行われません。実行予定がない上位ジョブネットは，先行のジョブやジョブネットの実行終了など，上位ジョブネット自身の実行条件を満たすと「計画未実行」状態となり，その配下のユニットは，待ち合わせ条件付きユニットを含めてすべて「計画未実行」状態となります。

待ち合わせ条件付きユニットの上位ジョブネットに実行予定がない場合の動作を，次の図に示します。

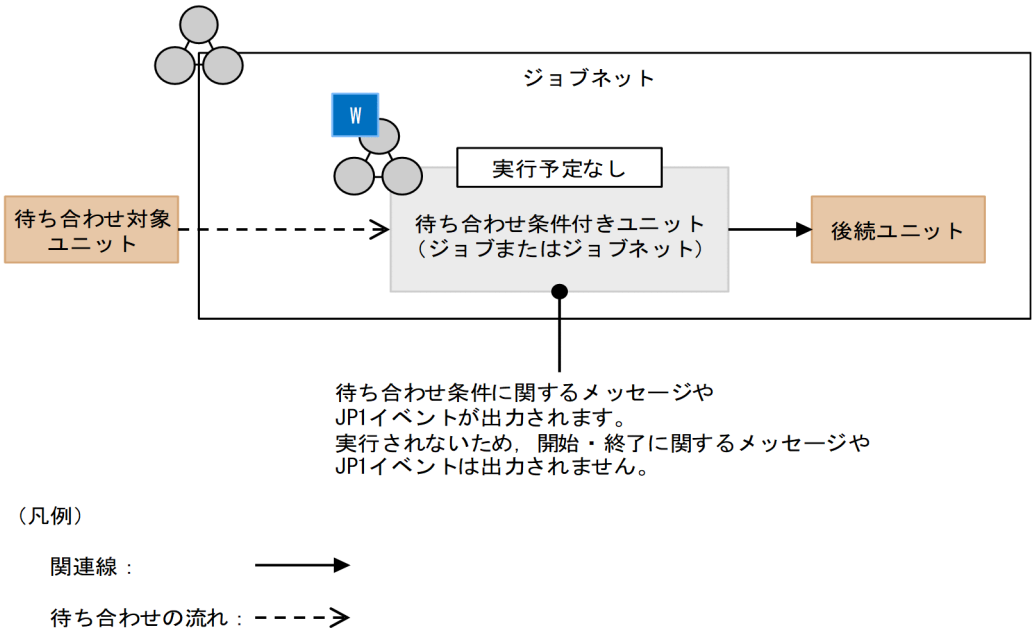
図 2-37　待ち合わせ条件付きユニットの上位ジョブネットに実行予定がない場合の動作



また，環境設定パラメーターPREWAITNOSCHUNITS に「yes」を指定した場合，待ち合わせ条件付きユニットに実行予定がないときでも待ち合わせ条件は有効になり，待ち合わせ状態に対応したメッセージや JP1 イベントが出力されます。待ち合わせ条件が成立すると，実行予定のない待ち合わせ条件付きユニットは実行されないため，開始メッセージ，終了メッセージおよび JP1 イベントは出力されません。ただし，待ち合わせでエラーが発生した場合は，待ち合わせ条件付きユニットが「異常検出終了」状態に遷移し，メッセージや JP1 イベントが出力されます。

環境設定パラメーターPREWAITNOSCHUNITS に「yes」を指定した場合のメッセージや JP1 イベントの動作を，次の図に示します。

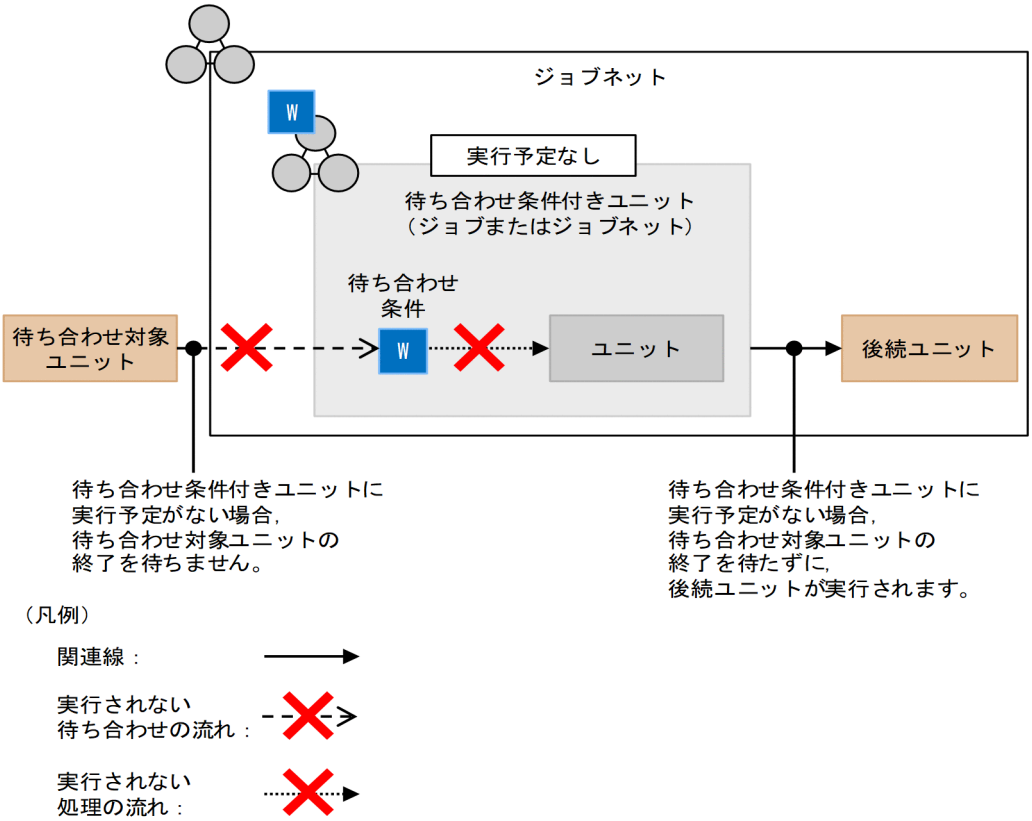
図 2-38　環境設定パラメーター PREWAITNOSCHUNITS に「yes」を指定した場合のメッセージや JP1 イベントの動作



補足事項

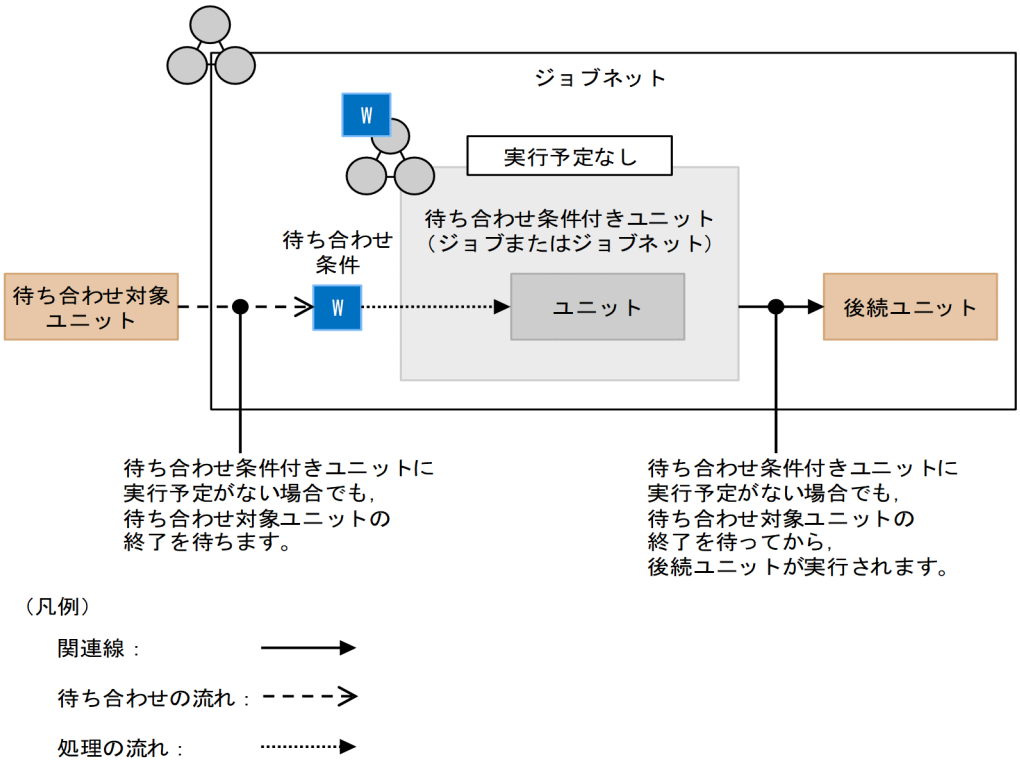
待ち合わせ条件付きユニットに実行予定がない場合，デフォルトの設定では，待ち合わせ条件付きユニットの待ち合わせ条件の処理も，ユニットの処理と合わせて実行しないで終了します。

図 2-39　環境設定パラメーター PREWAITNOSCHUNITS がデフォルトの場合の処理の流れ



環境設定パラメーターPREWAITNOSCHUNITS に「yes」を指定すると，待ち合わせ条件付きユニットの処理を実行しない範囲から待ち合わせ条件を出すことで，待ち合わせをします。なお，環境設定パラメーターPREWAITNOSCHUNITS に「yes」を指定していても，待ち合わせ条件付きユニットの上位ジョブネットに実行予定がない場合，上位ジョブネットの処理をしない範囲に待ち合わせ条件付きユニットの待ち合わせ条件も含まれるため，待ち合わせは行われません。

図 2-40　環境設定パラメーター PREWAITNOSCHUNITS に「yes」を指定した場合の処理の流れ



注意事項

環境設定パラメーターPREWAITNOSCHUNITS に「yes」を指定していても，次の場合は，待ち合わせは行われません。後続のユニットは，待ち合わせ対象ユニットの終了を待たずに実行されます。

- 実行予定のない待ち合わせ条件付きユニットが従属ユニットで，その判定ジョブが「正常終了-偽」状態になった場合
- 実行予定のない待ち合わせ条件付きユニットが OR ジョブの先行のイベントジョブで，別の先行のイベントジョブが終了した場合

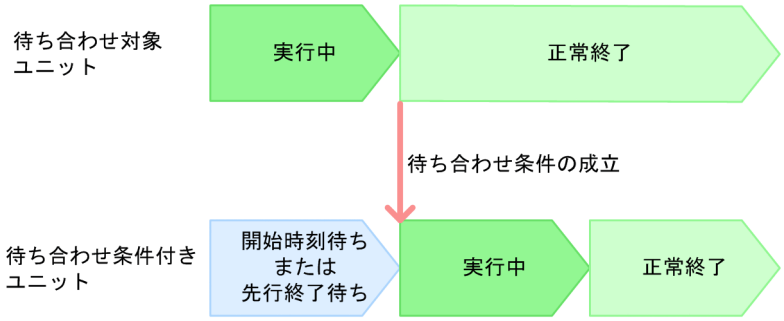
(4)　待ち合わせ条件付きユニットと待ち合わせ対象ユニットの状態遷移

(a)　待ち合わせ条件付きユニットまたは待ち合わせ対象ユニットが起動条件を使用しない場合の状態遷移

■ 待ち合わせ条件付きユニットに実行予定がある場合の状態遷移

待ち合わせが成立するときのユニットの状態遷移を，次の図に示します。

図 2-41　待ち合わせ条件付きユニットに実行予定があるときの待ち合わせ条件成立時のユニットの状態遷移（起動条件を使用しない場合）



待ち合わせ条件付きユニットは，待ち合わせ対象ユニットが「実行中」状態の間，「開始時刻待ち」または「先行終了待ち」状態で待ち合わせ条件の成立を待ちます。どちらの状態になるかは，待ち合わせ条件付きユニットのユニット種別によって異なります。

待ち合わせ条件付きユニットのユニット種別と待ち合わせ条件成立前の状態を，次の表に示します。待ち合わせ条件を設定できるユニット種別については，「(1)(a) 待ち合わせ条件付きユニットと待ち合わせ対象ユニット」を参照してください。

表 2-11　待ち合わせ条件付きユニットのユニット種別と待ち合わせ条件成立前の状態（待ち合わせ条件付きユニットに実行予定がある場合）

項番	待ち合わせ条件付きユニットのユニット種別	待ち合わせ条件成立前の状態
1	ルートジョブネット	開始時刻待ち
2	ネストジョブネット	先行終了待ち
3	• 標準ジョブ • ジョブネットコネクタ • イベントジョブ • アクションジョブ • カスタムジョブ • 引き継ぎ情報設定ジョブ • HTTP 接続ジョブ	先行終了待ち

待ち合わせ条件付きユニットが「開始時刻待ち」または「先行終了待ち」状態のときに，待ち合わせ対象ユニットが次の状態に遷移すると，待ち合わせ条件が成立します。

- 正常終了
- 警告検出終了
- 計画未実行

待ち合わせ条件が成立すると，待ち合わせ条件付きユニットが「実行中」状態に遷移し，実行を開始します。

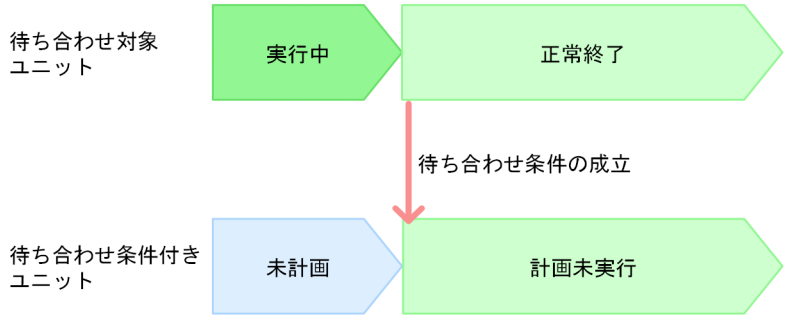
なお，待ち合わせ対象ユニットが異常終了すると，待ち合わせ条件付きユニットは「開始時刻待ち」または「先行終了待ち」状態のまま，待ち合わせ条件の成立を待ち続けます。この場合，待ち合わせ対象ユニットの異常の原因を取り除いたあと，待ち合わせ対象ユニットを再実行してください。

■ 待ち合わせ条件付きユニットに実行予定がない場合の状態遷移

環境設定パラメーターPREWAITNOSCHUNITS で「yes」を指定していると，待ち合わせ条件付きユニットに実行予定がない場合でも待ち合わせをします。環境設定パラメーターPREWAITNOSCHUNITS については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2(122) PREWAITNOSCHUNITS」を参照してください。

待ち合わせ条件が成立するときのユニットの状態遷移を，次の図に示します。

図 2-42　待ち合わせ条件付きユニットに実行予定がないときの待ち合わせ条件成立時のユニットの状態遷移（起動条件を使用しない場合）



環境設定パラメーターPREWAITNOSCHUNITS で「yes」を指定していると，実行予定がない待ち合わせ条件付きユニットは，待ち合わせ対象ユニットが「実行中」状態の間，「未計画」状態で待ち合わせ条件の成立を待ちます。

待ち合わせ条件付きユニットのユニット種別と待ち合わせ条件成立前の状態を，次の表に示します。待ち合わせ条件を設定できるユニット種別については，「(1)(a) 待ち合わせ条件付きユニットと待ち合わせ対象ユニット」を参照してください。

表 2-12　待ち合わせ条件付きユニットのユニット種別と待ち合わせ条件成立前の状態（実行予定がない場合）

項番	待ち合わせ条件付きユニットのユニット種別	待ち合わせ条件成立前の状態
1	ネストジョブネット	未計画
2	• 標準ジョブ • イベントジョブ • アクションジョブ • カスタムジョブ • 引き継ぎ情報設定ジョブ • HTTP 接続ジョブ	未計画

待ち合わせ条件付きユニットが「未計画」状態のときに，待ち合わせ対象ユニットが次の状態に遷移すると，待ち合わせ条件が成立します。

- 正常終了
- 警告検出終了
- 計画未実行

待ち合わせ条件が成立すると，待ち合わせ条件付きユニットが「計画未実行」状態に遷移し，後続ジョブの実行が開始されます。

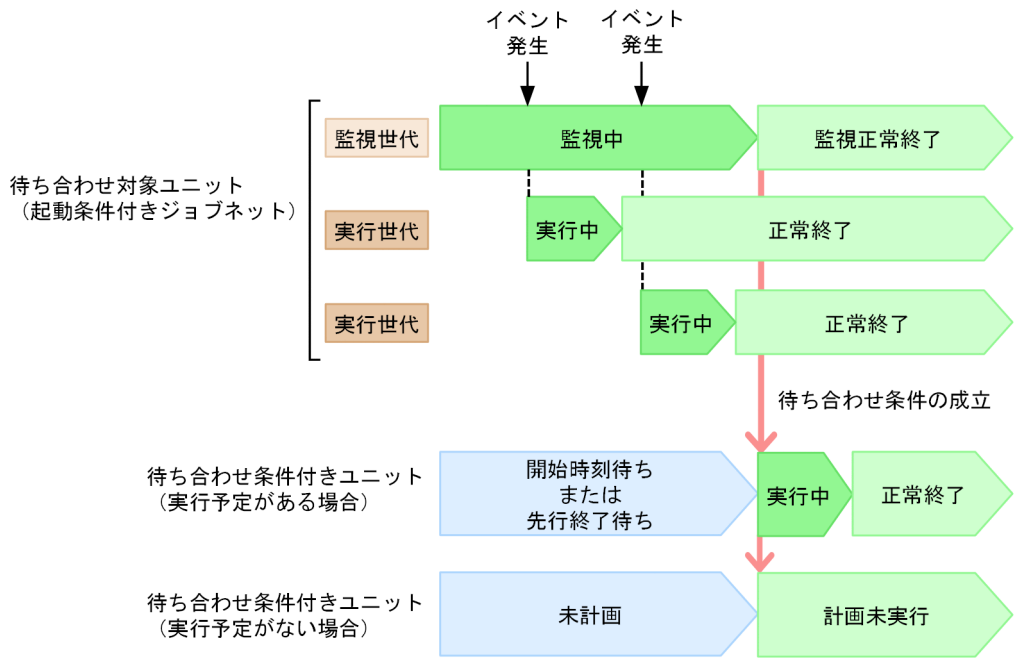
なお，待ち合わせ対象ユニットが異常終了すると，待ち合わせ条件付きユニットは「未計画」状態のまま，待ち合わせ条件の成立を待ち続けます。この場合，待ち合わせ対象ユニットの異常の原因を取り除いたあと，待ち合わせ対象ユニットを再実行してください。

(b) 待ち合わせ条件付きユニットまたは待ち合わせ対象ユニットが起動条件を使用する場合の状態遷移

■ 待ち合わせ対象ユニットが起動条件を使用する場合の状態遷移

待ち合わせ条件が成立するときのユニットの状態遷移を，次の図に示します。

図 2-43 待ち合わせ条件成立時のユニットの状態遷移（待ち合わせ対象ユニットが起動条件を使用する場合）



待ち合わせ条件付きユニットに実行予定がある場合，待ち合わせ対象ユニット（起動条件付きジョブネット）の監視世代が「監視中」状態，または実行世代が「実行中」状態の間，待ち合わせ条件付きユニットは「開始時刻待ち」または「先行終了待ち」状態で待ち合わせ条件の成立を待ちます。どちらの状態にな

るかは，待ち合わせ条件付きユニットのユニット種別によって異なります。待ち合わせ条件成立前の待ち合わせ条件付きユニットの状態については，表 2-11 を参照してください。

起動条件付きジョブネットが実行終了して待ち合わせ条件が成立すると，実行予定がある待ち合わせ条件付きユニットは「実行中」状態に遷移して実行を開始します。

待ち合わせ条件付きユニットに実行予定がない場合は，環境設定パラメーターPREWAITNOSCHUNITS で「yes」を指定していると，待ち合わせ対象ユニット（起動条件付きジョブネット）の監視世代が「監視中」状態，または実行世代が「実行中」状態の間，待ち合わせ条件付きユニットは「未計画」状態で待ち合わせ条件の成立を待ちます。環境設定パラメーターPREWAITNOSCHUNITS については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2(122) PREWAITNOSCHUNITS」を参照してください。また，待ち合わせ条件成立前の待ち合わせ条件付きユニットの状態については，表 2-12 を参照してください。

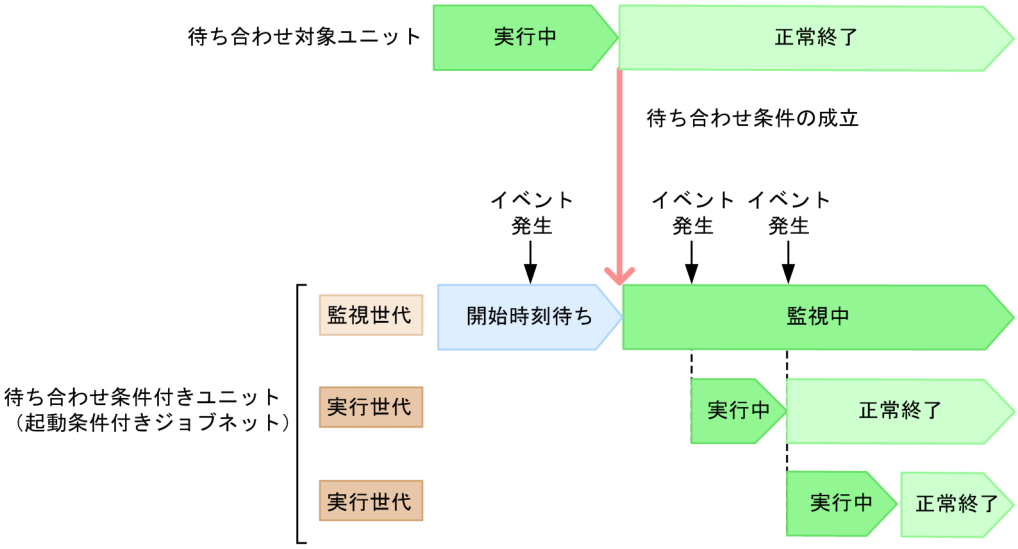
起動条件付きジョブネットが実行終了して待ち合わせ条件が成立すると，実行予定がない待ち合わせ条件付きユニットは「計画未実行」状態に遷移し，続けて後続ユニットが実行されます。

なお，起動条件付きジョブネットの実行終了とする監視世代および実行世代の状態は，待ち合わせ条件の設定で変更できます。詳細については，「(5)(c) 待ち合わせ対象ユニットに起動条件を使用する場合の，待ち合わせ条件の設定」を参照してください。

■ 待ち合わせ条件付きユニットが起動条件を使用する場合の状態遷移

待ち合わせ条件が成立するときのユニットの状態遷移を，次の図に示します。

図 2-44　待ち合わせ条件成立時のユニットの状態遷移（待ち合わせ条件付きユニットが起動条件を使用する場合）



待ち合わせ対象ユニットが「実行中」状態の間，待ち合わせ条件付きユニット（起動条件付きジョブネット）の監視世代は，「開始時刻待ち」状態で待ち合わせ条件の成立を待ちます。このとき，起動条件で監視するイベントが発生しても，起動条件の監視が開始されていないため，起動条件は成立しません。そのため，実行世代は実行を開始しません。

注意事項

起動条件付きジョブネットが待ち合わせ対象ユニットの実行終了を待ち合わせている間，イベントは監視されません。そのため，待ち合わせ対象ユニットが遅延すると，その分イベントの監視開始も遅延します。起動条件付きジョブネットへの待ち合わせ条件の設定は，イベントの監視開始が遅延しても問題ない使い方に限定してください。

イベントの監視開始が遅延しても問題ない使い方の例

- ファイル監視ジョブを使って監視開始前にすでにファイルが存在しても起動条件を成立させる

  この使い方は，監視開始と同時に起動条件付きジョブネットが実行を開始するため，待ち合わせ対象ユニットが遅延して監視開始が遅延しても問題ありません。

イベントの監視開始が遅延すると問題のある使い方の例

- 実行間隔制御ジョブを使って特定の時刻から一定間隔で繰り返しジョブネットを起動させる

  この使い方は，繰り返しの開始時刻が待ち合わせ対象ユニットの実行終了のタイミングに依存するため，繰り返しの開始および終了が，期待した時刻より遅延するおそれがあります。

(c) 待ち合わせ条件が成立しても待ち合わせ条件付きユニットが実行されない場合の状態遷移

待ち合わせ条件付きユニットや待ち合わせ対象ユニットを次のように定義していたり操作したりする場合，待ち合わせ条件が成立しても待ち合わせ条件付きユニットが実行されないことがあります。

- 待ち合わせ条件付きユニットに関連線で接続された先行ユニットがある場合
- 待ち合わせ条件付きユニットに保留属性を設定している場合
- 待ち合わせ条件付きユニットを同時に複数回実行する場合
- 待ち合わせ条件付きユニットまたは待ち合わせ対象ユニットを再実行する場合

このような場合での状態遷移の例について説明します。

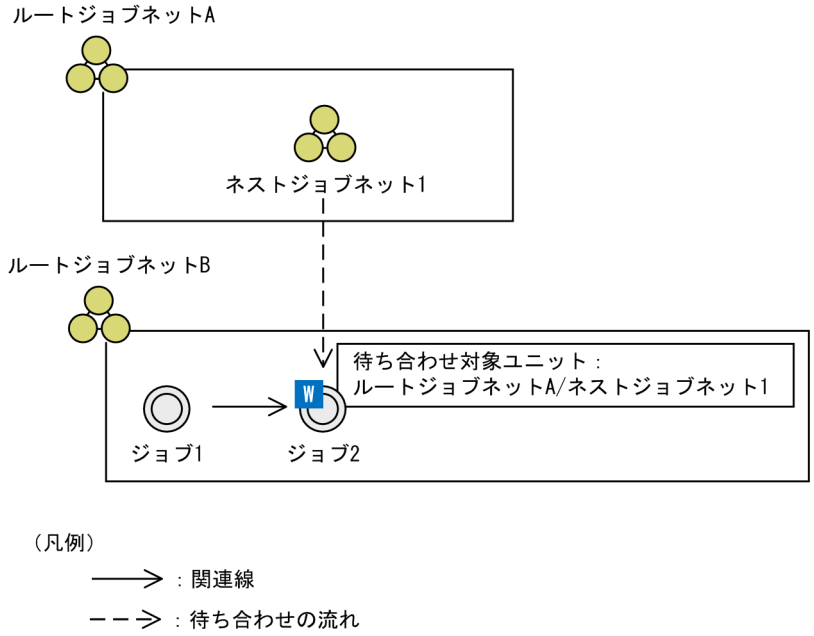
■ 待ち合わせ条件付きユニットに関連線で接続された先行ユニットがある場合

待ち合わせ条件付きユニットに関連線で接続された先行ユニットがある場合，待ち合わせ条件付きユニットは，先行ユニットが実行終了してから待ち合わせ条件が成立しているかどうかをチェックします。先行ユニットが実行終了しないうちに待ち合わせ条件が成立しても，待ち合わせ条件の成立チェックはされないため，待ち合わせ条件付きユニットは実行を開始しません。

関連線で接続された先行ユニットがある場合の例を，次の図に示します。

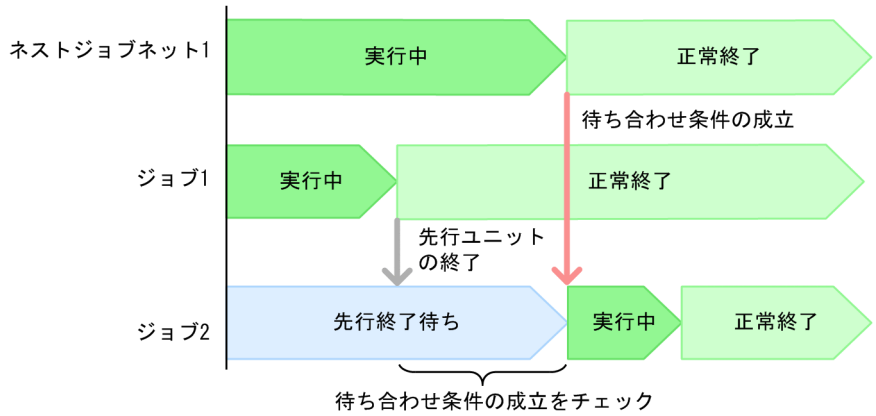
図 2-45　関連線で接続された先行ユニットがある場合の例

■ユニット構成

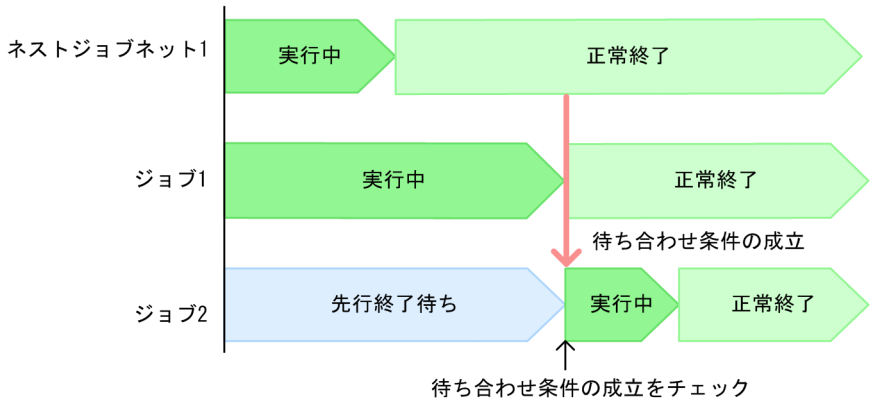


■状態遷移

＜例1：先行ユニットが待ち合わせ対象ユニットよりも先に実行終了する場合＞



＜例2：先行ユニットが待ち合わせ対象ユニットよりもあとに実行終了する場合＞



この例では，待ち合わせ条件付きユニットのジョブ 2 に，待ち合わせ対象ユニットとしてネストジョブネット 1 が定義されています。また，先行ユニットとしてジョブ 1 が定義されています。

例 1 では，ジョブ 1 が実行終了してからネストジョブネット 1 が実行終了しています。この場合，ジョブ 1 が実行終了した時点でジョブ 2 は待ち合わせ条件の成立チェックを開始します。そのため，ネストジョブネット 1 が実行終了して待ち合わせ条件が成立すると，ジョブ 2 は実行を開始します。

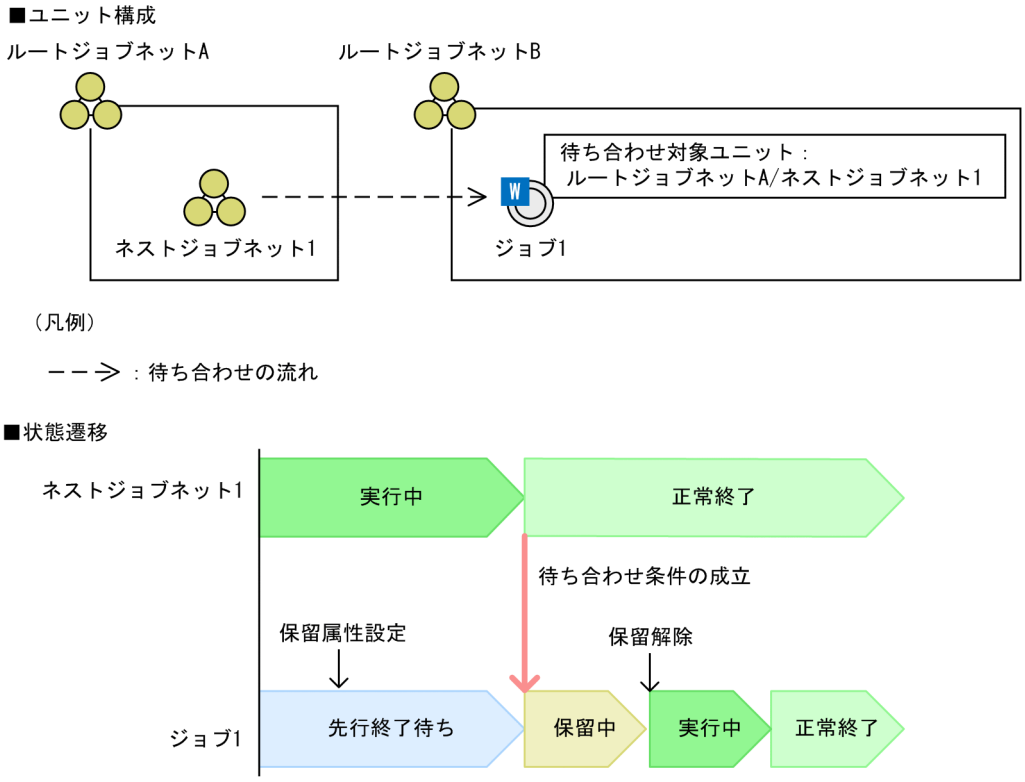
例 2 では，ネストジョブネット 1 が実行終了してからジョブ 1 が実行終了しています。この場合，ネストジョブネット 1 が正常終了しても，ジョブ 2 は待ち合わせ条件の成立チェックを開始していないため，ジョブ 2 は実行されません。ジョブ 1 が正常終了すると，ジョブ 2 は待ち合わせ条件の成立チェックを開始します。待ち合わせ条件の成立チェック開始時点でネストジョブネット 1 が正常終了していれば，待ち合わせ条件は成立し，ジョブ 2 はすぐに実行を開始します。

■ 待ち合わせ条件付きユニットに保留属性を設定している場合

待ち合わせ条件付きユニットに保留属性を設定している場合，待ち合わせ条件が成立しても，待ち合わせ条件付きユニットの実行は開始されないで保留されます。

保留属性を設定する場合の例を，次の図に示します。

図 2-46　保留属性を設定する場合の例



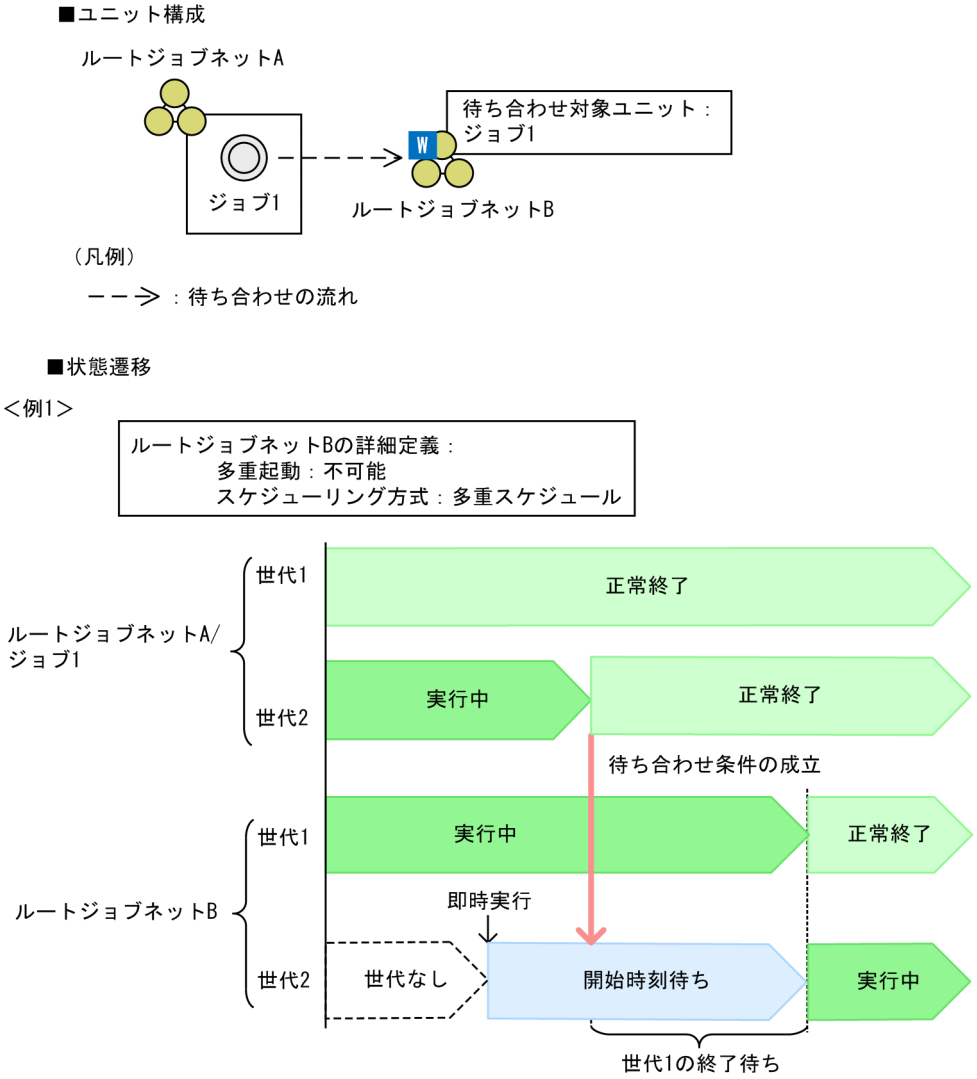
この例では，待ち合わせ条件付きユニットのジョブ 1 が「先行終了待ち」状態の間に，ジョブ 1 に保留属性を設定しています。この場合，待ち合わせ条件が成立しても，ジョブ 1 は実行を開始しないで「保留中」状態に遷移します。保留設定が解除されると，ジョブ 1 は実行を開始します。

■ 待ち合わせ条件付きユニットを重複して複数回実行する場合

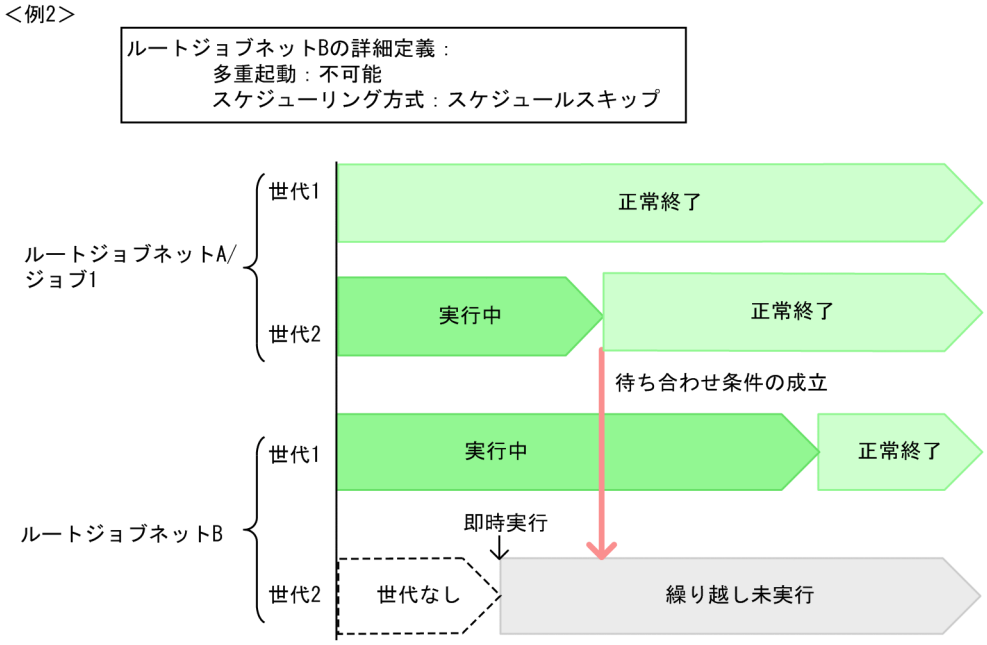
待ち合わせ条件付きユニットを重複して複数回実行する場合，ルートジョブネットの多重起動やスケジューリング方式の設定に従って，待ち合わせ条件が成立しても実行中の世代の終了を待ってから実行されたり，ルートジョブネットの次回実行予定が「繰り越し未実行」となって実行されなかったりすることがあります。

多重起動およびスケジューリング方式の違いによる状態遷移の例を，次の図に示します。

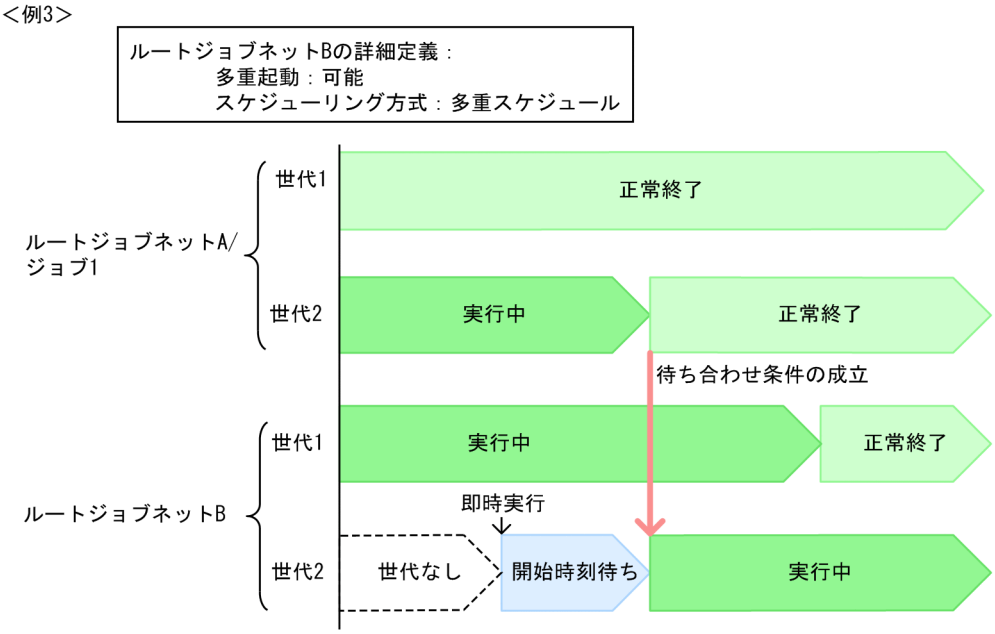
図 2-47　多重起動およびスケジューリング方式の違いの例



＜例2＞



＜例3＞



例 1 では，待ち合わせ条件付きユニットのルートジョブネット B の詳細定義に，多重起動として「不可能」を，スケジューリング方式として「多重スケジュール」を設定しています。ルートジョブネット B の世代 1 が「実行中」状態のときにルートジョブネット B を即時実行すると，世代 2 が生成されて，「開始時刻待ち」状態でルートジョブネット A/ジョブ 1 の世代 2 の実行終了を待ち合わせます。ジョブ 1 の世代 2 が実行終了して待ち合わせ条件が成立すると，多重起動できないため，ルートジョブネット B の世代 2 は「開始時刻待ち」状態のままルートジョブネット B の世代 1 の実行終了を待ちます。そのあと，ルートジョブネット B の世代 1 が実行終了すると，ルートジョブネット B の世代 2 は実行を開始します。

例 2 では，ルートジョブネット B の詳細定義に，多重起動として「不可能」を，スケジューリング方式として「スケジュールスキップ」を設定しています。ルートジョブネット B の世代 1 が「実行中」状態のと

きにルートジョブネット B を即時実行すると，世代 2 が生成されます。しかし，世代 1 が実行中のため，世代 2 はスキップされて，「繰り越し未実行」状態に遷移します。そのあと，ルートジョブネット A のジョブ 1 の世代 2 が正常終了しても，ルートジョブネット B の世代 2 は終了しているので，実行を開始しません。

例 3 では，ルートジョブネット B の詳細定義に，多重起動として「可能」を，スケジューリング方式として「多重スケジュール」を設定しています。ルートジョブネット B の世代 1 が「実行中」状態のときにルートジョブネット B を即時実行すると，世代 2 が生成されて，「開始時刻待ち」状態でルートジョブネット A/ジョブ 1 の世代 2 の実行終了を待ち合わせます。ジョブ 1 の世代 2 が実行終了して待ち合わせ条件が成立すると，ルートジョブネット B の世代 2 は実行を開始します。

注意事項

ルートジョブネットに待ち合わせ条件を設定する場合は，次回実行予定の開始予定時刻になる前に，前回の実行予定の待ち合わせ条件が成立して，実行を開始するように運用してください。多重起動を設定して多重スケジュールを指定していても，前回の実行予定が実行を開始していない場合は，次回実行予定は待ち合わせ条件が成立しても実行を開始しません。

■ 待ち合わせ条件付きユニットまたは待ち合わせ対象ユニットを再実行する場合

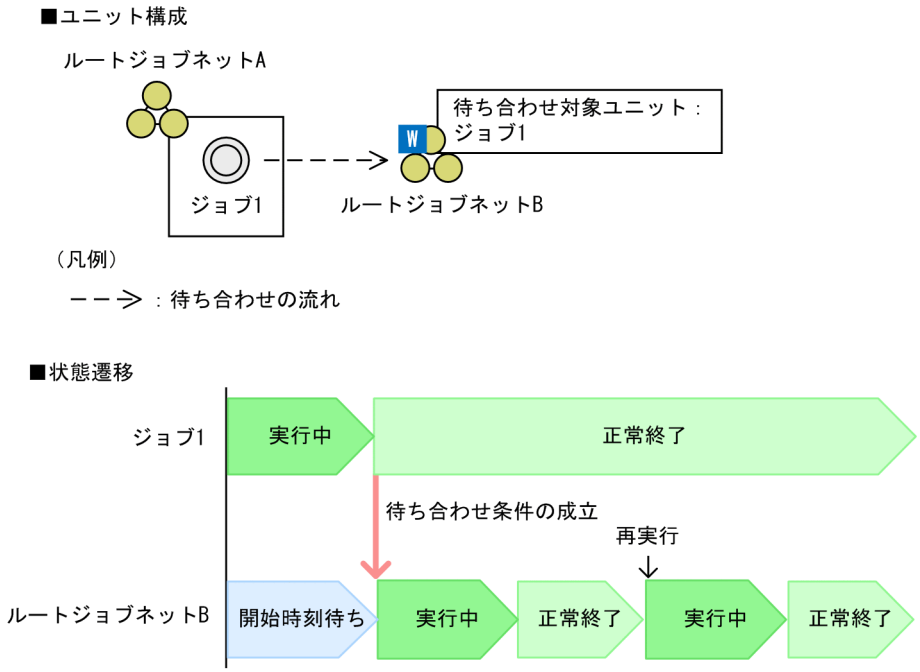
待ち合わせ条件付きユニットまたは待ち合わせ対象ユニットを再実行する場合の，ユニットの状態遷移について説明します。

待ち合わせ条件付きユニットを再実行する場合

一度待ち合わせ条件が成立した待ち合わせ条件付きユニットを再実行しても，待ち合わせ条件は未成立には戻りません。そのため，待ち合わせ条件が成立して実行終了した待ち合わせ条件付きユニットをそのまま再実行すると，待ち合わせ条件付きユニットは待ち合わせを開始しますが，待ち合わせ条件はすでに成立しているため，すぐに実行を開始します。

待ち合わせ条件付きユニットを再実行する場合の例を，次の図に示します。

図 2-48　待ち合わせ条件付きユニットを再実行する場合の例



この例では，待ち合わせ対象ユニットのジョブ 1 が正常終了して待ち合わせ条件が成立したあと，ルートジョブネット B を再実行しています。ルートジョブネット B を再実行しても，待ち合わせ条件はすでに成立しているため，ルートジョブネット B はジョブ 1 の実行終了を待ち合わせないで実行を開始します。

補足事項

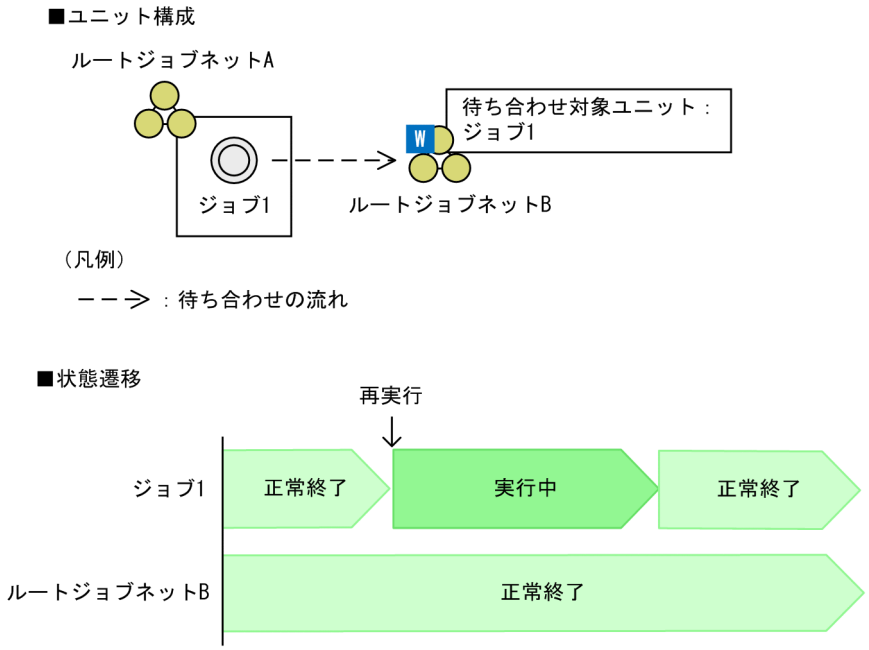
環境設定パラメーターPREWAITNOSCHUNITS で「yes」を指定していて，待ち合わせ条件付きユニットに実行予定がない場合，待ち合わせ条件付きユニットをそのまま再実行すると計画未実行状態となり，後続ユニットが実行されます。

待ち合わせ対象ユニットを再実行する場合

待ち合わせ対象ユニットを再実行すると，待ち合わせ対象ユニットだけが再実行され，待ち合わせ条件付きユニットは再実行されません。

待ち合わせ対象ユニットを再実行する場合の例を，次の図に示します。

図 2-49　待ち合わせ対象ユニットを再実行する場合の例



この例では，待ち合わせ対象ユニットのジョブ 1 および待ち合わせ条件付きユニットのルートジョブネット B が正常終了している場合に，ジョブ 1 を再実行しています。ジョブ 1 は実行を開始しますが，ルートジョブネット B は正常終了したまま，再実行しません。

注意事項

待ち合わせ対象ユニットに起動条件を使用する場合，監視世代は再実行しないでください。監視世代を再実行しても再監視はしないため，意図せず待ち合わせ条件が成立してしまうことがあります。監視世代が「監視打ち切り終了」状態や「監視未起動終了」状態など待ち合わせ条件が成立しない状態で終了した場合は，待ち合わせ条件を無効化することで，待ち合わせ条件付きユニットを実行開始させることができます。待ち合わせ条件の設定の一時的な変更については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.5.15 ジョブネットやジョブの待ち合わせ条件の設定を一時変更する」を参照してください。

(d)　待ち合わせ対象ユニットの後続ユニットを再実行した場合の待ち合わせ条件

待ち合わせ対象ユニットの後続ユニットを再実行すると，待ち合わせ対象ユニットの状態が変化することがあります。この場合に，待ち合わせ条件を成立させるかどうか，環境設定パラメーターPREWAITRERUNSTATUSで設定できます。JP1/AJS3 を新規インストールした状態では，待ち合わせ条件は成立しないように設定されます。

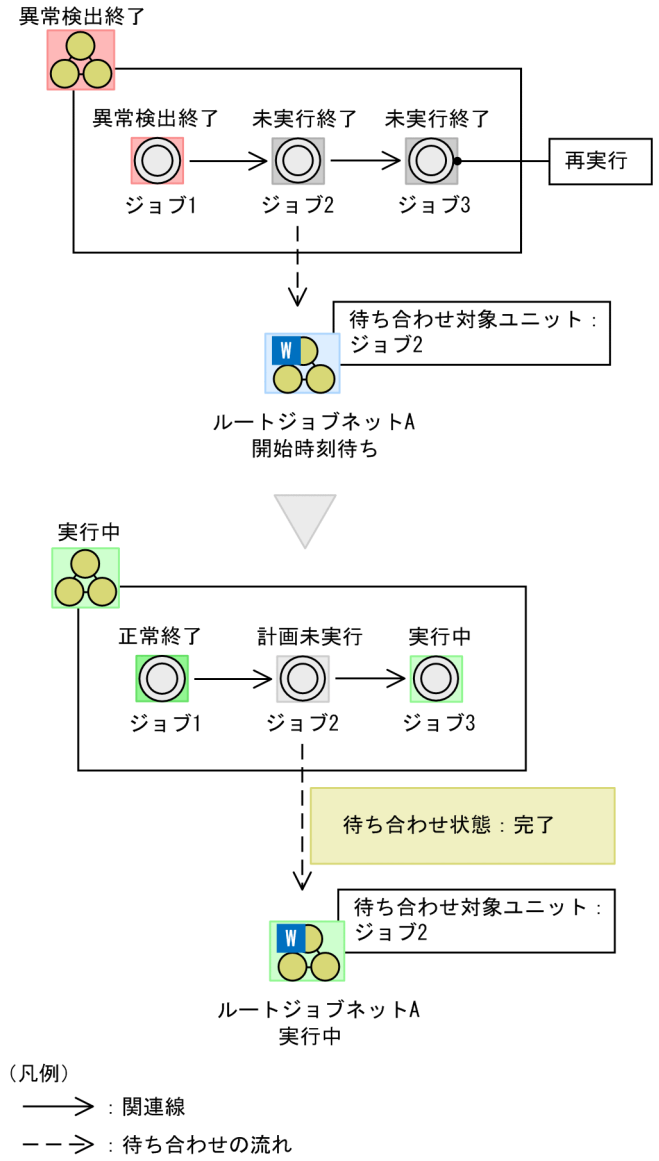
環境設定パラメーターPREWAITRERUNSTATUS の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2(110) PREWAITRERUNSTATUS」を参照してください。

待ち合わせ条件が成立するように設定する

環境設定パラメーターPREWAITRERUNSTATUS で「no」を指定していると，後続ユニットの再実行によって待ち合わせ対象ユニットの状態が変化すると，待ち合わせ条件が成立し，待ち合わせ条件付きユニットが実行を開始します。

環境設定パラメーターPREWAITRERUNSTATUS で「no」を指定している場合に，待ち合わせ対象ユニットの後続ユニットを再実行する例を次の図に示します。

図 2-50　待ち合わせ対象ユニットの後続ユニットを再実行する例（待ち合わせ条件が成立する設定）



この例では，待ち合わせ対象ユニットであるジョブ 2，およびその後続ユニットのジョブ 3 が「未実行終了」状態の場合に，ジョブ 3 を再実行しています。ジョブ 3 を再実行すると，ジョブ 2 は「未実行終了」状態から「計画未実行」状態に遷移します。

環境設定パラメーターPREWAITRERUNSTATUS で「no」を指定していると，待ち合わせ状態は「完了」となり，待ち合わせ条件が成立します。そして，待ち合わせ条件付きユニットのルートジョブネット A が実行を開始します。

環境設定パラメーターPREWAITRERUNSTATUS で「no」を指定しているときは，再実行による待ち合わせ対象ユニットの状態の変化に注意してください。待ち合わせ対象ユニットの後続ユニットを再実行する場合は，状態が変化するユニットの中に待ち合わせ対象ユニットがないか検索するなどして，影響範囲を明確にしてから再実行してください。再実行の詳細については，マニュアル「JP1/Automatic Job

Management System 3 導入ガイド　4.5.11 実行が終了したジョブネットやジョブを再実行する」を参照してください。待ち合わせ条件の検索方法については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　10.2 ユニットを検索して表示する」を参照してください。

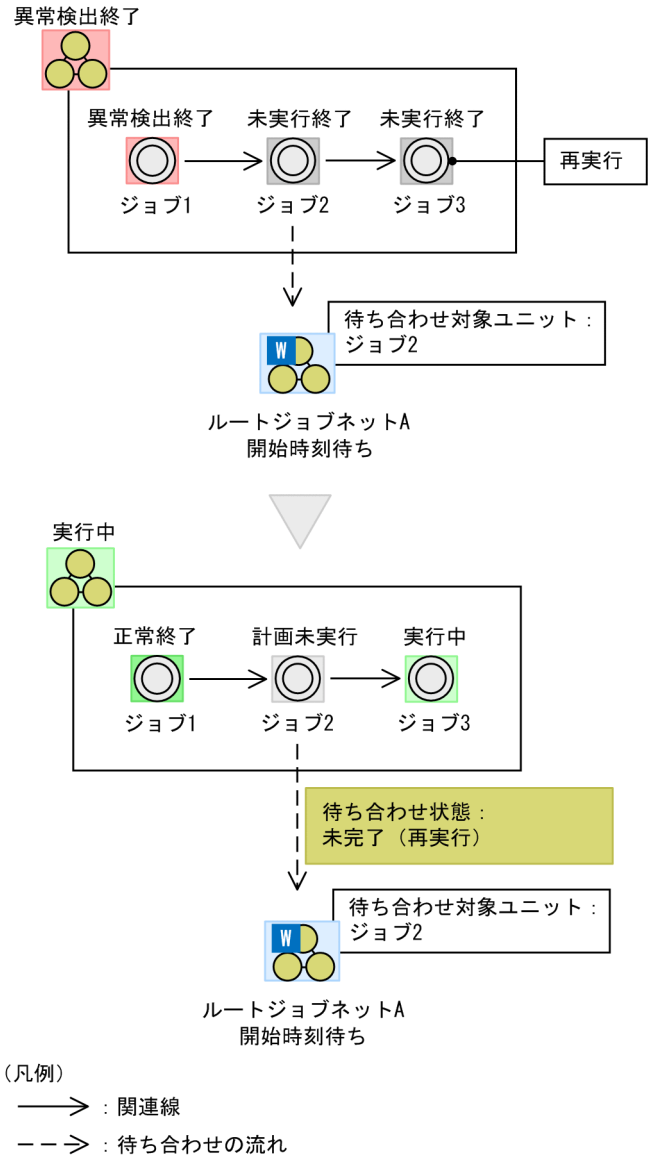
待ち合わせ条件が成立しないように設定する

環境設定パラメーターPREWAITRERUNSTATUS で「yes」を指定していると，後続ユニットの再実行によって待ち合わせ対象ユニットの状態が変化しても，待ち合わせ条件は成立しません。したがって，待ち合わせ条件付きユニットの状態は変わりません。

この設定だと，ユニットの再実行によって先行ユニットの状態が変更することで，意図しない待ち合わせ条件の成立を防ぐことができます。

環境設定パラメーターPREWAITRERUNSTATUS で「yes」を指定している場合に，待ち合わせ対象ユニットの後続ユニットを再実行する例を次の図に示します。

図 2-51　待ち合わせ対象ユニットの後続ユニットを再実行する例（待ち合わせ条件が成立しない設定）



ユニットの定義は，待ち合わせ条件が成立する設定の例と同じです。ジョブ 3 を再実行すると，ジョブ 2 は「未実行終了」状態から「計画未実行」状態に遷移します。

しかし，環境設定パラメーターPREWAITRERUNSTATUS で「yes」を指定していると，待ち合わせ状態は「未完了（再実行）」となり，待ち合わせ条件は成立しません。そのため，待ち合わせ条件付きユニットのルートジョブネット A の状態は変わりません。

なお，待ち合わせ対象ユニットであるジョブ 2 を再実行してジョブ 2 が実行終了すると，待ち合わせ条件が成立し，ルートジョブネット A が実行を開始します。

待ち合わせ対象ユニットがジョブネットで，配下のユニットを個別に再実行する場合，ジョブネットが終了状態になると，待ち合わせが完了します。必要に応じて保留設定やジョブ状態を変更して，待ち合わせが成立しない状態にしてください。

(5) 待ち合わせ条件付きユニットの動作の設定

待ち合わせ条件には，待ち合わせ対象ユニット名以外に，次の項目を設定できます。

- 待ち合わせ対象ユニットが複数ある場合の待ち合わせ方法
- 待ち合わせ対象ユニットに待ち合わせる世代がない場合の，待ち合わせ条件付きユニットの動作
- 待ち合わせ対象ユニットに起動条件を使用する場合の，待ち合わせ条件付きユニットの動作

待ち合わせ条件付きユニットの待ち合わせ方法，および待ち合わせる世代がない場合の待ち合わせ条件付きユニットの動作について，次に説明します。

(a) 待ち合わせ対象ユニットが複数ある場合の待ち合わせ方法

一つの待ち合わせ条件付きユニットに対して，待ち合わせ対象ユニットを最大で 32 個指定できます。待ち合わせ対象ユニットを複数指定した場合，待ち合わせ方法として，すべての待ち合わせ対象ユニットの実行終了を待ち合わせるのか，どれか一つの待ち合わせ対象ユニットの実行終了を待ち合わせるのかを設定できます。

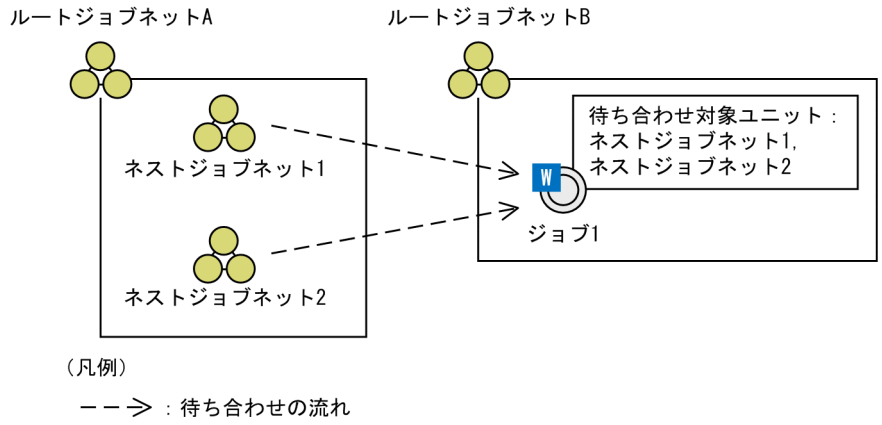
待ち合わせ方法は次の二つから選択します。

- AND
  指定した待ち合わせ対象ユニットがすべて実行終了すれば，待ち合わせ条件が成立します。
- OR
  指定した待ち合わせ対象ユニットのうちどれか一つが実行終了すれば，待ち合わせ条件が成立します。

AND と OR の動作の違いを，次の図に示します。

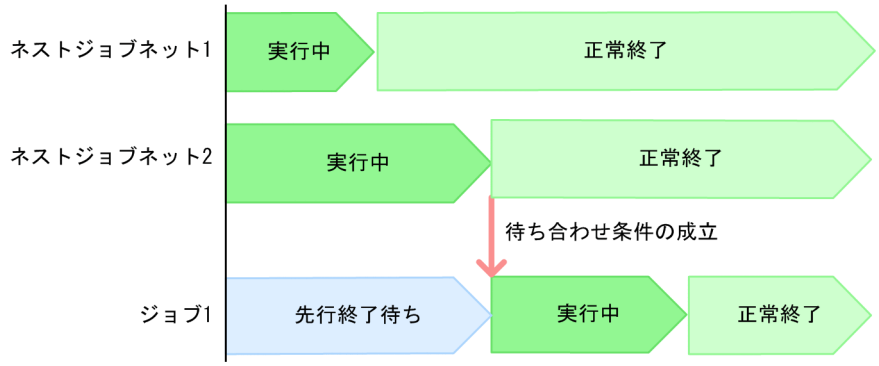
図 2-52　AND と OR の動作の違い

■ユニット構成

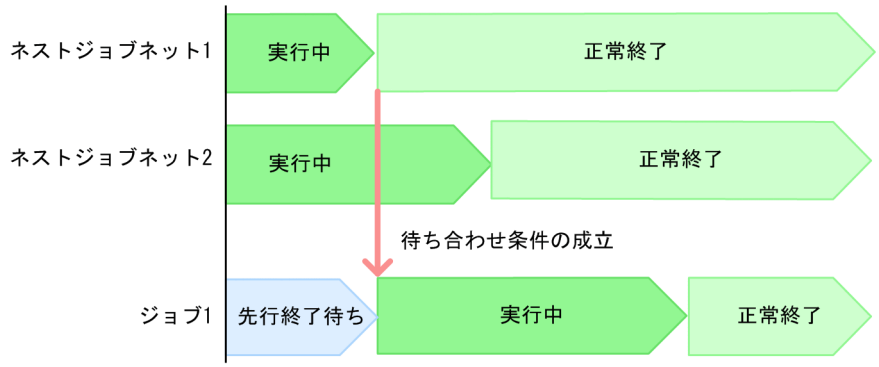


■状態遷移

＜例1：待ち合わせ方法が「AND」の場合＞



＜例2：待ち合わせ方法が「OR」の場合＞



この例では，待ち合わせ条件付きユニットのジョブ 1 に，待ち合わせ対象ユニットとしてネストジョブネット 1 およびネストジョブネット 2 を指定しています。

例 1 の待ち合わせ方法が「AND」の場合，ネストジョブネット 1 とネストジョブネット 2 の両方が実行終了すると，待ち合わせ条件が成立し，ジョブ 1 が実行を開始します。

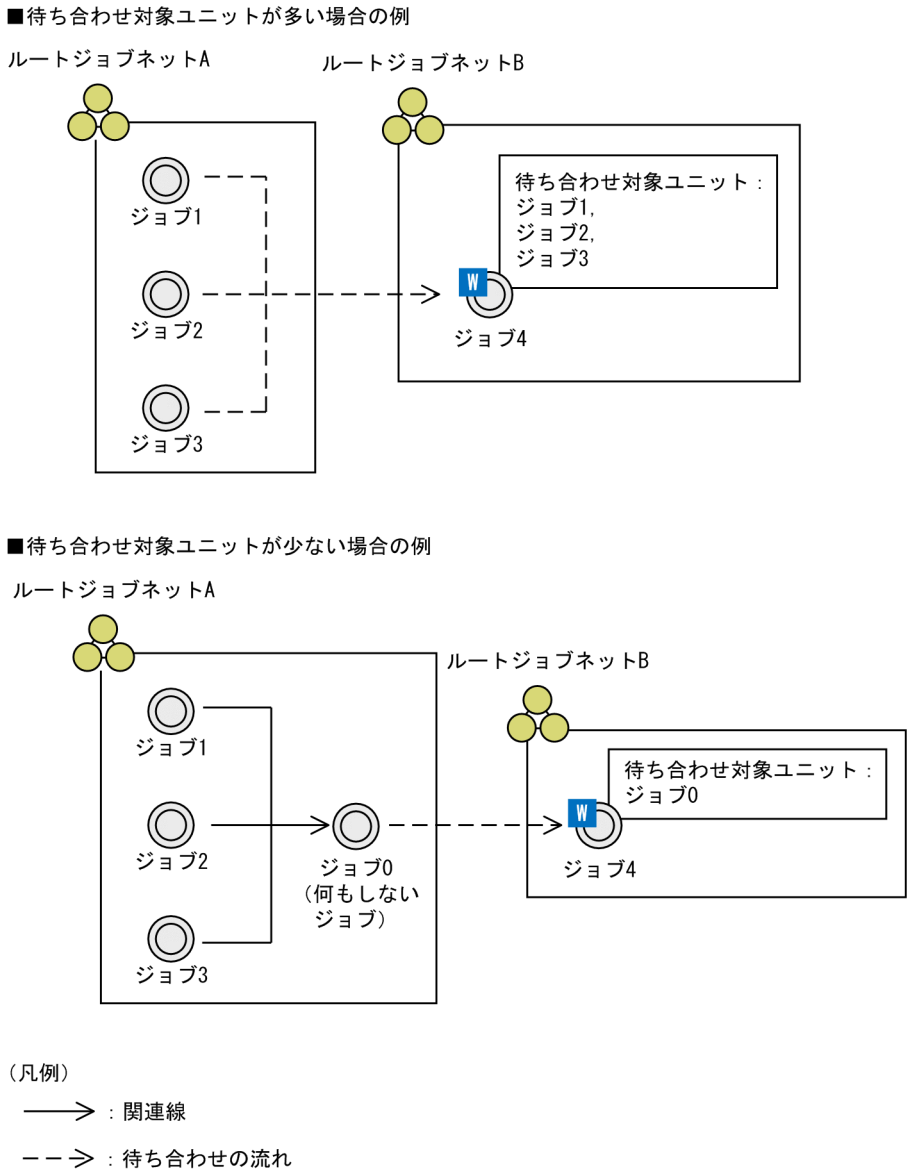
例 2 の待ち合わせ方法が「OR」の場合，ネストジョブネット 1 が実行終了した時点で待ち合わせ条件が成立し，ジョブ 1 が実行を開始します。

注意事項

一つの待ち合わせ条件付きユニットに対する待ち合わせ対象ユニットの数が多いと，JP1/AJS3 の処理速度に影響するおそれがあります。一つの待ち合わせ条件付きユニットに対する待ち合わせ対象ユニット数は，できるだけ少なくなるように設計してください。

待ち合わせ対象ユニットの数を少なくする例を，次の図に示します。

図 2-53　待ち合わせ対象ユニットの数を少なくする例



待ち合わせ対象ユニットが多い場合の例では，ジョブ 1，ジョブ 2，およびジョブ 3 を，待ち合わせ条件付きユニットのジョブ 4 の待ち合わせ対象ユニットとして指定しています。この例で待ち合わせ対象ユニットを減らすためには，待ち合わせ対象ユニットが少ない場合の例のように，ジョブ 1，ジョブ 2，およびジョブ 3 の後続ユニットとして何もしないジョブのジョブ 0 を定義して，それぞれを関連線で接続します。ジョブ 0 をジョブ 4 の待ち合わせ対象ユニットとして指定することで，待ち合わせ対象ユニットを減らすことができます。ただし，この図のように何もしないジョブを定義すると，ジョブ管理がしづらくなります。本来存在しないユニットを定義して待ち合わせ条件を指定するのは，性能が確保できない場合だけにしてください。

補足事項

待ち合わせ方法が「AND」の場合に，待ち合わせ対象ユニットの一部が実行終了している状態でスケジューラーサービスをホットスタートで再起動すると，再起動後でも，再起動前の待ち合わせ条件の成立状態が維持されます。そのため，再起動後に実行終了していない待ち合わせ対象ユニットが実行終了すると，待ち合わせ条件は成立します。

例えば図 2-52 の例 1 の場合で，ネストジョブネット 1 が正常終了していてネストジョブネット 2 が実行中のときにスケジューラーサービスをホットスタートで再起動したとします。この場合，再起動後にネストジョブネット 2 が実行終了すれば，待ち合わせ条件が成立します。

(b) 待ち合わせ対象ユニットに待ち合わせる世代がない場合の，待ち合わせ条件付きユニットの動作

待ち合わせ条件付きユニットは，「(2) 待ち合わせ条件付きユニットと待ち合わせ対象ユニットの待ち合わせルール」の待ち合わせルールに従って，待ち合わせ対象ユニットの世代の実行終了を待ち合わせます。しかし，待ち合わせ対象ユニットに待ち合わせルールを満たす世代がない場合，待ち合わせ条件付きユニットは，次のどちらかのように動作します。

- 実行を開始しない

  待ち合わせ対象ユニットに待ち合わせる世代がなくても，「開始時刻待ち」，「先行終了待ち」，または「未計画※」状態のまま待ち合わせ条件の成立を待ち続けます。

  注※

  環境設定パラメーターPREWAITNOSCHUNITS で「yes」を指定している場合は，待ち合わせ条件付きユニットが「未計画」状態でも待ち合わせをします。環境設定パラメーターPREWAITNOSCHUNITS については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2(122) PREWAITNOSCHUNITS」を参照してください。

- 実行を開始する

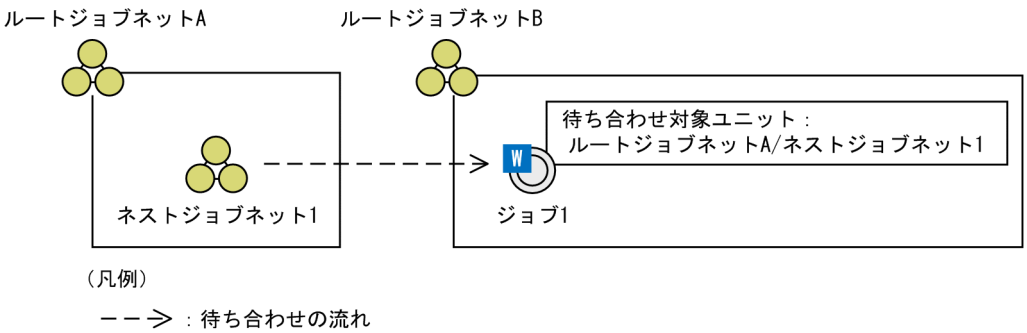
  待ち合わせ対象ユニットに待ち合わせる世代がないと，すぐに実行を開始します。複数の待ち合わせ対象ユニットと待ち合わせをする場合，実行終了していないすべての待ち合わせ対象ユニットに待ち合わせる世代がないと，実行を開始します。

なお，この設定は，待ち合わせ条件付きユニットと待ち合わせ対象ユニットが異なるルートジョブネット配下にある場合に有効です。待ち合わせ条件付きユニットと待ち合わせ対象ユニットが同じルートジョブネット配下にある場合，両ユニットは同じ世代で実行されるため，待ち合わせ対象ユニットに待ち合わせルールを満たす世代は必ず存在します。

待ち合わせ対象ユニットに待ち合わせる世代がない場合の設定による動作の違いを，次の図に示します。

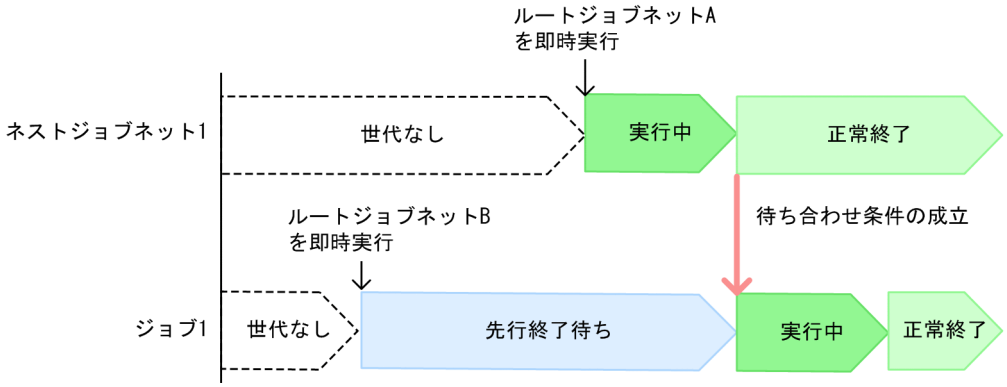
図 2-54　待ち合わせ対象ユニットに待ち合わせる世代がない場合の設定による動作の違い

■ユニット構成

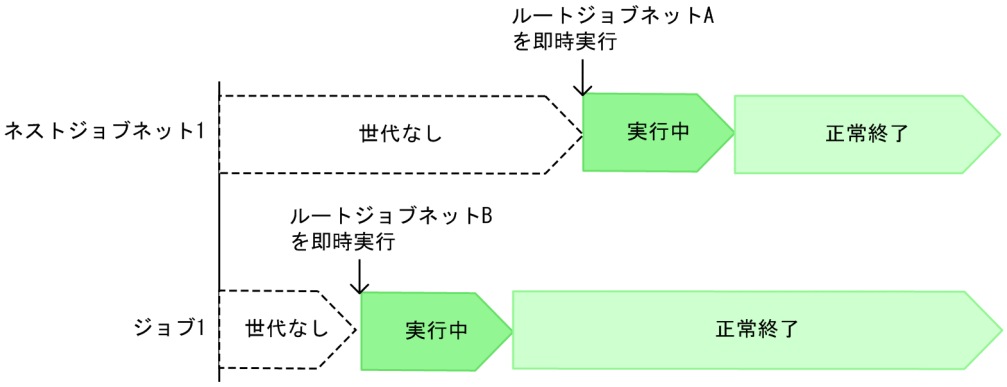


■状態遷移

＜例1：待ち合わせ対象ユニットに待ち合わせる世代がない場合の設定：実行を開始しない＞



＜例2：待ち合わせ対象ユニットに待ち合わせる世代がない場合の設定：実行を開始する＞



例 1 では，待ち合わせ対象ユニットに待ち合わせる世代がない場合の設定を，「実行を開始しない」にしています。待ち合わせ条件付きユニットのジョブ 1 の上位ユニットであるルートジョブネット B を即時実行すると，ジョブ 1 の世代が生成されます。このとき，待ち合わせ対象ユニットのネストジョブネット 1 に世代がないため，ジョブ 1 は「先行終了待ち」状態に遷移して待ち合わせ条件の成立を待ち続けます。ネストジョブネット 1 の上位ユニットであるルートジョブネット A を即時実行してネストジョブネット 1 が正常終了すると，待ち合わせ条件が成立して，ジョブ 1 が実行を開始します。

例 2 では，待ち合わせ対象ユニットに待ち合わせる世代がない場合の設定を，「実行を開始する」にしています。ルートジョブネット B を即時実行すると，ジョブ 1 の世代が生成されます。このとき，待ち合わせ対象ユニットのネストジョブネット 1 に世代がないため，ジョブ 1 はすぐに実行を開始します。ジョブ 1

の実行開始後にネストジョブネット 1 の上位ユニットであるルートジョブネット A を即時実行して，ネストジョブネット 1 が実行終了しても，ジョブ 1 は実行を開始しません。

補足事項

- 待ち合わせ対象ユニットが「未計画」状態でも，待ち合わせ対象ユニットのルートジョブネットに実行予定がある場合，待ち合わせる世代がない場合の設定は有効になりません。この場合，ルートジョブネットを実行すると，待ち合わせ対象ユニットは「計画未実行」状態に遷移します。そのため，待ち合わせ対象ユニットは実行終了したと見なされ，待ち合わせ条件が成立します。
- 「実行を開始する」を指定していても，待ち合わせ対象ユニットのルートジョブネットが「未登録」状態の場合，待ち合わせ条件付きユニットは実行を開始しません。

(c) 待ち合わせ対象ユニットに起動条件を使用する場合の，待ち合わせ条件の設定

待ち合わせ対象ユニットに起動条件を使用する場合，待ち合わせ条件の設定を変更することで，待ち合わせ対象ユニットの実行終了とする状態を次のように変更できます。

- 待ち合わせ対象ユニットの監視世代が「監視未起動終了」状態で終了した場合
  監視世代が「監視未起動終了」状態で終了した場合に，待ち合わせ条件が成立するようにできます。
- 待ち合わせ対象ユニットの実行世代が異常終了した場合
  実行世代が異常終了した場合に，待ち合わせ条件が成立するようにできます。
  なお，実行世代の異常終了状態が「繰り越し未実行」状態の場合には，待ち合わせ条件が成立するようにもできます。

次に，待ち合わせ条件の設定を変更する例を示します。

- イベントの監視中に起動条件が成立しないで終了する場合
  起動条件の有効範囲に指定した時間内に 1 回も起動条件が成立しない場合でも，運用上は正常として扱いたいときは，監視世代が「監視未起動終了」状態に遷移しても待ち合わせ条件が成立するようにできます。
- 実行世代が「異常検出終了」状態や「強制終了」状態になっても運用を継続する場合
  複数回起動条件が成立した場合に，生成された実行世代の一部が異常終了しても，運用上は正常として扱いたいときは，実行世代が異常終了しても待ち合わせ条件が成立するようにできます。
- 監視世代が「監視中」状態の間に起動条件が大量に成立しても，実行待ちの世代が滞留しないように設定している場合
  起動条件が大量に成立しても実行待ちの実行世代が滞留しないように設定している場合，実行を開始しない実行世代は「繰り越し未実行」状態に遷移します。実行世代が「繰り越し未実行」状態に遷移しても運用上は正常として扱いたいときは，実行世代が「繰り越し未実行」状態に遷移しても待ち合わせ条件が成立するようにできます。

■ 待ち合わせ対象ユニットに起動条件を使用する場合の，監視世代の実行終了の設定

待ち合わせ対象ユニットに起動条件を使用する場合に，監視世代の「監視未起動終了」状態への遷移を，監視世代の実行終了とするように設定できます。

待ち合わせ条件の設定にある「監視未起動終了の場合」の設定で，次のどちらかを選択します。

- 実行を開始しない（デフォルト）
  監視世代が「監視未起動終了」状態になっても，待ち合わせ条件付きユニットは実行を開始しません。
- 実行を開始する
  監視世代が「監視未起動終了」状態になったら，待ち合わせ条件付きユニットは実行を開始します。

この設定によって，起動条件付きジョブネットの実行終了とする状態は，次のようになります。

表 2-13 「監視未起動終了の場合」の設定と，起動条件付きジョブネットの実行終了とする状態

実行を開始しない（デフォルト）	実行を開始する
監視世代 • 監視正常終了 実行世代※ • 正常終了 • 警告検出終了	監視世代 • 監視正常終了 • 監視未起動終了 実行世代※ • 正常終了 • 警告検出終了

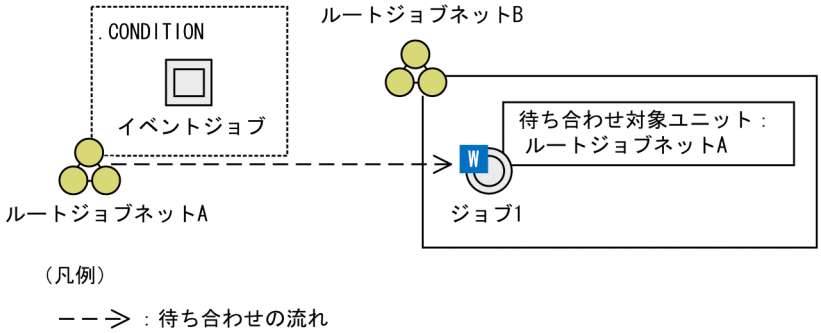
注※

実行世代の実行終了の状態も，待ち合わせ条件付きユニットの動作の設定で変更できます。ここでは，デフォルトの設定であると仮定します。

「監視未起動終了の場合」の設定の違いによる，待ち合わせ条件付きユニットの動作の違いを，次の図に示します。

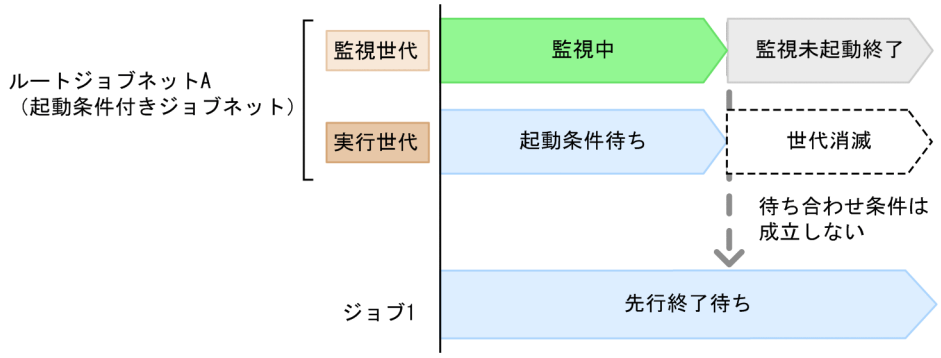
図 2-55 「監視未起動終了の場合」の設定の違いによる，待ち合わせ条件付きユニットの動作の違い

■ユニット構成

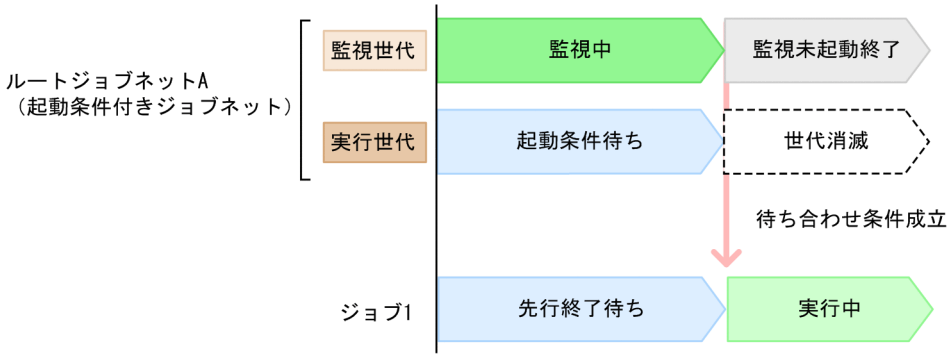


■状態遷移

＜例1：「監視未起動終了の場合」の設定が「実行を開始しない」の場合（デフォルトの動作）＞



＜例2：「監視未起動終了の場合」の設定が「実行を開始する」の場合＞



例 1 では，ジョブ 1 に定義した待ち合わせ条件の「監視未起動終了の場合」の設定を，「実行を開始しない」にしています。待ち合わせ対象ユニットのルートジョブネット A の監視世代が「監視中」状態の間，起動条件が 1 回も成立しないと，監視世代は「監視未起動終了」状態に遷移します。そのため，待ち合わせ条件付きユニットのジョブ 1 は「先行終了待ち」状態のまま，待ち合わせ条件の成立を待ち続けます。

例 2 では，ジョブ 1 に定義した待ち合わせ条件の「監視未起動終了の場合」の設定を，「実行を開始する」にしています。ルートジョブネット A の監視世代が「監視中」状態の間，起動条件が 1 回も成立しないと，監視世代は「監視未起動終了」状態に遷移します。この場合，待ち合わせ条件が成立して，待ち合わせ条件付きユニットのジョブ 1 は実行を開始します。

■ 待ち合わせ対象ユニットに起動条件を使用する場合の，実行世代の実行終了の設定

待ち合わせ対象ユニットに起動条件を使用する場合に，実行世代の「繰り越し未実行」状態，または異常終了を含むすべての終了状態への遷移を，実行世代の実行終了とするように設定できます。

待ち合わせ条件の設定にある「実行世代に異常終了がある場合」の設定で，次のどちらかを選択します。

- 実行を開始しない（デフォルト）

  実行世代が異常終了したら，待ち合わせ条件付きユニットは実行を開始しません。ただし，「繰り越し未実行」状態に限り，実行を開始するかどうかを選択できます。デフォルトでは，「繰り越し未実行」状態では実行を開始しません。

- 実行を開始する

  実行世代が異常終了しても，待ち合わせ条件付きユニットは実行を開始します。この設定をすると，すべての終了状態が実行終了と見なされます。

この設定によって，起動条件付きジョブネットの実行終了とする状態は，次のようになります。

表 2-14 「実行世代に異常終了がある場合」の設定と，起動条件付きジョブネットの実行終了とする状態

実行を開始しない		実行を開始する
「繰り越し未実行」状態でも開始しない（デフォルト）	「繰り越し未実行」状態なら開始する	
監視世代※ • 監視正常終了 実行世代 • 正常終了 • 警告検出終了	監視世代※ • 監視正常終了 実行世代 • 正常終了 • 警告検出終了 • 繰り越し未実行	監視世代※ • 監視正常終了 実行世代 • すべての終了状態

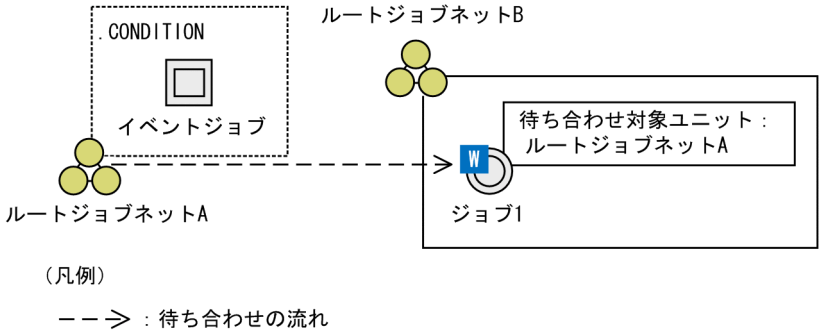
注※

監視世代の実行終了の状態も，待ち合わせ条件付きユニットの動作の設定で変更できます。ここでは，デフォルトの設定であると仮定します。

「実行世代に異常終了がある場合」の設定の違いによる，待ち合わせ条件付きユニットの動作の違いを，次の図に示します。

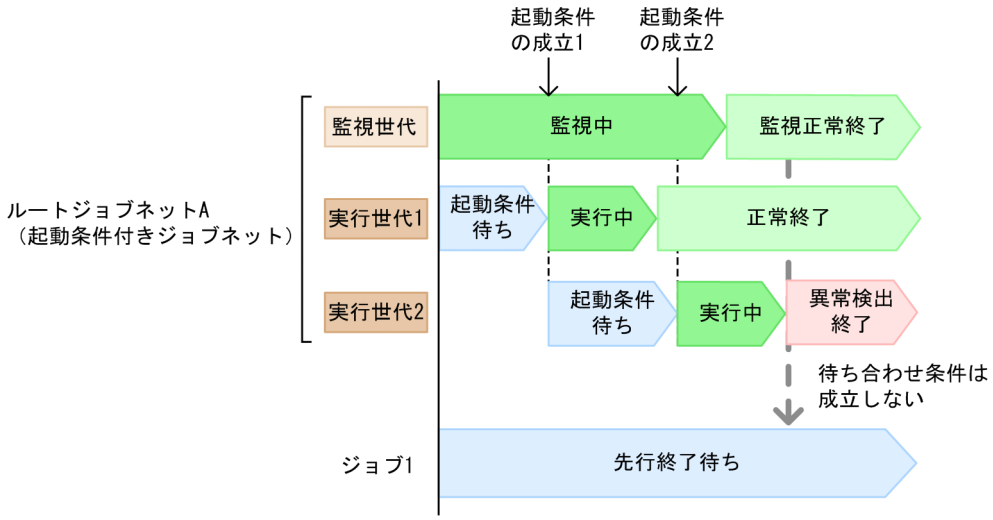
図 2-56 「実行世代に異常終了がある場合」の設定の違いによる，待ち合わせ条件付きユニットの動作の違い

■ユニット構成



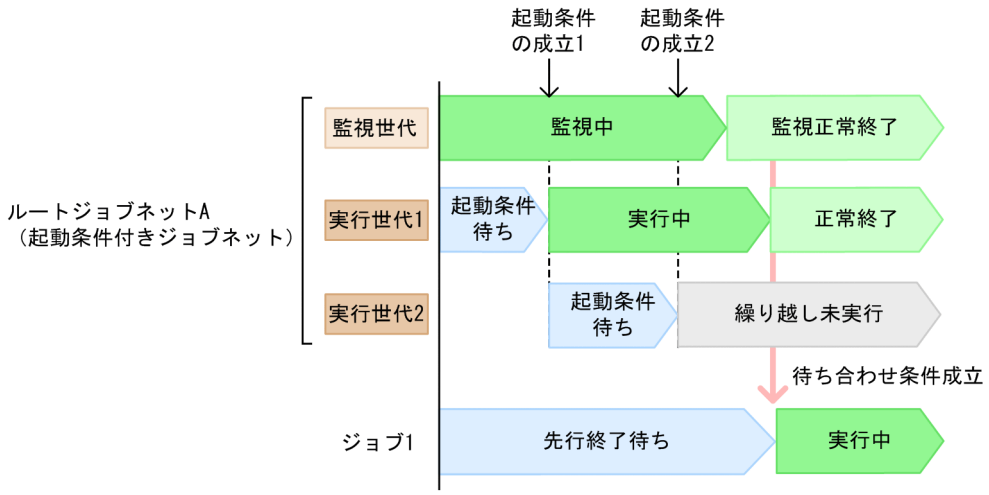
■状態遷移

＜例1：「実行世代に異常終了がある場合」の設定が「実行を開始しない」の場合＞



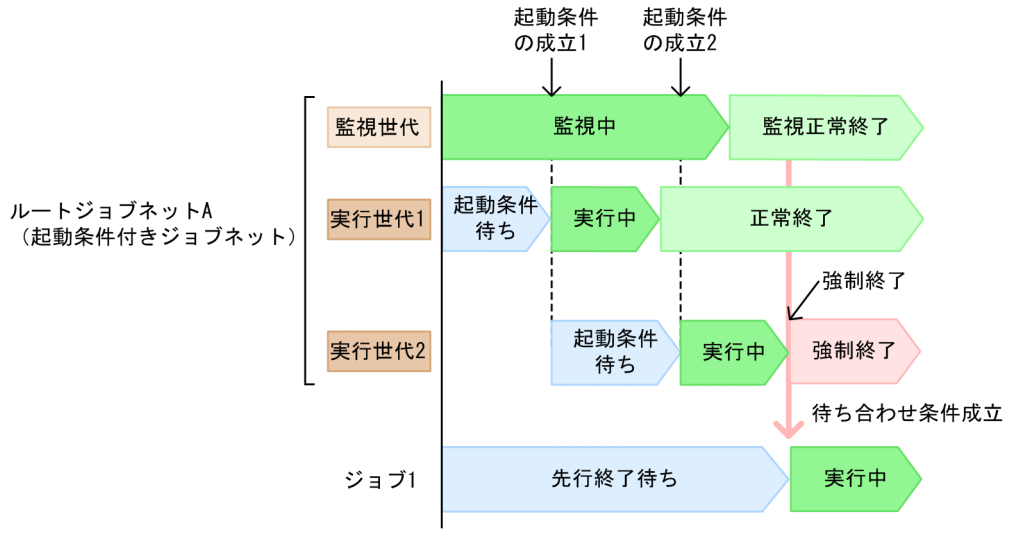
例 1 では，ジョブ 1 に定義した待ち合わせ条件の「実行世代に異常終了がある場合」の設定を，「実行を開始しない」にしています。実行世代 1 が正常終了して，実行世代 2 が異常終了すると，待ち合わせ条件付きユニットのジョブ 1 は「先行終了待ち」状態のまま，待ち合わせ条件の成立を待ち続けます。

<例2：「実行世代に異常終了がある場合」の設定が「実行を開始しない」で，
「繰り越し未実行」状態なら実行する場合>



例２では，ジョブ１に定義した待ち合わせ条件の「実行世代に異常終了がある場合」の設定を，「実行を開始しない」にしています。ただし，「繰り越し未実行」状態なら実行を開始する設定にしています。また，ルートジョブネット A は実行世代を滞留させないように設定しています。この場合，実行世代１の実行中に起動条件が成立して実行世代２が「繰り越し未実行」状態に遷移しても，実行世代１が「正常終了」状態に遷移すると待ち合わせ条件が成立します。

<例3：「実行世代に異常終了がある場合」の設定が「実行を開始する」の場合>



例３では，ジョブ１に定義した待ち合わせ条件の「実行世代に異常終了がある場合」の設定を，「実行を開始する」にしています。この場合，実行世代２の実行中に実行世代２を強制終了して「強制終了」状態に遷移させても，待ち合わせ条件が成立します。

注意事項

「実行世代に異常終了がある場合」の設定は，保存されている世代を対象にして判定されます。そのため，例えば保存世代数を超過したことによって実行世代の一部の保存世代が削除されると，残された実行世代から待ち合わせ条件の成立が判定されます。つまり，「実行世代に異常終了がある場合」の設定

を「実行を開始しない」としていても，異常終了した実行世代が削除されて残された世代がすべて正常終了であると，待ち合わせ条件が成立します。

待ち合わせ対象ユニットに起動条件付きジョブネットを指定する場合は，待ち合わせ中に実行世代の保存世代が削除されないように，起動条件によって起動する実行世代の数も考慮して保存世代数を見積もってください。

起動条件付きジョブネットの保存世代数については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.2.3(3) 起動条件付きジョブネットの保存世代数管理の例」，および「7.2 保存世代数と性能との関係」を参照して，設定値を検討してください。

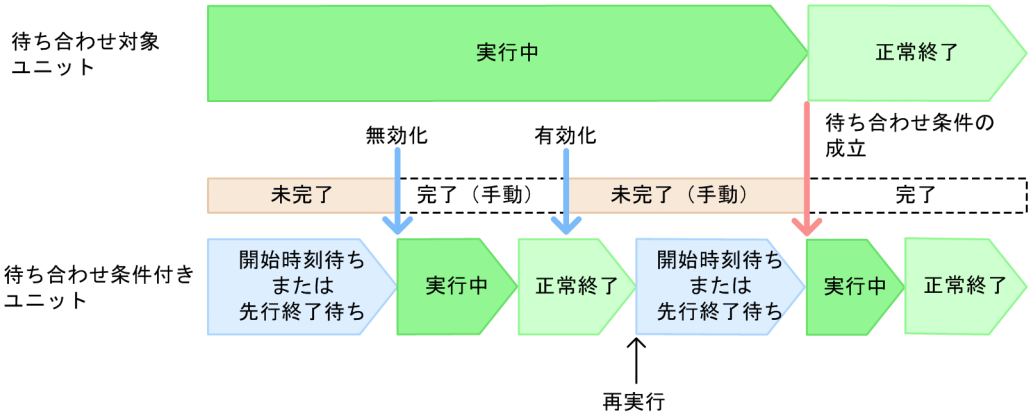
(6)　待ち合わせ条件の一時変更

待ち合わせ条件で指定した待ち合わせ対象ユニットとの待ち合わせを，一時的に有効／無効に変更できます。

実行終了した待ち合わせ条件付きユニットの待ち合わせを有効にすると，待ち合わせ状態が「未完了（手動）」になります。このユニットを再実行すると，待ち合わせを再開します。また，待ち合わせ対象ユニットの実行終了を待ち合わせている待ち合わせ条件付きユニットの待ち合わせを無効にすると，待ち合わせ状態が「完了（手動）」になります。待ち合わせ方法が「AND」の場合，すべての待ち合わせ状態を「完了（手動）」にすると，待ち合わせ条件付きユニットは実行を開始します。待ち合わせ方法が「OR」の場合，待ち合わせ対象ユニットのどれか一つの待ち合わせ状態を「完了（手動）」に変更すると，待ち合わせ条件付きユニットは実行を開始します。

待ち合わせ条件の有効化／無効化と待ち合わせ状態の関係を，次の図に示します。

図 2-57　待ち合わせ条件の有効化／無効化と待ち合わせ状態の関係



待ち合わせ対象ユニットが実行中で，待ち合わせ条件付きユニットが「開始時刻待ち」，「先行終了待ち」，または「未計画※」状態にあるとき，待ち合わせ状態は「未完了」です。この状態で待ち合わせを無効にすると，待ち合わせ状態は「完了（手動）」になり，待ち合わせ条件付きユニットが実行を開始します。待ち合わせ条件付きユニットの実行終了後，待ち合わせを有効にすると，待ち合わせ状態は「未完了（手動）」に遷移します。この状態で待ち合わせ条件付きユニットを再実行すると，待ち合わせ条件付きユニットは待ち合わせを再開します。そのため，待ち合わせ対象ユニットが正常終了すると，待ち合わせ状態は「完了」に遷移して待ち合わせ条件付きユニットは実行を開始します。

注※

環境設定パラメーターPREWAITNOSCHUNITSで「yes」を指定している場合は，待ち合わせ条件付きユニットが「未計画」状態でも待ち合わせが成立します。環境設定パラメーターPREWAITNOSCHUNITSについては，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2(122) PREWAITNOSCHUNITS」を参照してください。

待ち合わせ条件の設定の一時変更は，［待ち合わせ条件の状態一覧］ウィンドウの［操作］－［待ち合わせ］－［待ち合わせ有効化］または［待ち合わせ無効化］を選択することで実行できます。

待ち合わせ条件の設定の一時的な変更については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.5.15 ジョブネットやジョブの待ち合わせ条件の設定を一時変更する」を参照してください。

(7) 待ち合わせ対象ユニットの定義が不正な場合の待ち合わせ条件付きユニットの動作

待ち合わせ条件を使用してユニット間の実行順序を制御する場合に，待ち合わせ対象ユニットに対して次のような操作をすると，待ち合わせ条件で実行順序を制御できなくなります。

- 待ち合わせ条件の設定時に，待ち合わせ対象ユニットとして指定できないユニット種別のユニットを指定する。
- 待ち合わせ条件の設定後に，待ち合わせ対象ユニットを削除する。
- 待ち合わせ条件の設定後に，待ち合わせ対象ユニットの名称を変更する。

このことを，待ち合わせ対象ユニットの定義不正といいます。待ち合わせ対象ユニットの定義不正が発生した場合の待ち合わせ条件付きユニットの動作は，待ち合わせ対象ユニットを実行登録しているかどうかと，待ち合わせ対象ユニットがサスペンド中かどうかで異なります。

待ち合わせ対象ユニットが未登録またはサスペンド中の場合

メッセージKAVS4957-EまたはKAVS4971-Eが出力され，待ち合わせ条件付きユニットの実行は開始されません。

待ち合わせ対象ユニットがサスペンド中である場合，そのユニットが実行終了しても待ち合わせは完了しません。また，異常終了したり，削除されたりしても，待ち合わせ条件付きユニットは異常終了したり，メッセージを出力したりしません。このような場合の待ち合わせ条件付きユニットの動作は，待ち合わせ対象ユニットのサスペンドが解除された時点の，待ち合わせ対象ユニットの状態に依存します。

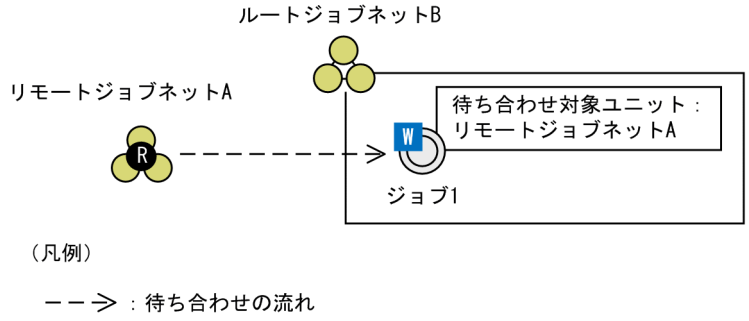
待ち合わせ対象ユニットが実行登録済みでサスペンド中ではない場合

メッセージKAVS4954-Eが出力され，待ち合わせ条件付きユニットが異常終了します。

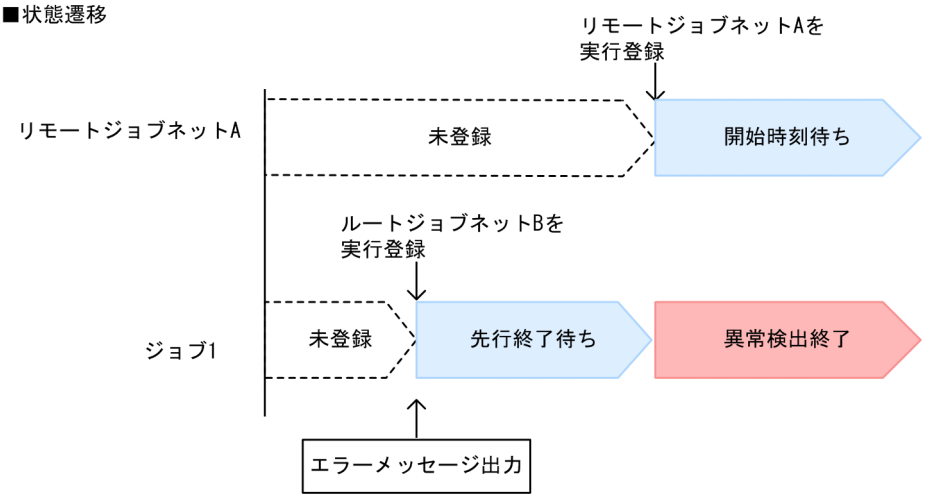
待ち合わせ対象ユニットの定義不正の例を，次の図に示します。

図 2-58　待ち合わせ対象ユニットの定義不正の例

■ユニット構成



■状態遷移



この例では，待ち合わせ対象ユニットにリモートジョブネットを指定しているため，定義不正になっています。リモートジョブネット A を未登録のままルートジョブネット B を実行登録すると，待ち合わせ条件付きユニットのジョブ 1 は「先行終了待ち」状態で待ち合わせ対象ユニットの実行終了を待ち合わせます。このとき，待ち合わせ対象ユニットが未登録であることを知らせるエラーメッセージが出力されます。そのあと，リモートジョブネット A を実行登録すると，リモートジョブネット A には実行予定世代が生成されますが，ジョブ 1 は待ち合わせ対象ユニットの定義不正のため異常終了します。

なお，待ち合わせ条件の設定後，待ち合わせ対象ユニットのルートジョブネットやプランニンググループを削除・移動したり名称を変更したりした場合も，待ち合わせ対象ユニットの定義不正です。この場合，待ち合わせ条件付きユニットは実行開始後，すぐに異常終了します。

待ち合わせ方法に「OR」を指定している場合，待ち合わせ対象ユニットの定義不正を検知する前にほかの待ち合わせ対象ユニットが実行終了して待ち合わせ条件が成立すると，待ち合わせ条件付きユニットの実行が開始されます。しかし，待ち合わせ対象ユニットの定義不正を先に検知すると，待ち合わせ条件付きユニットは異常終了します。

待ち合わせ対象ユニットの定義不正の防止方法

待ち合わせ対象ユニットの定義不正を防止するために，待ち合わせ対象ユニットのユニット名を変更したり，待ち合わせ対象ユニットを削除したりした場合，待ち合わせ条件付きユニットに設定してある待ち合わせ条件もあわせて変更してください。

また，待ち合わせ対象ユニットの定義不正を事前にチェックすることもできます。チェック方法には，次の二つがあります。

- 定義内容の事前チェック

  定義内容の事前チェック機能を使用すると，待ち合わせ条件が正しいかチェックできます。定義内容の事前チェックについては，「8. 定義内容の事前チェック」を参照してください。

  なお，次の点はチェックできないため，待ち合わせ条件設定後に動作を確認する必要があります。

  ・待ち合わせ条件による実行順序のループ

  ・プランニンググループ配下のネストジョブネットやジョブを待ち合わせ対象ユニットとした場合の定義不正

  ・起動条件を使用した場合の定義不正

- ［待ち合わせ条件の設定一覧］ウィンドウでのチェック

  JP1/AJS3 - View の［待ち合わせ条件の設定一覧］ウィンドウで，待ち合わせ対象ユニットの定義不正をチェックできます。［待ち合わせ条件の設定一覧］ウィンドウでの待ち合わせ対象ユニットの確認方法については，「(8) 待ち合わせ対象ユニットの確認方法」を参照してください。

(8) 待ち合わせ対象ユニットの確認方法

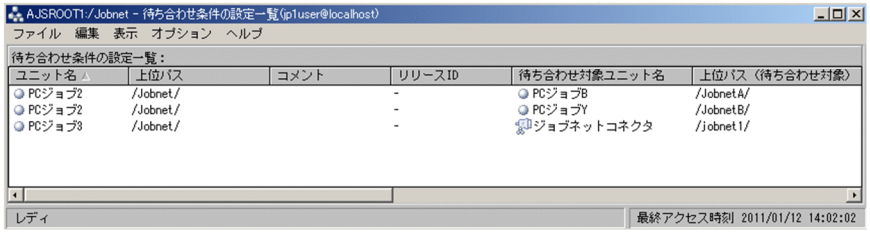
JP1/AJS3 - View またはajsshow コマンドを使って待ち合わせ条件付きユニットを定義・監視する場合，次の方法を使うと，待ち合わせ条件付きユニットと待ち合わせ対象ユニットを一覧で確認できます。

待ち合わせ条件の設定内容を確認する

JP1/AJS3 - View の［待ち合わせ条件の設定一覧］ウィンドウは，待ち合わせ条件の設定内容を確認するときに使用します。

［待ち合わせ条件の設定一覧］ウィンドウを，次の図に示します。

図 2-59 ［待ち合わせ条件の設定一覧］ウィンドウ



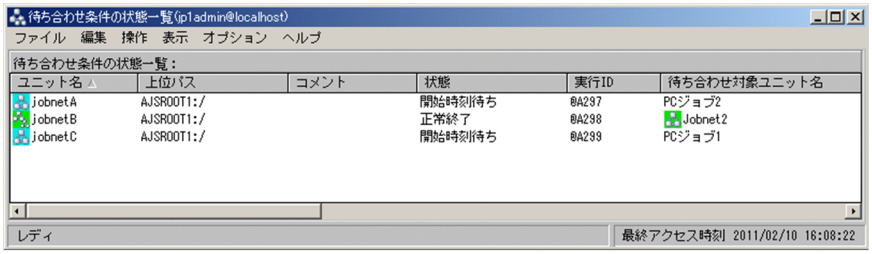
任意のユニットの配下にあるすべての待ち合わせ条件付きユニットを一覧で表示して，それぞれの待ち合わせ条件付きユニットに対して設定してある待ち合わせ対象ユニットを表示します。それぞれの待ち合わせ対象ユニットに対して，定義不正がないかを確認したり，待ち合わせ方法および待ち合わせる世代がない場合の動作の設定を確認したりできます。

待ち合わせ条件の状態を確認する

JP1/AJS3 - View の［待ち合わせ条件の状態一覧］ウィンドウおよびajsshow コマンドの-xw オプションは，運用中の待ち合わせ条件付きユニットに対して設定されている待ち合わせ対象ユニットの状態を確認するときに使用します。

［待ち合わせ条件の状態一覧］ウィンドウを，次の図に示します。

図 2-60 ［待ち合わせ条件の状態一覧］ウィンドウ



任意の待ち合わせ条件付きユニットを一覧で表示して，それぞれの待ち合わせ条件付きユニットに対して設定してある待ち合わせ対象ユニットを表示します。待ち合わせ対象ユニットの［ジョブネットモニタ］ウィンドウを表示しなくても，待ち合わせ対象ユニットが正常に実行しているかどうかを確認できます。待ち合わせ状態を確認することもできます。

なお，待ち合わせ対象ユニットに起動条件を使用している場合，待ち合わせ対象ユニットの情報には監視世代の情報が表示されます。監視世代の状態と待ち合わせ状態から，待ち合わせ対象ユニットの実行状況を把握できます。詳細については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　8.3.3(1) 待ち合わせ対象ユニットが起動条件を使用する場合に，待ち合わせの状況を確認する」を参照してください。

また，待ち合わせ条件付きユニットの待ち合わせを，待ち合わせ対象ユニットごとに有効／無効にしたり，待ち合わせ対象ユニットの状態を表示する［ジョブネットモニタ］ウィンドウを開いたりすることもできます。

`ajsshow` コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajsshow」を参照してください。

注意事項

- 待ち合わせ条件付きユニットに対して設定してある待ち合わせ対象ユニットの数が多い場合，［待ち合わせ条件の状態一覧］ウィンドウの表示に時間が掛かります。「(5)(a) 待ち合わせ対象ユニットが複数ある場合の待ち合わせ方法」の注意事項を参考に，待ち合わせ対象ユニットの数はできるだけ減らしてください。
- ［待ち合わせ条件の状態一覧］ウィンドウから待ち合わせ対象ユニットの［ジョブネットモニタ］ウィンドウを開くには，待ち合わせ対象ユニット，およびその上位ジョブネットに対して JP1_AJS_Guest 権限が必要です。
- ［待ち合わせ条件の状態一覧］ウィンドウおよび`ajsshow` コマンドで確認できる待ち合わせ対象ユニットの情報は，常に待ち合わせルールに基づいて求めた情報を表示します。そのため，待ち合わせの有効化と再実行によって，再度待ち合わせを開始する際に，待ち合わせ条件付きユニットがどのように動作するか（待ち合わせ対象ユニットの実行終了を待つか，待ち合わせがすぐ完了して実行を開始するか，など）を予測することができます。

  一方，待ち合わせ対象ユニットや待ち合わせ条件付きユニットに世代の追加・削除などの操作を行うと，待ち合わせルールによって求まる待ち合わせ対象ユニットの世代が変更されるため，待ち合わせ状態が「完了」になったときの待ち合わせ対象ユニットの情報とは異なる世代の情報が表示される場合があります。常に待ち合わせ状態が「完了」になったときの待ち合わせ対象ユニットの情報を表示するには，待ち合わせ条件付きユニットと待ち合わせ対象ユニットの世代を，どちらも 1 日分の世代数を保存するようにして，求まる待ち合わせ対象ユニットの世代が一

定になるようにしてください。待ち合わせルールについては，「2.2.5(2) 待ち合わせ条件付きユニットと待ち合わせ対象ユニットの待ち合わせルール」を参照してください。

［待ち合わせ条件の設定一覧］ウィンドウおよび［待ち合わせ条件の状態一覧］ダイアログボックスの詳細については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12. ウィンドウとダイアログボックス」を参照してください。

(9)　待ち合わせ条件を使用した実行順序制御とジョブネットコネクタを使用した実行順序制御の使い分け

異なるルートジョブネット同士の実行順序を制御したい場合，待ち合わせ条件を設定したり，ジョブネットコネクタを使用したりして，ユニット同士を連携できます。

運用に合わせて，ジョブネットコネクタと待ち合わせ条件のどちらを使用するかを選択してください。

待ち合わせ条件を使用した場合と，ジョブネットコネクタを使用した場合の長所と短所を，次の表に示します。

表 2-15　待ち合わせ条件を使用した場合とジョブネットコネクタを使用した場合の長所と短所

観点	待ち合わせ条件	ジョブネットコネクタ
ユニット構成に関する長所と短所	長所 後続側のユニット（待ち合わせ条件付きユニット）に待ち合わせ条件を設定するだけで定義できるため，ジョブネットの構造を変更する必要がない。 短所 特になし。	長所 特になし。 短所 • ジョブネットコネクタを新しく定義する必要がある。 • ネストジョブネットを接続先のジョブネットとして定義したい場合，ネストジョブネットを新しくルートジョブネットとして分割する必要がある。
ユニットの詳細定義に関する長所と短所	長所 先行側のユニット（待ち合わせ対象ユニット）の詳細定義を編集する必要がない。 短所 特になし。	長所 特になし。 短所 接続先のジョブネットの詳細定義に，接続範囲やジョブネットコネクタ名を定義する必要がある。
連携方法の設定に関する長所と短所	長所 • ルートジョブネット，ネストジョブネット，および各種ジョブ同士を連携できる。 • 一つの待ち合わせ条件付きユニットに対して，複数の待ち合わせ対象ユニットを設定できる。 短所 異なるスケジューラーサービスのユニット同士を連携できない。連携させたい場合は，ジョブ	長所 異なるスケジューラーサービスにあるルートジョブネット同士でも連携できる。そのため，連携方法をジョブネットコネクタに統一できる。 短所 • 連携できるユニットは，ルートジョブネットだけである。

観点	待ち合わせ条件	ジョブネットコネクタ
連携方法の設定に関する長所と短所	ネットコネクタなどを利用してジョブネットを定義し直す必要がある。そのため，待ち合わせ条件とそれ以外の方法を併用する必要がある。	• 一つのジョブネットコネクタにつき，連携できるルートジョブネットは一つだけである。
ユニットの監視に関する長所と短所	長所 ［待ち合わせ条件の状態一覧］ウィンドウを使用して，待ち合わせ対象ユニットと待ち合わせ条件付きユニットを一覧で確認できる。 短所 • 待ち合わせ対象ユニットと待ち合わせ条件付きユニットは，関連線で関連づけられないため，実行順序が把握しづらい。 • 待ち合わせ対象ユニットの状態が，待ち合わせ条件付きユニットの状態に反映されない。	長所 • ジョブネットコネクタをほかのユニットと関連線で関連づけられるため，実行順序が把握しやすい。 • 接続先のジョブネットの状態がジョブネットコネクタに反映されるため，ジョブネットコネクタを定義しているルートジョブネットを監視すれば，接続先のジョブネットの状態を一元監視できる。 短所 特になし。

2.2.6　マクロ変数の使用を検討する

マクロ変数を使用すると，ジョブの実行ごとに変化するパラメーターの値などを，後続ユニットで使用して業務を自動化できます。

(1)　マクロ変数を指定できる定義項目

マクロ変数を使用して情報を引き継ぐには，後続ユニットの詳細定義でマクロ変数を指定します。

マクロ変数を指定できる定義項目を，次に示します。

表 2-16　マクロ変数を指定できる定義項目一覧

ユニット種別	定義項目
ジョブネット	実行エージェント※
UNIX ジョブ	実行エージェント※
	コマンド文
	スクリプトファイル名
	パラメーター
	環境変数
	環境変数ファイル名※
	ワークパス※
	標準入力ファイル名※

ユニット種別	定義項目
UNIX ジョブ	標準出力ファイル名※
	標準エラー出力ファイル名※
	終了判定のファイル名（終了判定の条件に「ファイルが存在すれば正常」，または「ファイルが更新されれば正常」を指定した場合）※
	実行時のユーザー※
	転送元ファイル名※
	転送先ファイル名※
PC ジョブ	実行エージェント※
	実行ファイル名
	パラメーター
	環境変数
	環境変数ファイル名※
	ワークパス※
	標準入力ファイル名※
	標準出力ファイル名※
	標準エラー出力ファイル名※
	終了判定のファイル名（終了判定の条件に「ファイルが存在すれば正常」，または「ファイルが更新されれば正常」を指定した場合）※
	実行時のユーザー※
	転送元ファイル名※
	転送先ファイル名※
QUEUE ジョブ	ホスト名※
	キュー名※
	ジョブ名※
	実行ファイル名
	パラメーター
	転送元ファイル名※
	転送先ファイル名※
判定ジョブ	変数名
フレキシブルジョブ	宛先

ユニット種別	定義項目
フレキシブルジョブ	中継エージェント
	実行ファイル名
	パラメーター
	環境変数
	終了判定のファイル名（終了判定の条件に「ファイルが存在すれば正常」，または「ファイルが更新されれば正常」を指定した場合）
HTTP 接続ジョブ	実行エージェント
	接続設定ファイル名
	送信情報ファイル名
	送信情報ファイル名（URL パラメーター）
	送信情報ファイル名（メッセージボディ）
	ステータス格納ファイル名
	受信ヘッダー格納ファイル名
	受信ボディ格納ファイル名
	標準出力ファイル名
	標準エラー出力ファイル名
	終了判定ファイル名（終了判定の条件に「ファイルが存在すれば正常」，または「ファイルが更新されれば正常」を指定した場合）
	実行時のユーザー
JP1 イベント受信監視ジョブ	実行エージェント※
	イベント発行元ホスト名※
JP1 イベント受信監視ジョブー拡張属性指定	任意の拡張属性※
ファイル監視ジョブ	実行エージェント※
	監視対象ファイル名※
メール受信監視ジョブ	実行エージェント※
ログファイル監視ジョブ	実行エージェント※
	ログファイル名※
Windows イベントログ監視ジョブ	実行エージェント※
実行間隔制御ジョブ	実行エージェント※
JP1 イベント送信ジョブ	実行エージェント※
	イベント送信先ホスト名

ユニット種別	定義項目
JP1 イベント送信ジョブ	メッセージ
	拡張属性
メール送信ジョブ	実行エージェント※
	宛先
	件名
	本文
	本文－ファイル名
	プロファイル名
メール送信ジョブ－添付ファイル	添付ファイル
	添付ファイル－リストファイル名
メッセージキュー送信ジョブ－詳細設定	メッセージデータファイル名
MSMQ 送信ジョブ	キューパス名
	キューラベル名
	メッセージラベル
MSMQ 送信ジョブ－メッセージ本文	本文ファイル名
JP1/Cm2 状態通知ジョブ	実行エージェント※
	付加情報
ローカル電源制御ジョブ	実行エージェント※
リモート電源制御ジョブ	実行エージェント※
	対象ホスト
カスタム UNIX ジョブ	実行エージェント※
	コマンド文
	スクリプトファイル名
	パラメーター
	環境変数
	環境変数ファイル名※
	ワークパス※
	標準入力ファイル名※
	標準出力ファイル名※
	標準エラー出力ファイル名※

ユニット種別	定義項目
カスタム UNIX ジョブ	終了判定のファイル名（終了判定の条件に「ファイルが存在すれば正常」，または「ファイルが更新されれば正常」を指定した場合）※
	実行時のユーザー※
	転送元ファイル名※
	転送先ファイル名※
カスタム PC ジョブ	実行エージェント※
	実行ファイル名
	パラメーター
	環境変数
	環境変数ファイル名※
	ワークパス※
	標準入力ファイル名※
	標準出力ファイル名※
	標準エラー出力ファイル名※
	終了判定のファイル名（終了判定の条件に「ファイルが存在すれば正常」，または「ファイルが更新されれば正常」を指定した場合）※
	実行時のユーザー※
	転送元ファイル名※
	転送先ファイル名※
標準カスタムジョブ	実行エージェント※
	標準出力ファイル名※
	標準エラー出力ファイル名※
	終了判定のファイル名（終了判定の条件に「ファイルが存在すれば正常」，または「ファイルが更新されれば正常」を指定した場合）※
	実行時のユーザー※

注※

JP1/AJS3 - View および JP1/AJS3 - Manager のバージョンが 09-50 以降の場合に，マクロ変数を指定できます。

(2) マクロ変数の確認

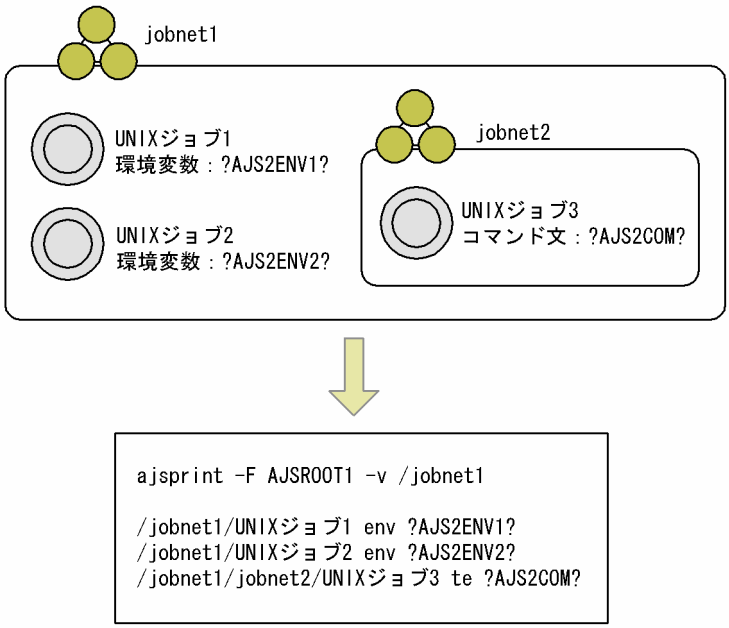
マクロ変数を使用している場合，ジョブで使用しているマクロ変数名を確認したり，ジョブの実行後にマクロ変数で引き継いだ結果を確認したりできます。

(a) 使用しているマクロ変数の確認

JP1/AJS3 - View の［変数使用状況］ダイアログボックス，またはajsprint コマンドの-v オプションの指定で，配下のユニットで使用しているマクロ変数名の一覧を参照できます。詳細については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12.3.25 ［変数使用状況］ダイアログボックス」，およびマニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajsprint」を参照してください。

ajsprint コマンドでの参照例を次の図に示します。

図 2-61　配下のユニットで使用しているマクロ変数の参照例



(b) マクロ変数で引き継いだ結果の確認

JP1/AJS3 - View の［引き継ぎ結果］ダイアログボックスでは，実行登録時に指定したマクロ変数名と引き継ぎ情報，およびジョブ実行時に引き継いだマクロ変数名と引き継ぎ結果が参照できます。詳細については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12.7.10 ［引き継ぎ結果］ダイアログボックス」を参照してください。

また，ajsshow コマンドでは，-i オプションで 2 バイトフォーマット指示子%MV を指定すると，実行登録時に指定したマクロ変数名と引き継ぎ情報，ジョブ実行時に引き継いだマクロ変数名と引き継ぎ結果が参照できます。出力結果が複数ある場合は「,（コンマ）」で区切られて出力されます。出力世代が複数ある場合は改行されて出力されます。

ajsshow コマンドの出力例を次に示します。

```
ajsshow -F AJSROOT2 -g 2 -i "%MV" /net

"AJS2ENV:/jp1_data/Job_Report","AJS2COM:c:¥temp¥test.exe"
"AJS2ENV:/jp1_data/Job_Report","AJS2COM:c:¥temp¥test.exe"

ajsshow -F AJSROOT2 -i "%MV" /net2

"AJS2ENV:/jp1_data2/Job△Report","AJS2COM:c:¥temp¥test2.exe"
```

ajsshow コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajsshow」を参照してください。

なお，ユニット種別および状態によって，引き継ぎ結果の参照可否が異なります。ユニット種別および状態による引き継ぎ結果の参照可否を，次の表に示します。

表 2-17　引き継ぎ結果の参照可否

ユニット種別	状態	参照可否
ルートジョブネット	未登録	×
	上記以外	○※1
ネストジョブネット	－	×
ジョブ※2，※3	未登録	×
	未計画	×
	先行終了待ち	×
	保留中	×
	キューイング	×
	実行待ち	×
	実行中	×
	正常終了	○
	正常終了-偽	○
	警告検出終了	○
	異常検出終了	○
	起動失敗	○
	終了状態不明	○
	強制終了	○
	未実行終了	×
	計画未実行	×
	監視中断	○

ユニット種別	状態	参照可否
ジョブ※2，※3	閉塞	×

（凡例）

○：参照できる。

×：参照できない。

－：該当しない。

注※1

リモートジョブネットは対象外です。

注※2

次に示すジョブは対象外です。

・リモートジョブネット配下のジョブ

・サスペンド中に削除されたジョブ

・サスペンド中に追加されたジョブ（ただし，サスペンド解除後は対象）

注※3

ジョブ状態を変更したジョブは，変更後の状態に依存します。

マクロ変数を設定しているジョブを再実行する場合，前回実行時の引き継ぎ情報がそのまま使用されます。そのため，前回実行時の引き継ぎ結果を参照することによって，再実行するジョブに引き継がれる情報を事前に確認できます。

補足事項

次の場合，前回実行時に引き継がれた結果と異なる結果が表示されます。

- 引き継ぎ元の先行ジョブが再実行中の場合
  その先行ジョブから引き継ぐ情報は表示されません。
- 引き継ぎ元の先行ジョブを再実行し，実行が終了した場合
  その先行ジョブの再実行後の引き継ぎ情報が表示されます。
- サスペンド中に引き継ぎ元の先行ジョブを削除した場合
  その先行ジョブから引き継がれていた情報は表示されません。
- ジョブネット中に，同名のマクロ変数を指定している引き継ぎ情報設定ジョブを複数定義している場合
  関連線に関係なく，最後に実行された引き継ぎ情報設定ジョブによって設定された引き継ぎ情報が表示されます。
- 引き継ぎ情報設定ジョブに指定している正規表現を変更してから，その引き継ぎ情報設定ジョブを再実行し，実行が終了した場合
  引き継ぎ情報設定ジョブの再実行によって，変更後の正規表現に従って設定された引き継ぎ情報が表示されます。
- 先行ジョブが前回実行時とは異なる標準出力ファイルを出力した状態で，引き継ぎ情報設定ジョブを再実行し，実行が終了した場合
  引き継ぎ情報設定ジョブの再実行によって，再実行時の標準出力ファイルから設定された引き継ぎ情報が表示されます。

(3) マクロ変数の展開失敗時の状態設定

ジョブの実行時にマクロ変数の引き継ぎ情報の展開が失敗した場合，そのジョブの状態を「起動失敗」にするか，マクロ変数名を文字列として扱いそのままジョブネットを実行させるかを設定できます。

マクロ変数の引き継ぎ情報の展開が失敗した場合の状態は，環境設定パラメーターMACROCHANGEFAIL で設定できます。

マクロ変数の引き継ぎ情報の展開が失敗した場合の状態の設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.2.10 マクロ変数の展開失敗時の動作の設定」（Windows の場合），またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.2.10 マクロ変数の展開失敗時の動作の設定」（UNIX の場合）を参照してください。

(4) マクロ変数の使用方法

マクロ変数の使用方法を次に示します。詳細については，それぞれの項目の参照先を参照してください。

表 2-18　マクロ変数の使用方法

項番	使用方法	参照先
1	イベントジョブで受信したイベント情報を後続ユニットへ引き継ぐ	• 2.4.4(6) イベントジョブの受信情報の引き継ぎ • 7.6.7 引き継ぎ情報定義時の注意事項
2	ジョブ実行時の状況によって変わる情報を標準出力ファイルに出力して，後続ユニットへ引き継ぐ	2.4.9 動的に変わる情報を後続ユニットに引き継ぐ（引き継ぎ情報設定ジョブを使ったジョブネットの定義例）
3	実行登録時にジョブネットに指定した値を，配下のユニットへ引き継ぐ	マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.1.2 実行登録時のマクロ変数の指定」

(5) 注意事項

- 一つのルートジョブネット配下のマクロ変数名と引き継ぎ情報のサイズの合計が，4,096 バイトを超えないようにしてください。マクロ変数名と引き継ぎ情報のサイズの合計が 4,096 バイトを超えた場合，超えた部分の情報が引き継がれなくなったり，引き継ぎ情報設定ジョブが異常終了したりするおそれがあります。なお，一つのルートジョブネット配下のマクロ変数名と引き継ぎ情報のサイズの合計は，次の方法で見積もってください。
  1. ルートジョブネット配下のユニットで使用しているすべてのマクロ変数名を列挙する。
  2. 手順 1 で列挙したすべてのマクロ変数について，引き継ぐ情報の最大サイズを検討し，次の式でマクロ変数名と引き継ぎ情報のサイズの合計をそれぞれ算出する。
     マクロ変数名のサイズ ＋ **引き継ぎ情報の最大サイズ** ＋ 5（単位：バイト）
  3. 手順 2 で算出したすべての値を合計する。
- 一つのルートジョブネット配下に同じ名称のマクロ変数が複数ある場合，次の順番で有効になります。
  1. ジョブネット中（起動条件中以外）のイベントジョブの引き継ぎ情報

2. 起動条件中のイベントジョブの引き継ぎ情報

3. 実行登録時の引き継ぎ情報

4. 引き継ぎ情報設定ジョブで設定した引き継ぎ情報※

注※

一つのルートジョブネット配下に，同じ名称のマクロ変数を指定した引き継ぎ情報設定ジョブが複数ある場合，あとから実行した引き継ぎ情報設定ジョブの引き継ぎ情報が有効です。

一つのルートジョブネット配下には同じ名称のマクロ変数を指定しないことを推奨します。

- JP1/AJS3 09-50 以降で新たにマクロ変数を指定できるようになった定義項目にマクロ変数を指定しているユニットを，JP1/AJS3 - Manager のバージョンが 09-50 より前の環境で実行した場合，マクロ変数が展開されません。マクロ変数名がそのまま引き継ぎ情報として引き継がれます。

  対象となる定義項目については，「(1) マクロ変数を指定できる定義項目」またはマニュアル「JP1/Automatic Job Management System 3 操作ガイド　12. ウィンドウとダイアログボックス」の各種定義用ダイアログボックスの説明を参照してください。

- 引き継ぎ情報は世代単位にマクロ変数名と値の情報を保持しており，異なる世代に情報を引き継ぐことはできません。そのため，引き継ぎ情報は，異なる世代で実行されるユニットに引き継ぐことはできません。また，異なる世代で実行されたユニットから引き継ぐこともできません。

  異なる世代で実行されるユニットは次のとおりです。

  - ジョブネットコネクタの接続先のジョブネットとなるユニット（ルートジョブネットまたはプランニンググループ）
  - 待ち合わせ対象ユニットが別のルートジョブネット配下のときの，待ち合わせ対象ユニット

- リモートジョブネットを使用する場合，引き継ぎ情報は，リモートジョブネット配下のユニットに引き継ぐことはできません。また，リモートジョブネット配下のユニットから引き継ぐこともできません。

2.3　業務の自動化検討時のヒント

これまで挙げた項目以外の検討事項を検討時のヒントとして次に示します。

2.3.1　負荷を分散させた処理

JP1/AJS3 では，ジョブやジョブネットの実行先として実行エージェントグループを指定することで，その実行エージェントグループに属する実行エージェント間でジョブの実行を分散させることができます。負荷を分散するパターンは，次のとおりです。

- 複数の実行エージェントに，ジョブを均等に分散して実行する。
- それぞれの実行エージェントでのジョブ実行数の上限に差を付けて実行する。
- ある実行エージェントでのジョブ実行数が上限に達したら，別の実行エージェントに分散して実行する。

実行先として実行エージェントグループを指定できるユニットは次のものです。

- ルートジョブネット
- ネストジョブネット
- PC ジョブ
- UNIX ジョブ
- フレキシブルジョブ※
- HTTP 接続ジョブ
- アクションジョブ
- カスタムジョブ

注※
　フレキシブルジョブの場合，中継エージェントに実行エージェントグループを指定できます。

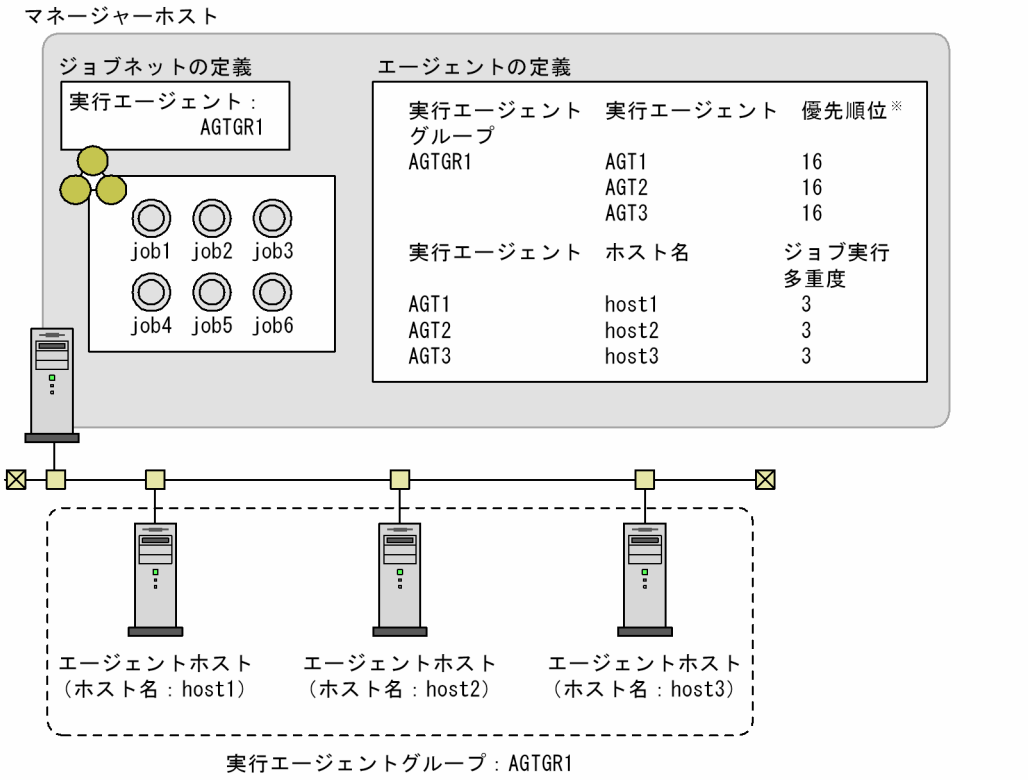
負荷分散のパターンごとに，どのように負荷分散処理されるかを説明します。この例では，job1～job6 の六つのジョブを定義したジョブネットの実行エージェントとして，実行エージェントグループ AGTGR1 を指定しているものとします。

(1)　均等に負荷を分散する場合

均等に負荷を分散させる場合は，実行エージェントグループにグルーピングしているそれぞれの実行エージェントの優先順位およびジョブ実行多重度を同じ値にします。

均等に負荷を分散する場合の例を次の図に示します。

図 2-62　均等に負荷を分散する場合の例



注※　実行エージェントの優先順位は，1～16の範囲で優先度の高いものから順に16，15，14･･･と指定します。デフォルトは16（いちばん高い）です。

例えば，実行エージェントグループ AGTGR1 にグルーピングした実行エージェント AGT1，AGT2，AGT3 の優先順位を「16」，ジョブ実行多重度を「3」と定義した場合，ジョブの実行順序は次のように決定されます。

1. job1 の実行エージェントが決定される。

   「ジョブ実行数 / ジョブ実行多重度」は次のとおりです。

   - 実行エージェント AGT1：「0 / 3」
   - 実行エージェント AGT2：「0 / 3」
   - 実行エージェント AGT3：「0 / 3」

   すべての実行エージェントのエージェント使用率※が同じであり，それぞれの優先順位も同じであるため，job1 の実行エージェントは，実行エージェントグループ AGTGR1 に定義されている AGT1～AGT3 のどれかに決定します。

   この例では，AGT1 に決定します。

2. job2 の実行エージェントが決定される。

   job1 の実行エージェントが AGT1 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「1 / 3」
   - 実行エージェント AGT2：「0 / 3」

- 実行エージェント AGT3：「0 / 3」

これによって，実行エージェント AGT2 と AGT3 のエージェント使用率※が同じになります。それぞれの優先順位は同じであるため，job2 の実行エージェントは，実行エージェントグループ AGTGR1 に定義されている AGT2 か AGT3 のどちらかに決定します。

この例では，AGT2 に決定します。

3. job3 の実行エージェントが決定される。

job2 の実行エージェントが AGT2 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

- 実行エージェント AGT1：「1 / 3」
- 実行エージェント AGT2：「1 / 3」
- 実行エージェント AGT3：「0 / 3」

これによって，実行エージェント AGT3 のエージェント使用率※がいちばん低くなります。そのため，job3 の実行エージェントは AGT3 に決定します。

4. job4 の実行エージェントが決定される。

job1～job3 の実行エージェントが決定し，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

- 実行エージェント AGT1：「1 / 3」
- 実行エージェント AGT2：「1 / 3」
- 実行エージェント AGT3：「1 / 3」

再びすべての実行エージェントのエージェント使用率※が同じとなったため，job4 の実行エージェントは，実行エージェントグループ AGTGR1 に定義されている AGT1～AGT3 のどれかに決定します。

この例では，AGT1 に決定します。

5. job5 の実行エージェントが決定される。

job4 の実行エージェントが AGT1 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

- 実行エージェント AGT1：「2 / 3」
- 実行エージェント AGT2：「1 / 3」
- 実行エージェント AGT3：「1 / 3」

実行エージェント AGT2 と AGT3 のエージェント使用率※が同じになったため，job5 の実行エージェントは，AGTGR1 に定義されている AGT2 か AGT3 のどちらかに決定します。

この例では，AGT2 に決定します。

6. job6 の実行エージェントが決定される。

job5 の実行エージェントが AGT2 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

- 実行エージェント AGT1：「2 / 3」

- 実行エージェント AGT2：「2 / 3」
- 実行エージェント AGT3：「1 / 3」

実行エージェント AGT3 のエージェント使用率※がいちばん低くなるため，job6 の実行エージェントは AGT3 に決定します。

注※

エージェント使用率とは，エージェントホストのジョブの実行多重度に対する実行中のジョブ数の割合です。

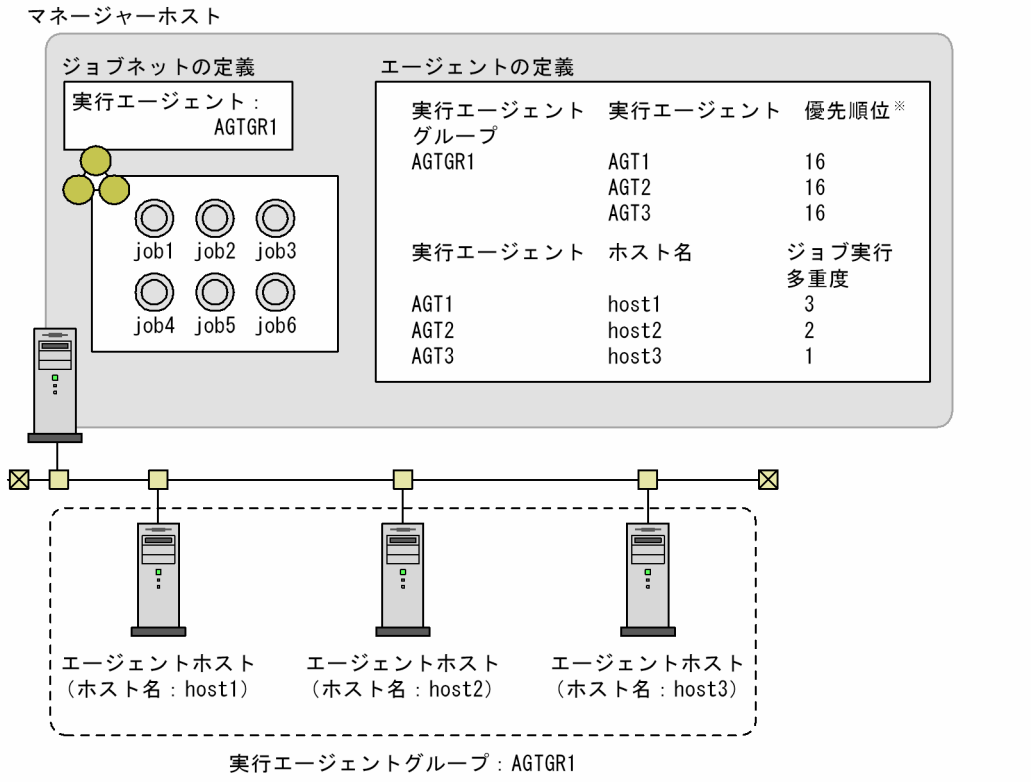
実行エージェントグループに複数接続した実行エージェントへのジョブの配信方法およびエージェント使用率の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド 5.4.9 実行ホスト（エージェントホスト）へのジョブ実行配信」を参照してください。

(2) 実行エージェントでのジョブ実行数の上限に差を付ける場合

それぞれの実行エージェントで実行するジョブ数の上限に差を付けて負荷を分散させる場合は，ジョブ実行多重度にそれぞれの値を設定します。優先順位には同じ値を設定します。

実行エージェントでのジョブ実行数の上限に差を付ける場合の例を次の図に示します。

図 2-63 実行エージェントでのジョブ実行数の上限に差を付ける場合の例



注※ 実行エージェントの優先順位は，1～16の範囲で優先度の高いものから順に16，15，14･･･と指定します。デフォルトは16(いちばん高い)です。

例えば，実行エージェントグループ AGTGR1 にグルーピングした実行エージェント AGT1，AGT2，AGT3 のジョブ実行多重度を，それぞれ「3」，「2」，「1」と定義した場合，ジョブの実行順序は次のようにして決定されます。

1. job1 の実行エージェントが決定される。

   「ジョブ実行数 / ジョブ実行多重度」は次のとおりです。

   - 実行エージェント AGT1：「0 / 3」
   - 実行エージェント AGT2：「0 / 2」
   - 実行エージェント AGT3：「0 / 1」

   それぞれの優先順位は同じであるため，job1 の実行エージェントは，実行エージェントグループ AGTGR1 に定義されている実行エージェントのうち，次にジョブを実行した場合のエージェント使用率※がいちばん低くなる実行エージェントに決定します。

   この例では，AGT1 に決定します。

2. job2 の実行エージェントが決定される。

   job1 の実行エージェントが AGT1 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「1 / 3」
   - 実行エージェント AGT2：「0 / 2」
   - 実行エージェント AGT3：「0 / 1」

   それぞれの優先順位は同じであるため，job2 の実行エージェントは，実行エージェントグループ AGTGR1 に定義されている実行エージェントのうち，次にジョブを実行した場合のエージェント使用率※がいちばん低くなる実行エージェントに決定します。

   この例では，AGT2 に決定します。

3. job3 の実行エージェントが決定される。

   job2 の実行エージェントが AGT2 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「1 / 3」
   - 実行エージェント AGT2：「1 / 2」
   - 実行エージェント AGT3：「0 / 1」

   これによって，実行エージェント AGT1 のエージェント使用率※がいちばん低くなります。そのため，job3 の実行エージェントは AGT1 に決定します。

4. job4 の実行エージェントが決定される。

   job3 の実行エージェントが AGT1 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「2 / 3」
   - 実行エージェント AGT2：「1 / 2」

- 実行エージェント AGT3：「0 / 1」

それぞれの優先順位は同じであり，実行エージェントが次にジョブを実行した場合のエージェント使用率※もすべての実行エージェントで同じであるため，job4 の実行エージェントは，実行エージェントグループ AGTGR1 に定義されている AGT1～AGT3 のどれかに決定します。

この例では，AGT1 に決定します。

5. job5 の実行エージェントが決定される。

job4 の実行エージェントが AGT1 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

- 実行エージェント AGT1：「3 / 3」
- 実行エージェント AGT2：「1 / 2」
- 実行エージェント AGT3：「0 / 1」

実行エージェント AGT2 と AGT3 の，次にジョブを実行した場合のエージェント使用率※は同じです。それぞれの優先順位も同じであるため，job5 の実行エージェントは，実行エージェントグループ AGTGR1 に定義されている AGT2 か AGT3 のどちらかに決定します。

この例では，AGT2 に決定します。

6. job6 の実行エージェントが決定される。

job5 の実行エージェントが AGT2 に決定したため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

- 実行エージェント AGT1：「3 / 3」
- 実行エージェント AGT2：「2 / 2」
- 実行エージェント AGT3：「0 / 1」

実行エージェント AGT1，AGT2 のどちらも，ジョブ実行数の上限に達しているため，job6 の実行エージェントは AGT3 に決定します。

注※

エージェント使用率とは，エージェントホストのジョブの実行多重度に対する実行中のジョブ数の割合です。

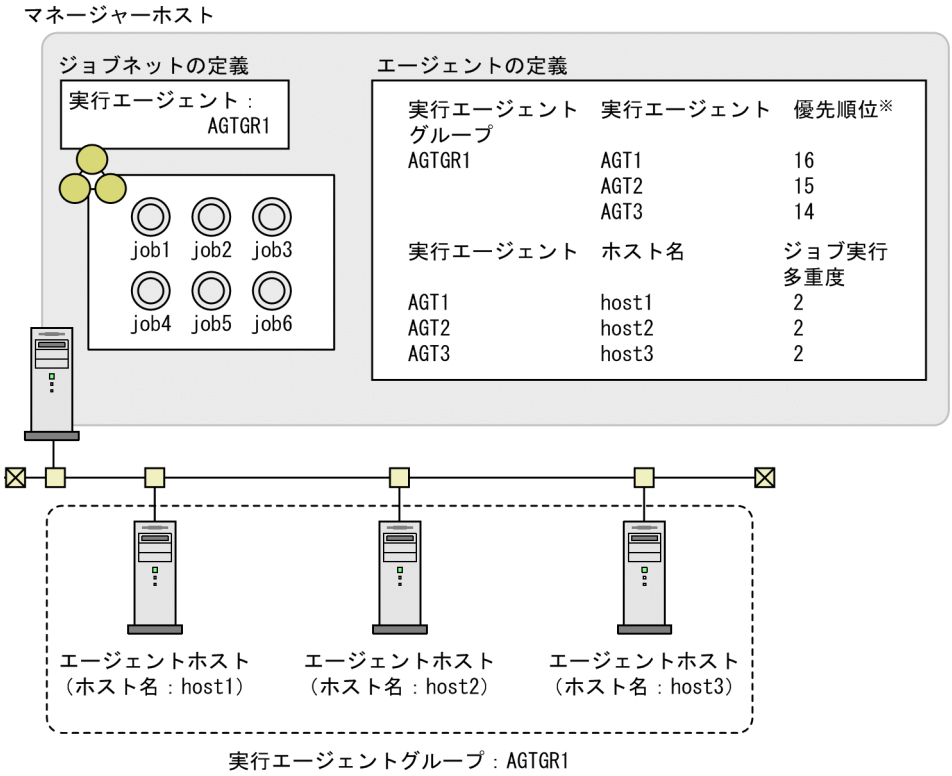
実行エージェントグループに複数接続した実行エージェントへのジョブの配信方法およびエージェント使用率の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド 5.4.9 実行ホスト（エージェントホスト）へのジョブ実行配信」を参照してください。

(3) ジョブ実行数が上限に達したら別の実行エージェントに分散して実行する場合

実行エージェントでのジョブ実行数が上限に達したら，別の実行エージェントに負荷を分散させる場合は，それぞれの実行エージェントに優先順位を設定します。また，必要に応じてジョブ実行多重度もそれぞれの実行エージェントに設定します。

実行エージェントでのジョブ実行数が上限に達したら，別の実行エージェントに分散して実行する場合の例を次の図に示します。

図 2-64　ジョブ実行数が上限に達したら別の実行エージェントに分散して実行する場合の例



注※　実行エージェントの優先順位は，1～16の範囲で優先度の高いものから順に16，15，14･･･と指定します。デフォルトは16（いちばん高い）です。

例えば，実行エージェントグループ AGTGR1 にグルーピングした実行エージェント AGT1，AGT2，AGT3 の優先順位をそれぞれ「16（いちばん高い）」，「15（2 番目に高い）」，「14（3 番目）」とし，ジョブ実行多重度をすべて「2」とした場合，ジョブの実行順序は次のようにして決定されます。

1. job1 が実行される。

   「ジョブ実行数 / ジョブ実行多重度」は次のとおりです。

   - 実行エージェント AGT1：「0 / 2」
   - 実行エージェント AGT2：「0 / 2」
   - 実行エージェント AGT3：「0 / 2」

   優先順位がいちばん高い実行エージェントは AGT1 であるため，job1 は AGT1 で実行されます。

2. job2 が実行される。

   job1 が AGT1 で実行されているため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「1 / 2」
   - 実行エージェント AGT2：「0 / 2」
   - 実行エージェント AGT3：「0 / 2」

実行エージェント AGT2 と AGT3 のエージェント使用率※が低くなりますが，優先順位がいちばん高いのは AGT1 のため，job2 も AGT1 で実行されます。

3. job3 が実行される。

   job2 が AGT1 で実行されているため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「2 / 2」
   - 実行エージェント AGT2：「0 / 2」
   - 実行エージェント AGT3：「0 / 2」

   優先順位がいちばん高いのは AGT1 ですが，ジョブ実行多重度の上限に達しているため，job3 は次に優先順位が高い AGT2 で実行されます。

4. job4 が実行される。

   job3 が AGT2 で実行されているため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「2 / 2」
   - 実行エージェント AGT2：「1 / 2」
   - 実行エージェント AGT3：「0 / 2」

   job3 と同様に，job4 は AGT2 で実行されます。

5. job5 が実行される。

   job4 が AGT2 で実行されているため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「2 / 2」
   - 実行エージェント AGT2：「2 / 2」
   - 実行エージェント AGT3：「0 / 2」

   実行エージェント AGT1，AGT2 とも，ジョブ実行多重度の上限に達しているため，job5 は AGT3 で実行されます。

6. job6 が実行される。

   job5 が AGT3 で実行されているため，「ジョブ実行数 / ジョブ実行多重度」は次のようになります。

   - 実行エージェント AGT1：「2 / 2」
   - 実行エージェント AGT2：「2 / 2」
   - 実行エージェント AGT3：「1 / 2」

   実行エージェント AGT1，AGT2 とも，ジョブ実行多重度の上限に達しているため，job6 は AGT3 で実行されます。

注※

エージェント使用率とは，エージェントホストのジョブの実行多重度に対する実行中のジョブ数の割合です。

実行エージェントグループに複数接続した実行エージェントへのジョブの配信方法およびエージェント使用率の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド 5.4.9 実行ホスト（エージェントホスト）へのジョブ実行配信」を参照してください。

2.3.2　業務の自動化を実現する JP1/AJS3 の機能

JP1/AJS3 で業務を自動化するに当たって，どのような機能を使用すればよいかについて紹介します。

JP1/AJS3 で実現したいことと，それを実現する JP1/AJS3 の機能の対応を次の表に示します。

表 2-19　業務の自動化を実現する JP1/AJS3 の機能

項番	JP1/AJS3 で実現したいこと	実現する JP1/AJS3 の機能	参照先
1	前の処理の結果に応じて次の処理を複数パターン用意したい。	判定ジョブ	「2.4.3 先行ジョブの結果でそのあとの処理を動的に変える（判定ジョブを使ったジョブネットの定義例）」
2	任意のホストからの JP1 イベントを受信したい。	JP1 イベント受信監視ジョブ	「2.4.4 事象の発生を契機に処理を実行する（イベントジョブを使ったジョブネットの定義例）」
3	ファイルの作成，削除，および更新を検知したい。	ファイル監視ジョブ	
4	ユーザーのログファイルや syslog に出力される特定のデータを監視したい。	ログファイル監視ジョブ，または JP1/Base のログファイルトラップ機能と JP1 イベント受信監視ジョブの組み合わせ	
5	Windows イベントログに出力されるメッセージを監視したい。	Windows イベントログ監視ジョブ，または JP1/Base のイベントログトラップ機能と JP1 イベント受信監視ジョブの組み合わせ	
6	ジョブとジョブの間隔を一定時間空けて実行したい。または，一定間隔でジョブネットを実行したい。	実行間隔制御ジョブ	
7	一定時間ごとにジョブネットを実行したい。	起動条件を設定した実行間隔制御ジョブ	
8	先行するイベントジョブの情報を使用して，後続の処理（ジョブ，ジョブネット）を実行したい。	イベントジョブの受信情報の引き継ぎ	
9	特定のフォルダに転送されてくる，ファイル名が不特定なファイルを使用して処理（ジョブ，ジョブネット）を実行したい。	• 起動条件 • ファイル監視ジョブ • イベントジョブの受信情報の引き継ぎ	
10	任意のホストに JP1 イベントを送信したい。	JP1 イベント送信ジョブ	「2.4.5 先行ジョブの終了や事象の発生を契機に JP1 イベントを送信する（JP1 イベント送

項番	JP1/AJS3 で実現したいこと	実現する JP1/AJS3 の機能	参照先
10	任意のホストに JP1 イベントを送信したい。	JP1 イベント送信ジョブ	信ジョブを使ったジョブネットの定義例)」
11	ジョブ実行時エラーが発生したときに，回復させるジョブ・ジョブネットを実行したい。	• リカバリージョブ • リカバリージョブネット	「2.4.6 ジョブが異常終了したときに特定の処理を実行する(リカバリーユニットを使ったジョブネットの定義例)」
12	異なるルートジョブネットの実行順序を制御したい。	ジョブネットコネクタ	「2.4.7 ルートジョブネットの実行順序を制御する（ジョブネットコネクタを使ったジョブネットの定義例)」
13	異なるジョブネットにあるユニット同士の実行順序を制御したい。	待ち合わせ条件	「2.4.8 異なるジョブネットにあるユニット同士の実行順序を制御する（待ち合わせ条件を使ったジョブネットの定義例)」
14	先行ジョブの標準出力ファイルから必要な情報を切り出して，後続の処理（ジョブ，ジョブネット）を実行したい。	引き継ぎ情報設定ジョブ	「2.4.9 動的に変わる情報を後続ユニットに引き継ぐ（引き継ぎ情報設定ジョブを使ったジョブネットの定義例)」
15	ジョブに定義した実行ファイルが異常終了した場合に，ジョブを自動的にリトライして業務を続行したい。	異常終了時の自動リトライ	「2.4.13 ジョブの異常終了時に自動でリトライする」
16	異常があったら次のジョブネット実行時に該当のジョブネットの実行を保留したい。	ジョブネットの保留方法の設定 • 前回異常時だけ保留 • 前回異常警告時だけ保留	マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12.3.8 ［詳細定義－［ジョブネット]］ダイアログボックス（ルートジョブネット用)」
17	運用の途中からジョブネットの定義内容を切り替えたい。	ジョブネットリリース	マニュアル「JP1/Automatic Job Management System 3 運用ガイド　7.3 実行登録中にジョブネットの定義を切り替える」
18	スケジュールが異なる複数のルートジョブネットをグループとしてまとめて一つのジョブネットのようにして業務を実行したい。	プランニンググループ	マニュアル「JP1/Automatic Job Management System 3 導入ガイド　10.1 プランニンググループを使用したルートジョブネットの計画切り替え」
19	特定の差出人からのメールや，特定の件名のメールを契機として処理（ジョブ，ジョブネット）を実行したい。	• メールシステム連携 • メール受信監視ジョブ	マニュアル「JP1/Automatic Job Management System 3 連携ガイド　2. メールシステムとの連携」
20	ジョブやジョブネットが終了した場合，またはシステムに異常が発生したときに，メールで通知して欲しい。	• メールシステム連携 • メール送信ジョブ	

項番	JP1/AJS3 で実現したいこと	実現する JP1/AJS3 の機能	参照先
21	先行のジョブやジョブネットが終了したら，自動的に電源をオフにしたい。または，再起動したい。	• JP1/Power Monitor 連携 • ローカル電源制御ジョブ	マニュアル「JP1/Automatic Job Management System 3 連携ガイド　3. JP1/Power Monitor との連携」
22	エージェントホストでジョブやジョブネットを実行する前に，そのエージェントホストの電源を自動的にオンにしたい。	• JP1/Power Monitor 連携 • リモート電源制御ジョブ	
23	JP1/Cm2/NNM や HP NNM を使用して，JP1/AJS3 の運用状態やジョブの実行状態を監視したい。	• JP1/Cm2/NNM または HP NNM 連携 • JP1/Cm2 状態通知ジョブ	マニュアル「JP1/Automatic Job Management System 3 連携ガイド　付録 A JP1/Cm2/NNM または HP NNM を使った監視」
24	毎回パラメーター値が変動する処理を含んだ業務を，ジョブネットの定義を変更しないで自動化したい。	マクロ変数名と引き継ぎ情報	マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.1.3 マクロ変数を使用した業務」

2.4　さまざまなジョブネット定義

ここでは，JP1/AJS3 の各ユニットの特性を踏まえながら，業務や処理の内容に応じて，ジョブを組み合わせてジョブネットを定義する例を紹介します。

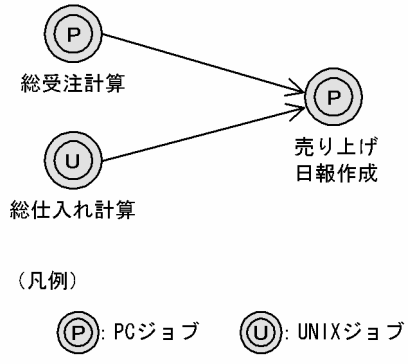
なお，ここでは GUI に基づいた表現で説明しています。ジョブネットを構成する各ユニット定義の設定項目や操作方法については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　5. ジョブネットの定義」を参照してください。

2.4.1　ファイルを指定して処理を実行する（標準ジョブを使ったジョブネットの定義例）

ファイルを実行して処理するようなジョブネットの定義には，標準ジョブを使用します。

「プログラム juchu.exe で総受注計算してファイルに出力した結果と，shiire.exe で総仕入れ計算してファイルに出力した結果から，nippou.exe で売り上げデータを算出して売り上げ日報を作成・印刷する。なお，総受注計算と売り上げ日報の作成は hostA（Windows）で，総仕入れデータ計算は hostB（UNIX）で実行する」というジョブネットを，標準ジョブを使って定義する例を次の図に示します。

図 2-65　標準ジョブを使ったジョブネットの定義例



「総受注計算」と「売り上げ日報作成」の実行先ホストは Windows であるため PC ジョブを，「総仕入れ計算」の実行先ホストは UNIX であるため UNIX ジョブを使用します。

「総受注計算」には，［実行エージェント］に「hostA」を，［実行ファイル名］に juchu.exe を指定します。

「総仕入れ計算」には，［実行エージェント］に「hostB」を，［実行ファイル名］に shiire.exe を指定します。

「売り上げ日報作成」には，［実行エージェント］に「hostA」を，［実行ファイル名］に nippou.exe を指定します。

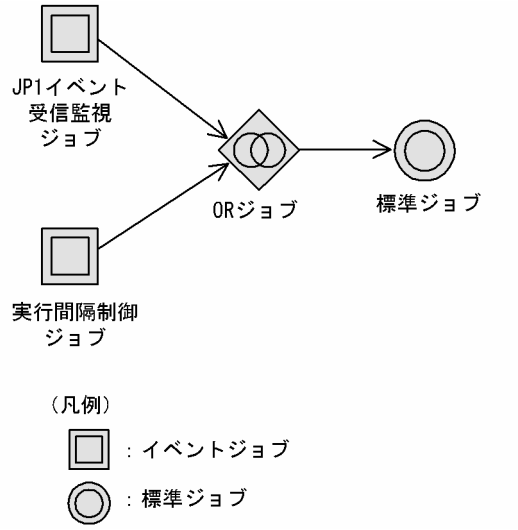
「総受注計算」と「総仕入れ計算」の実行順序に前後関係はありませんが，双方の結果から売り上げデータを算出するため，図のようなジョブフローになります。

2.4.2 複数の条件のうち一つが成立したら処理を実行する（OR ジョブを使ったジョブネットの定義例）

複数の条件のうち，どれか一つでも成立したら処理を実行させるようなジョブネットの定義には，OR ジョブを使用します。

「JP1 イベントの受信を 10 分間監視し，受信したら直ちに後続ジョブを実行させる。ただし，10 分間を経過した場合は JP1 イベントを受信しなくても，その時点で後続ジョブを実行させる」というジョブネットを，OR ジョブを使って定義する例を次の図に示します。

図 2-66　OR ジョブを使ったジョブネットの定義例



OR ジョブの先行ジョブとして定義できるのは，イベントジョブだけです。この場合，JP1 イベントの受信を監視するため JP1 イベント受信監視ジョブを定義します。また，10 分間という時間経過の監視には実行間隔制御ジョブを使用し，［待ち時間］を「10 分」と定義します。

OR ジョブの先行ジョブとして定義されたイベントジョブが監視している事象のうちどれか一つでも発生した場合，後続の OR ジョブが実行され，その他のイベントジョブは監視を打ち切って「計画未実行」状態となります。したがって，この場合は JP1 イベント受信監視ジョブで監視している JP1 イベントが受信されると，実行間隔制御ジョブは時間経過の監視を打ち切ります。イベントの受信によって後続ジョブが実行終了すると，そのジョブネットは実行終了となります。

なお，後続ジョブが終了判定する場合の OR ジョブの戻り値は，実行された先行ジョブであるイベントジョブの戻り値と同じ値になります。

注意事項

- イベントジョブを実行中止すると，イベントジョブは「未計画」状態となります。そのあと，実行中止したイベントジョブの先行ジョブが終了すると「未計画」状態のイベントジョブは「計画未実行」状態になりますが，その他のイベントジョブが実行中であっても「計画未実行」状態となってしまい，後続の OR ジョブが実行されるため注意が必要です。

なお，イベントジョブを強制終了したり，打ち切り時間に到達したりして，イベントジョブが「強制終了」状態，および「異常検出終了」状態に遷移した場合も，その他のイベントジョブは監視を打ち切り，「計画未実行」状態になります。しかし，この場合はORジョブの先行ジョブに異常終了したイベントジョブがあるため，ORジョブは実行されません。

- ORジョブの先行ジョブとして定義されたイベントジョブまたはその先行ユニットから再実行する場合で，ORジョブの先行ジョブに「計画未実行」状態で終了しているイベントジョブがあると，再実行した直後に，ORジョブが実行されるため注意が必要です。ORジョブの先行ジョブとして定義されたイベントジョブを再実行して再度事象を監視させたい場合は，ORジョブの直前のイベントジョブをすべて「異常検出終了」状態に状態変更したあと，［再実行方法］に［異常終了ジョブから］を指定してルートジョブネットを再実行するなどの対応をしてください。
- 先行ユニットが実行中などで，関連線に従って，実行できないイベントジョブがある場合，その他のイベントジョブが終了しても，ORジョブの後続ユニットは実行されません。先行ユニットが終了すると，イベントジョブが「計画未実行」状態となり，ORジョブの後続ユニットが実行されます。

2.4.3 先行ジョブの結果でそのあとの処理を動的に変える（判定ジョブを使ったジョブネットの定義例）

先行ジョブの終了結果，ファイルの有無，先行ジョブの引き継ぎ情報で，あとの処理内容が動的に変わるようなジョブネットの定義には，判定ジョブを使用します。

(1) 判定ジョブの使用例

JP1/AJS3には，前に実行された処理の終了結果によって次に実行する処理の振り分けができる判定ジョブというジョブがあります。この判定ジョブには，次に示す二つの判定パターンがあります。

- リターンコードによる判定
- ファイルの有無による判定

ここでは，この二つの判定パターンによる判定ジョブの使用例について説明します。

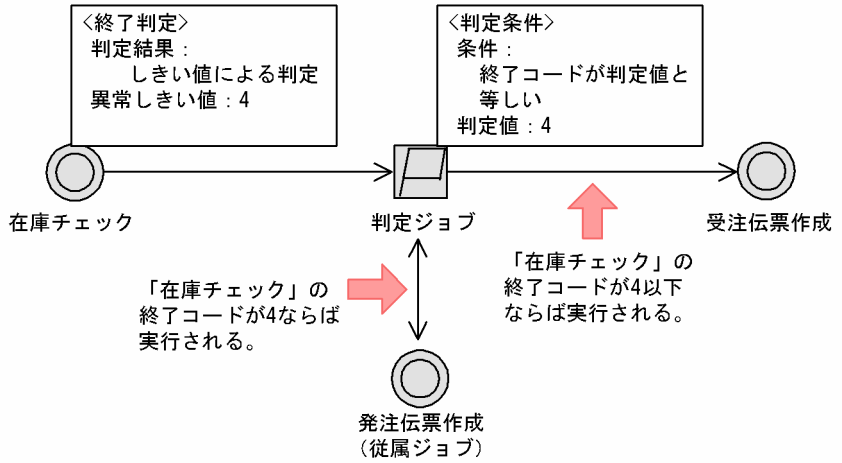
また，判定ジョブに関連づけられている従属ユニットが異常終了した場合の再実行方法についても説明します。

(a) リターンコードによる判定

リターンコードによる判定ジョブの使用例を次の図に示します。

「在庫チェックジョブの実行結果を判定して，在庫数が不足気味の場合は発注伝票を作成し，在庫数に問題がなければ受注伝票を作成する。」というジョブネットを定義する例で説明します。

図 2-67　リターンコードによる判定ジョブの使用例



「在庫チェック」は，在庫数が十分である場合は戻り値 3 以下，在庫数が不足気味の場合は戻り値 4，在庫数が不足している場合は戻り値 5 以上を返すものとします。

上記の図のように処理フローを作成することで，次に示す処理が実行されます。

- 在庫数が十分な場合

  リターンコードが「3」以下であるため，「発注伝票作成」をスキップして「受注伝票作成」を実行します。

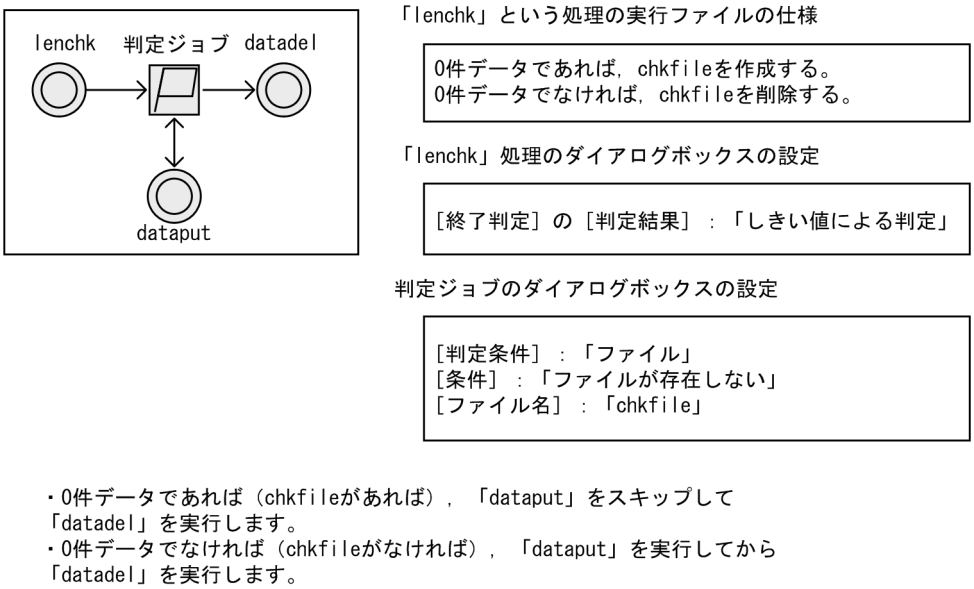
- 在庫数が十分ではない場合

  リターンコードが「4」であるため，「発注伝票作成」を実行してから「受注伝票作成」を実行します。

(b)　ファイルの有無による判定

ファイルの有無による判定ジョブの使用例を次の図に示します。

図 2-68　ファイルの有無による判定ジョブの使用例



上記の図のように処理フローを作成することで，「lenchk」という実行ファイル内で実行される，ほかの処理（命令）での異常も検知して，従属ジョブ「dataput」を実行できます。

なお，ファイルの有無による判定は，業務管理ホスト（JP1/AJS3 - Manager）がインストールされているホストのファイルを判定します。ただし，環境設定によっては他ホストのファイルの有無をチェックすることもできますが，通信障害などによって正しく判定できないおそれがあります。ファイルの有無によって判定する場合は，マネージャーホスト上のファイルを指定することを推奨します。詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.1.1(1)(c) 判定ジョブ」を参照してください。

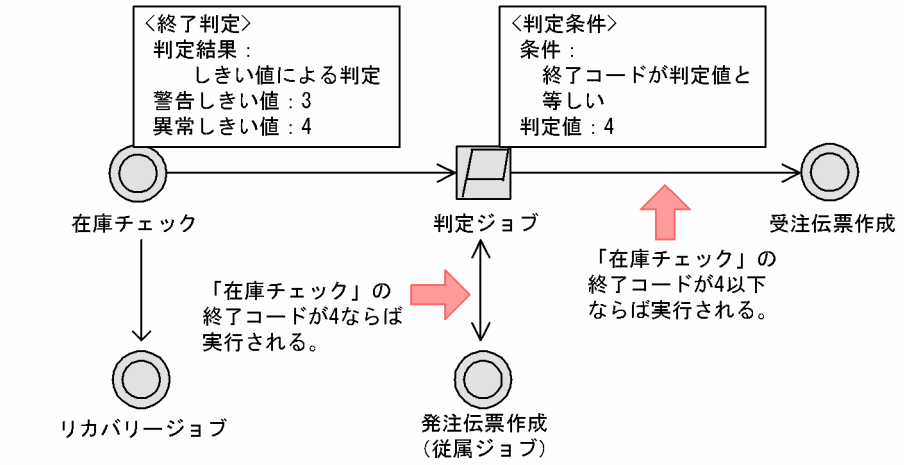
(c)　従属ユニットの再実行方法

従属ユニットの再実行方法は，通常のユニットの再実行方法とは異なります。再実行方法の注意事項は，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.5.11 実行が終了したジョブネットやジョブを再実行する」を参照してください。

(2)　判定ジョブを使ったジョブネットの定義例

「在庫チェックジョブの実行結果を判定して，在庫数がやや不足気味の場合は発注伝票を作成し，在庫数に問題がなければ受注伝票を作成する。在庫数が完全に不足している場合，在庫チェックジョブは異常終了し，リカバリージョブを実行する。」というジョブネットを，判定ジョブを使って定義する例を次の図に示します。なお，リカバリージョブについては，「2.4.6 ジョブが異常終了したときに特定の処理を実行する（リカバリーユニットを使ったジョブネットの定義例）」を参照してください。

図 2-69　判定ジョブを使ったジョブネットの定義例



この例では，先行ジョブの戻り値で判定しています。「在庫チェック」は，在庫数が十分の場合は戻り値 3 以下，在庫数がやや不足気味の場合は戻り値 4，在庫数が不足している場合は戻り値 5 以上を返すものとします。

まず，「在庫チェック」の［終了判定］を［しきい値による判定］と設定し，［警告しきい値］を「3」，［異常しきい値］を「4」とします。このように設定することで，「在庫チェック」は実行後に戻り値によって判定され，次の状態に遷移します。

- 戻り値が 3 以下：「正常終了」状態
- 戻り値が 4：「警告終了」状態
- 戻り値が 5 以上：「異常終了」状態

判定ジョブの判定条件には，［終了コードが判定値と等しい］と設定し，［判定値］を「4」とします。

これによって，「在庫チェック」を実行すると，以降のジョブは次のように実行されます。

- 「在庫チェック」の戻り値が 3 以下の場合
  判定ジョブでは，判定値 4 と「在庫チェック」の終了コードが合致しないため，従属ジョブの「発注伝票作成」は実行されないで「受注伝票作成」が実行されます。
- 「在庫チェック」の戻り値が 4 であった場合
  判定ジョブの判定条件に合致するため，従属ジョブである「発注伝票作成」を実行します。「発注伝票作成」が正常終了したら「受注伝票作成」を実行します。
- 「在庫チェック」の戻り値が 5 以上の場合
  リカバリージョブが実行されます。異常終了の場合，後続ジョブは実行されないため，リカバリージョブの実行後に「受注伝票作成」は実行されません。

なお，判定ジョブを連続して定義すると，先行ジョブの終了結果に対して，判定ジョブの数だけ処理を定義できます。判定ジョブを連続して定義した場合について，次に説明します。

(3) 判定ジョブを連続して定義した例

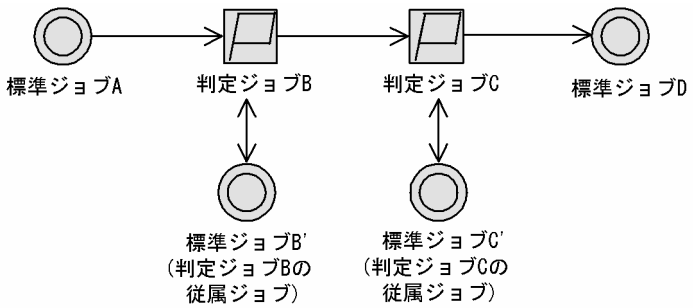
判定ジョブは連続して定義することもできます。

終了コード（戻り値）による判定条件を設定した判定ジョブを連続して定義した場合，2 番目以降の判定ジョブの判定対象となるのは，始めに定義した判定ジョブの先行ジョブに設定された終了コードです。従属ユニットの終了コードではありません。

一方，ファイルの有無を判定条件として設定した判定ジョブを連続して定義した場合，2 番目以降の判定ジョブの判定対象となるのは，始めに定義した判定ジョブが条件判定した時点でのファイル情報ではなく，自判定ジョブが条件判定する時点でのファイル情報です。

判定ジョブを連続して定義した場合の例を，次の図に示します。

図 2-70　判定ジョブを連続して定義した例



各ユニットは次のように定義されているものとします。

- 標準ジョブ A：[終了判定] は [しきい値による判定]，[警告しきい値] は「5」
- 判定ジョブ B：判定条件は [終了コードが判定値と等しい]，[判定値] は「0」
- 判定ジョブ C：判定条件は [終了コードが判定値と等しい]，[判定値] は「4」

このジョブネットを実行した場合，標準ジョブ A の戻り値によって，以降のジョブは次のように実行されます。

- 標準ジョブ A の戻り値が 0 の場合

  判定ジョブ B の判定条件に合致するため，判定ジョブ B の従属ジョブである標準ジョブ B'が実行されます。続いて判定ジョブ C で判定されます。判定ジョブ C の判定条件には合致しないため，判定ジョブ C の従属ジョブである標準ジョブ C'は実行されないで標準ジョブ D が実行されます。
- 標準ジョブ A の戻り値が 4 の場合

  判定ジョブ B の判定条件には合致しないため，判定ジョブ B の従属ジョブである標準ジョブ B'は実行されません。続いて判定ジョブ C で判定されます。判定ジョブ C の判定条件に合致するため，判定ジョブ C の従属ジョブである標準ジョブ C'が実行されたあと，標準ジョブ D が実行されます。
- 標準ジョブ A の戻り値が 0 と 4 以外の場合

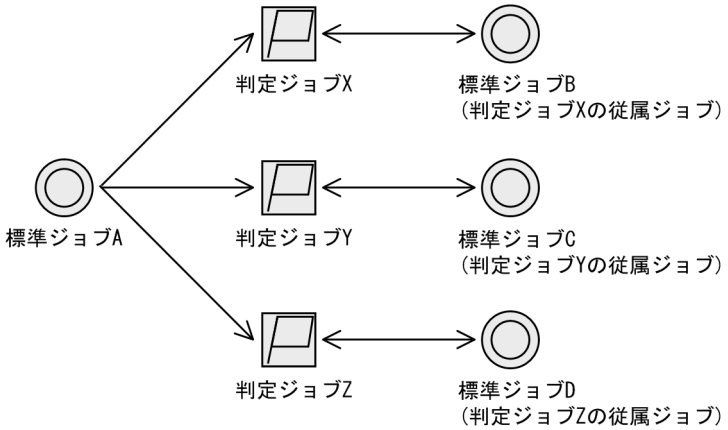
  判定ジョブ B の判定条件にも判定ジョブ C の判定条件にも合致しないため，従属ジョブである標準ジョブ B'，標準ジョブ C'は実行されません。この場合，標準ジョブ D だけが実行されます。

(4) 判定ジョブを並列に定義した例

判定ジョブは並列に定義することもできます。

判定ジョブを並列に定義した場合の例を，次の図に示します。

図 2-71 判定ジョブを並列に定義した例



各ユニットは次のように定義されているものとします。

標準ジョブ A

[終了判定]：[しきい値による判定]

［警告しきい値］：30

判定ジョブ X

判定条件：終了コードが判定値の範囲内

［下限値］：1 以上

［上限値］：9 以下

判定ジョブ Y

判定条件：終了コードが判定値の範囲内

［下限値］：10 以上

［上限値］：19 以下

判定ジョブ Z

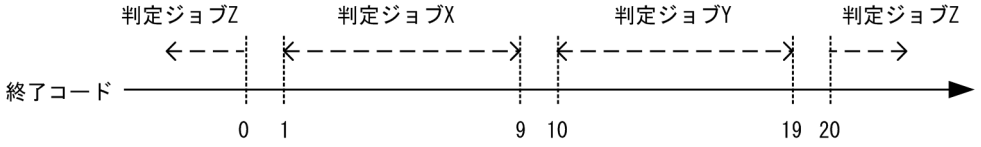
判定条件：終了コードが判定値の範囲外

［下限値］：1 以上

［上限値］：19 以下

それぞれの判定ジョブの判定値の範囲は，次のようになります。

図 2–72　判定値の範囲



このジョブネットを実行した場合，標準ジョブ A の終了コードによって，以降のジョブは次のように実行されます。

- 標準ジョブ A の終了コードが 1 以上 9 以下の場合

  判定ジョブ X の判定条件に合致するため，判定ジョブ X の従属ジョブである標準ジョブ B が実行されます。

- 標準ジョブ A の終了コードが 10 以上 19 以下の場合

  判定ジョブ Y の判定条件に合致するため，判定ジョブ Y の従属ジョブである標準ジョブ C が実行されます。

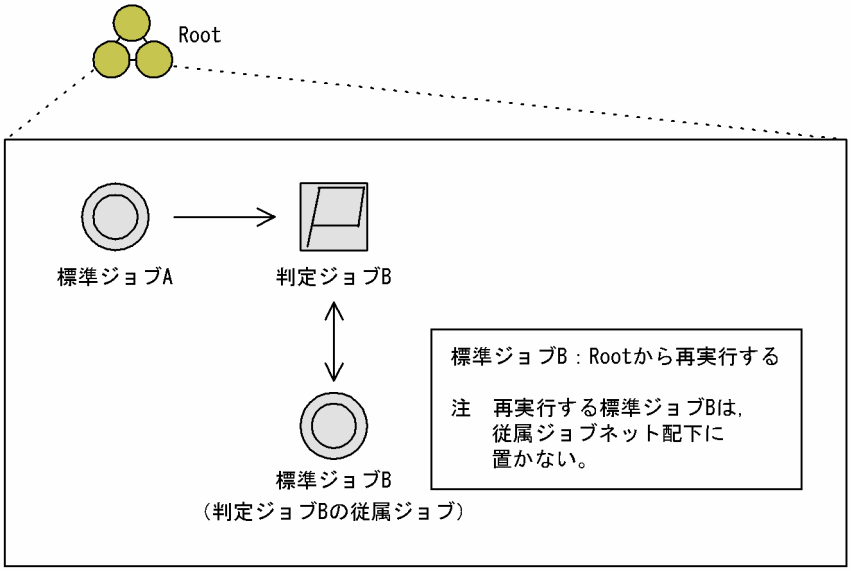
- 標準ジョブ A の終了コードが 1 より小さい，または 19 より大きい場合

  判定ジョブ Z の判定条件に合致するため，判定ジョブ Z の従属ジョブである標準ジョブ D が実行されます。

(5)　注意事項

判定ジョブの結果でルートジョブネットや先行ユニットを再実行する場合は，次の図に示すように再実行するジョブを従属ジョブとして定義してください。

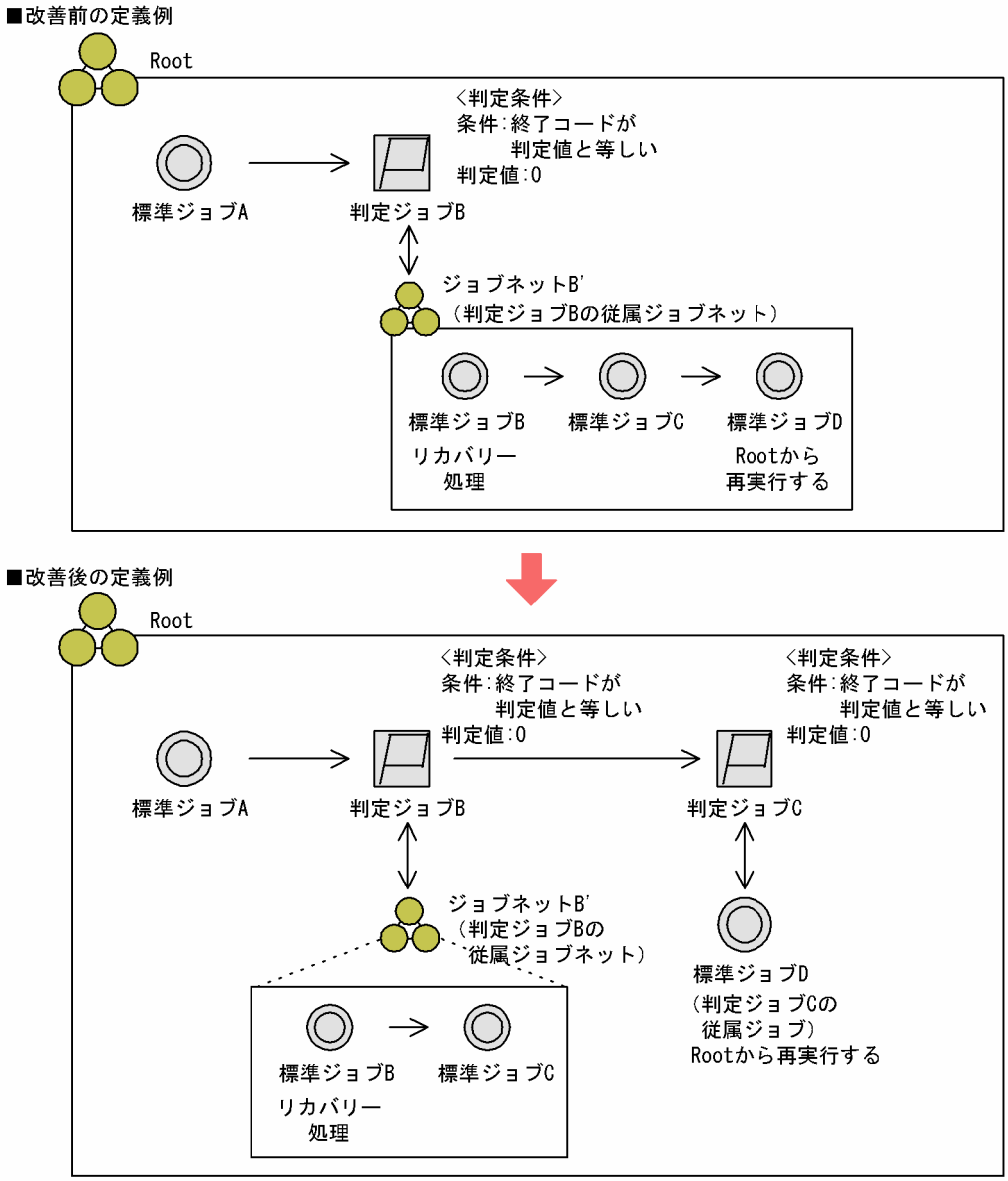
図 2-73　判定ジョブの結果でルートジョブネットを再実行する例



再実行するジョブを従属ジョブネット配下に定義した場合，再実行されたジョブが終了するタイミングによっては，従属ジョブネットが再び実行されます。

このため，再実行する前にリカバリー処理などをする場合は，次の図の「■改善前の定義例」のように「リカバリー処理」と「再実行」を一つの従属ジョブネットの中に定義するのではなく，「■改善後の定義例」のように「リカバリー処理」を行う従属ジョブネットと「再実行」を行う従属ジョブとに分けて定義してください。

図 2-74　再実行前にリカバリー処理をする場合の定義例

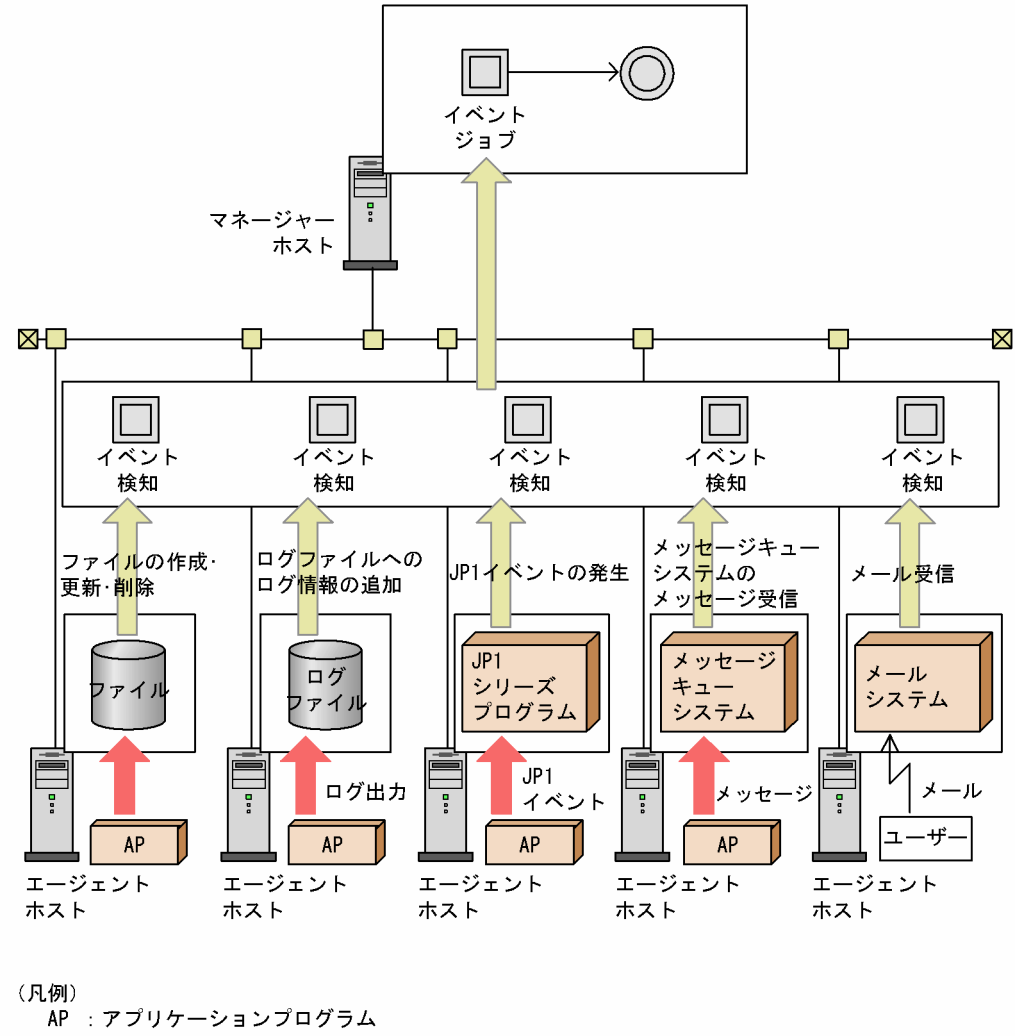


2.4.4　事象の発生を契機に処理を実行する（イベントジョブを使ったジョブネットの定義例）

イベントの受信やファイルの更新などのような事象の発生を契機に処理を実行させる場合は，イベントジョブを使ってジョブネットを定義します。

イベントジョブは，ネットワーク上のマネージャーホストおよびエージェントホストで発生した事象すべてを監視対象にできます。イベントジョブの監視対象を次に示します。

図 2-75　イベントジョブの監視対象



イベントジョブで監視する事象が検知されるタイミング

イベントジョブを使用する場合，イベントジョブを実行した時間と実際にイベントの受信監視ができる状態になる時間とにタイムラグが発生することがあります。このタイムラグの間に発生したイベントは検知できないため，イベントジョブを使ってジョブネットを定義する際は，このタイムラグを十分に考慮する必要があります。

なお，JP1 イベント受信監視ジョブの場合は，この問題の解決策として実行前のイベント検索機能を使用する方法があります。詳細については，「(1) JP1 イベントの受信を契機に処理を実行する（JP1 イベント受信監視ジョブ）」を参照してください。

不特定のタイミングで複数回発生するような事象をジョブネット実行の契機とする場合

不特定のタイミングで複数回発生するような事象をジョブネット実行の契機とする場合は，ジョブネットに起動条件を設定することを推奨します。起動条件の設定については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.4 起動条件の定義」を参照してください。

なお，監視する事象が計画的に発生することがわかっているような場合は，起動条件を使わないでイベントジョブをジョブネットの先頭に置くようにしてください。

イベントジョブに打ち切り時間を設定する場合

イベントジョブに打ち切り時間を指定した場合，打ち切り時間は実行先のエージェントホストで計られます。このため，監視中に実行先のエージェントホストが電源ダウンなどで再起動したあともイベントジョブによる事象の監視が継続された場合，打ち切り時間は実行先のエージェントホストが再起動した時刻から計り直されます。エージェントホストの状態に関係なく，ジョブ実行開始時刻からの絶対時刻で監視を打ち切る場合は，ジョブネットの起動条件にイベントジョブを定義し，起動条件の有効範囲を絶対時刻で指定してください。起動条件の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.4 起動条件の定義」を参照してください。

また，打ち切り時間を設定した場合は，打ち切り時間経過後のイベントジョブをどのような状態として扱うかを指定できます。「強制終了」，「正常終了」，「警告検出終了」，または「異常検出終了」から選べます。デフォルトは「強制終了」です。

これによって，イベントジョブの打ち切り時間経過後にジョブネットを中断または続行できます。

イベント情報の引き継ぎについて

イベントジョブで受信したイベント情報を変数（マクロ変数）として定義しておき，後続ユニットに引き継ぐことができます。イベント情報の引き継ぎについては，「(6) イベントジョブの受信情報の引き継ぎ」を参照してください。

イベントジョブを使ったジョブネットの定義例を紹介します。イベントジョブを使ったジョブネット定義の詳細については，「7.6 イベントジョブ使用時の注意事項」もあわせて参照してください。

なお，メール受信監視ジョブは，JP1/AJS3 とプログラムとの連携が必要です。詳細については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド」を参照してください。

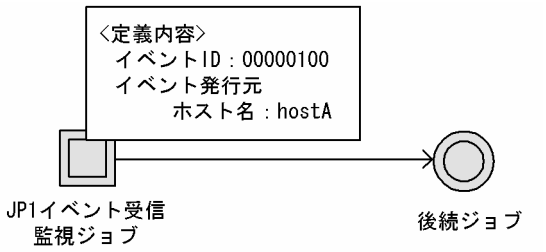
(1)　JP1 イベントの受信を契機に処理を実行する（JP1 イベント受信監視ジョブ）

JP1 イベントとは，JP1/Base で管理される，JP1 シリーズプログラムで事象が発生するたびに発行されるイベントをいいます。JP1 イベントは，エラー，警告，通知などの重大度やメッセージなどの情報を持っているため，エラーや警告イベントの受信や，特定のメッセージの受信を契機に後続ジョブやジョブネットを実行させることができます。また，JP1/Base のイベント変換機能を使って，JP1 シリーズ以外のアプリケーションの終了を契機に後続ジョブやジョブネットを実行させることもできます。

JP1 イベント受信監視ジョブを実行する際は，実行前に JP1/Base イベントサービスを起動しておく必要があります。JP1/Base イベントサービスの API 設定は「keep-alive」にしておいてください。起動されていない場合，JP1 イベント受信監視ジョブはサービスが起動されるまで「実行中」状態のままとなります。

「hostA から JP1 イベントを受信したら，後続ジョブを実行する（ジョブネットを定義するホストを hostB とする）」というジョブネットを，JP1 イベント受信監視ジョブを使って定義する例を，次の図に示します。

図 2-76　JP1 イベント受信監視ジョブを使ったジョブネットの定義例



JP1 イベント受信監視ジョブは，他ホストのジョブネットと連携できます。

hostA でジョブネットに JP1 イベント送信ジョブ（イベント ID：100）を定義します。hostA で発行した JP1 イベントを監視対象とするために，JP1 イベント受信監視ジョブの［イベント発行元ホスト名］に「hostA」を設定します。また，hostA から JP1 イベントを受信した場合に後続ジョブを実行させるために，［イベント ID］には「100」を設定します。

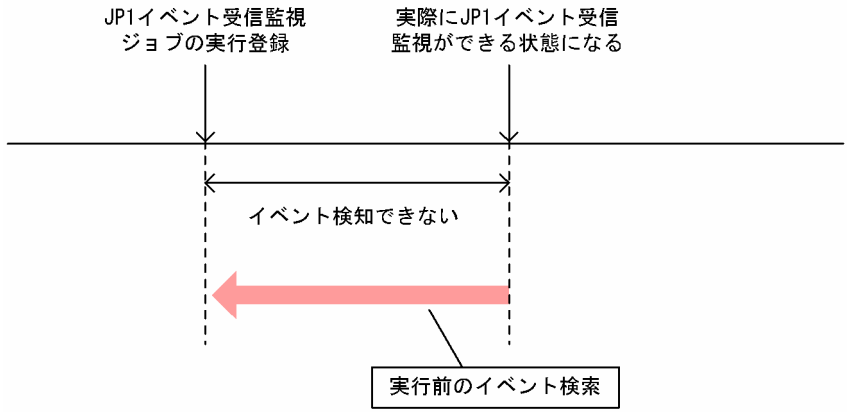
これによって，hostA でジョブネット中の JP1 イベント送信ジョブが実行されると，JP1 イベントの監視が終了して，後続ジョブが実行されます。

なお，JP1 イベント受信監視ジョブを定義した場合，JP1 イベント受信監視ジョブが実行されてから実際に JP1 イベント受信監視ができる状態になるまでにタイムラグが発生し，この間に発生した JP1 イベントは検知できません。この問題を解決するための機能として，実行前のイベント検索があります。

- **実行前のイベント検索**

  JP1 イベント受信監視ジョブが実際に JP1 イベントの受信を監視できる状態になる時点より前に発生したイベントを検索し，該当したイベントがあった場合にそれをイベント発生と見なします。

  図 2-77　実行前のイベント検索



(2)　ファイルの更新を契機に処理を実行する（ファイル監視ジョブ）

ファイル更新や新規ファイルの作成などを契機にジョブを実行するジョブネットの定義には，ファイル監視ジョブを使用します。

ファイル監視ジョブで指定できる監視条件を次に示します。

監視条件

監視対象のファイルがどのような状態になったときに条件成立とするのかを指定します。監視条件は，次の四つから設定できます。

- 指定したファイル名のファイルが作成されたとき
- 指定したファイル名のファイルが削除されたとき
- 指定したファイル名のファイルのサイズが変更されたとき
- 指定したファイル名のファイルの最終書き込み時刻が更新されたとき

補足事項

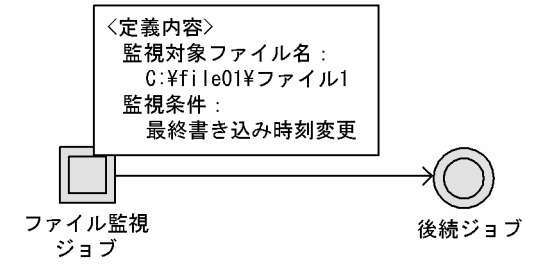
- ファイルの作成を監視条件とした場合に，ファイル監視ジョブが監視を開始した時点で指定した名称のファイルがすでにあったとき，ファイル監視ジョブの監視条件を成立させるかどうかを指定できます。
- 監視条件は複数指定できます。例えば，ファイルが削除または更新されたら後続ジョブを実行するように定義する場合は，ファイルの削除と最終書き込み時刻変更を指定できます。ただし，ファイルのサイズ変更と最終書き込み時刻変更は同時に指定できません。
- 監視条件が成立する場合および成立しない場合の詳細については，「7.6.2(1) ファイル監視ジョブで監視する事象」を参照してください。
- ファイルの作成，サイズ変更，および最終書き込み時刻変更の監視条件に合致した場合，監視対象のファイルにファイル監視ジョブ以外のプロセスがアクセスしていないかチェックします。これを**クローズチェック**といいます。

  監視条件に合致しても，ファイル監視ジョブ以外のプロセスが監視対象のファイルにアクセスしていた場合は条件成立となりません。その場合，次回の監視間隔に繰り越して監視し，そこでファイル監視ジョブ以外のプロセスのアクセスがなければ条件成立となります。

  このクローズチェックをすることで，ファイルコピー中など転送中の監視対象のファイルが条件成立となることを防ぐことができます。
- 監視対象は，ネットワークを介したファイルを指定できます。

「あるファイル（ファイル名：ファイル 1）の書き込み時間を監視し，ファイルが更新されたときに後続ジョブを実行する」というジョブネットを，ファイル監視ジョブを使って定義する例を次の図に示します。

図 2-78　ファイル監視ジョブを使ったジョブネットの定義例



ファイル監視ジョブに監視対象ファイル名と監視条件を設定します。［監視対象ファイル名］にはファイル 1 のパスを，［監視条件］には［最終書き込み時刻変更］を指定します。

これによって，監視対象のファイルがクローズ（アクセスしているアプリケーションがなくなった）し，最終書き込み時刻が更新された時点でファイル監視ジョブが終了となり，条件成立となります。

ネットワークを介したファイルを監視する場合は，環境設定パラメーターNetworkFilewatch に「Y」を設定します。ネットワークを介したファイルを監視するための設定については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.3.19 ネットワークを介したファイルをファイル監視ジョブで監視するための設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.3.18 ネットワークを介したファイルをファイル監視ジョブで監視するための設定」（UNIX の場合）を参照してください。

また，監視対象に指定するネットワークを介したファイルに，次のとおりアクセス権限を設定してください。

Windows の場合

監視対象のファイルの変更権限を持つユーザーアカウントを，JP1/AJS3 サービスを起動するユーザーアカウントに設定してください。JP1/AJS3 サービスを起動するユーザーアカウントを変更する場合の詳細については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編）　4.2.3 JP1/AJS3 のサービスの設定を変更する必要がある場合（Windows 限定）」を参照してください。

UNIX の場合

監視対象に指定したネットワークファイルに対して，クライアント側から参照権限でアクセスできるように設定してください。

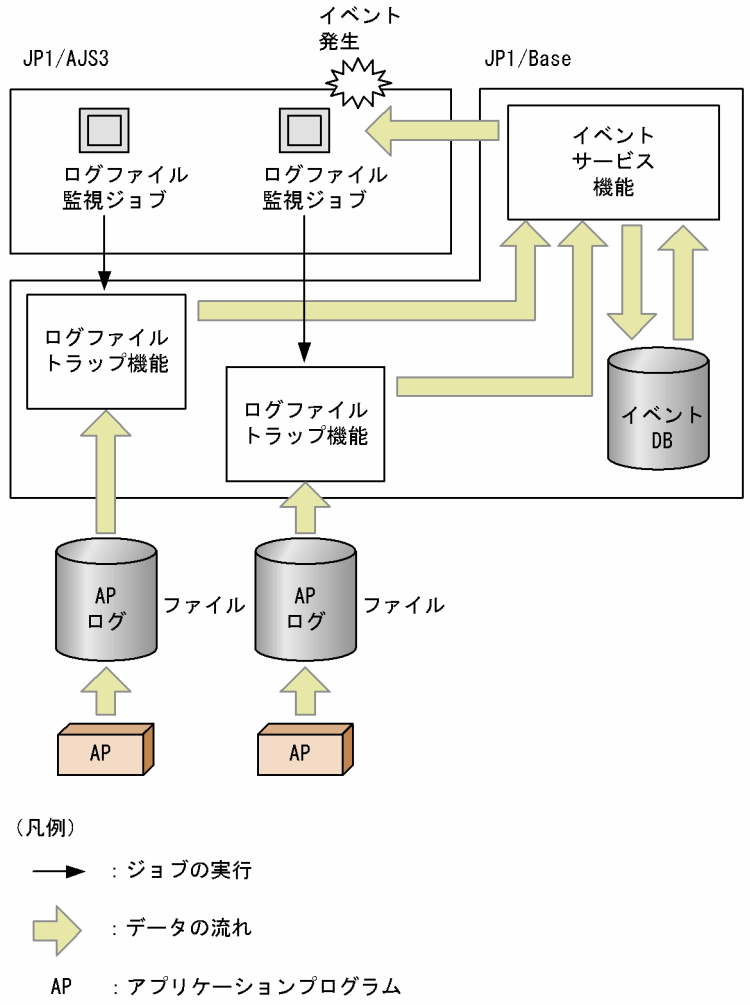
なお，Windows で NFS サーバ上のファイルを監視する場合，または UNIX でネットワークを介したファイルを監視する場合は，ファイル更新が終わった時点で処理の完了を示す別ファイルを作成し，その別ファイルを監視対象とする運用を推奨します。更新するファイルを，直接ファイル監視ジョブで監視しないでください。Windows で NFS サーバ上のファイルを監視する場合，または UNIX でネットワークを介したファイルを監視する場合の注意事項については，「7.6.2 ファイル監視ジョブの注意事項」を参照してください。

(3)　ログファイルの更新を契機に処理を実行する（ログファイル監視ジョブ）

ログファイル監視ジョブは，JP1/Base のログファイルトラップ機能を使って実行されます。JP1/Base のログファイルトラップ機能とは，アプリケーションプログラムが出力するログファイルのレコードを JP1 イベントに変換して，イベント DB に登録するものです。ログファイル監視ジョブを定義することで，ログファイルの更新を契機にジョブやジョブネットを実行させることができます。JP1/Base のログファイルトラップ機能については，マニュアル「JP1/Base 運用ガイド」を参照してください。また，ログファイル監視ジョブの注意事項については，「7.6.4 ログファイル監視ジョブの注意事項」を参照してください。

ログファイル監視ジョブの動作の概要を次の図に示します。

図 2-79　ログファイル監視ジョブの動作の概要



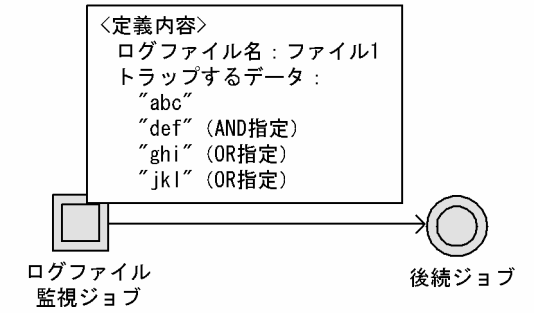
ログファイル監視ジョブは，監視対象に指定されたホストにあるログファイルに特定のデータが出力されたかを監視します。ログファイル監視ジョブには，監視するログファイル名と監視する文字列を指定します。文字列の指定には正規表現を使うこともできます。正規表現については，マニュアル「JP1/Base 運用ガイド」（Windows の場合）または UNIX のドキュメント（UNIX の場合）を参照してください。

監視間隔を 1～86,400（単位：秒）で指定できます。監視対象として指定できるログファイルは，テキスト形式のものになります。また，複数のログファイル（最大 8 件）を監視することもできます。

ログファイル監視ジョブを使った条件の例を次に示します。

「ログファイル（ファイル名：ファイル 1）に特定の文字列が含まれたログデータが書き込まれたら，後続ジョブを実行する。」というジョブネットを，ログファイル監視ジョブを使って定義する例を次の図に示します。

図 2-80　ログファイル監視ジョブを使ったジョブネットの定義例



ログファイル監視ジョブにログファイル名と監視する文字列を設定します。［ログファイル名］にはファイル 1 のパスを，［トラップするデータ］にはこの図のように特定の文字列を指定します。

これによって，ファイル 1 の文字列に，「"abc"と"def"」，「"ghi"」，「"jkl"」の三つのうちどれか一つでも書き込まれたら，ログファイルからログデータを取り出した時点でログファイル監視ジョブは終了となり，条件成立となって後続ジョブが実行されます。

また，複数のログファイルを監視し，そのうちのどれか一つに特定の文字列が含まれたログデータが書き込まれたら，後続ジョブを実行させることもできます。

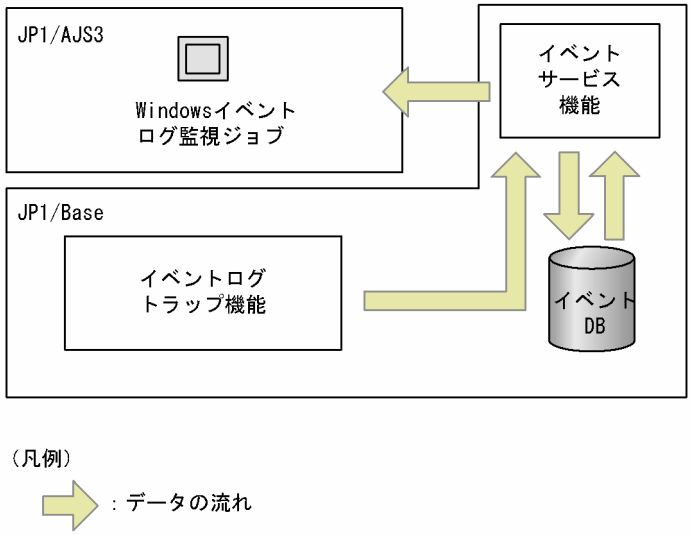
(4)　Windows イベントログの受信を契機に処理を実行する（Windows イベントログ監視ジョブ）

Windows イベントログ監視ジョブは，JP1/Base のイベントログトラップ機能を使って実行されます。JP1/Base のイベントログトラップ機能とは，Windows イベントログのレコードを JP1 イベントに変換してイベント DB に登録するものです。Windows イベントログ監視ジョブを定義することで，Windows イベントログの受信を契機にジョブやジョブネットを実行できます。

JP1/Base のイベントログトラップ機能，およびイベントログトラップ機能の動作定義については，マニュアル「JP1/Base 運用ガイド」を参照してください。また，Windows イベントログ監視ジョブの注意事項については，「7.6.5 Windows イベントログ監視ジョブの注意事項」を参照してください。

Windows イベントログ監視ジョブの動作の概要を次の図に示します。

図 2-81　Windows イベントログ監視ジョブの動作の概要



Windows イベントログ監視ジョブで監視できる内容について，次に示します。

ログ種別

指定できるログの種別を次に示します。

- アプリケーション
- セキュリティ
- システム
- DNS Server
- Directory Service
- ファイル複製サービス
- 任意のログ種別※

注※

任意のログ種別として指定できるログ種別の確認方法は，マニュアル「JP1/Base 運用ガイド」を参照してください。

イベントの種別

指定できるイベントの種別を次に示します。

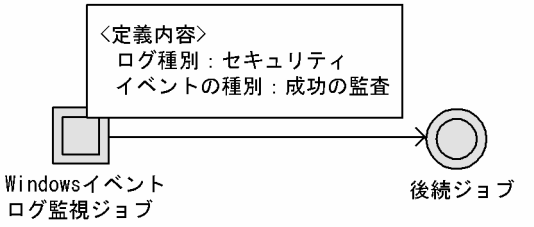
- 詳細
- 情報
- 警告
- エラー
- 重大
- 成功の監査

- 失敗の監査

Windows イベントログ監視ジョブを使ったジョブネットの定義例を次に示します。

「セキュリティシステムでの認証が成功したことを通知する Windows イベントが Windows のイベントログに出力されたら，後続ジョブを実行する」というジョブネットを，Windows イベントログ監視ジョブを使って定義する例を次の図に示します。

図 2-82　Windows イベントログ監視ジョブを使ったジョブネットの定義例



まず，Windows イベントログ監視ジョブの実行前に，JP1/Base の動作定義ファイルにイベントログトラップ機能の動作を設定します。この例の場合，ログ種別は「Security」，属性名（type の場合）は「Audit_success」です。

Windows イベントログ監視ジョブには，監視するログ種別とイベントの種類を指定します。［ログ種別］には［セキュリティ］，［イベントの種別］には［成功の監査］を設定します。

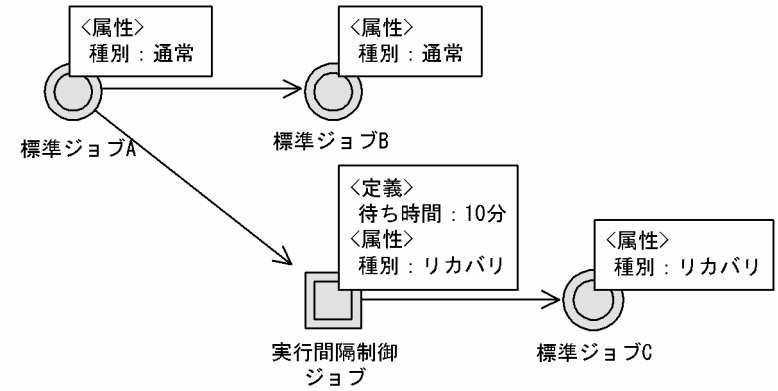
これによって，指定した Windows イベントが出力されて Windows のイベントログのデータを取り出した時点で Windows イベントログ監視ジョブは終了となり，条件成立となって後続ジョブが実行されます。

(5)　時間の経過を監視して処理を実行する（実行間隔制御ジョブ）

何分間待ってからジョブを実行する，などのようなジョブネットの定義には，実行間隔制御ジョブを使用します。

「先行ジョブが異常終了したら，10 分後にリカバリージョブを実行する」というジョブネットを，実行間隔制御ジョブを使って定義する例を次の図に示します。

図 2-83　実行間隔制御ジョブを使ったジョブネットの定義例



まず，先行ジョブが異常終了してから 10 分後に次の処理を実行させるため，時間の経過を監視する実行間隔制御ジョブを使用します。実行間隔制御ジョブには，［待ち時間］を「10 分」と定義します。また，先行ジョブが異常終了したときに実行させるため，［種別］を［リカバリ］にします。実行間隔制御ジョブの後続ジョブである標準ジョブ C も同様に［種別］を［リカバリ］にします。

これによって，標準ジョブ A が異常終了した場合，リカバリージョブである実行間隔制御ジョブが 10 分間の時間経過を監視し，10 分経過するとその後続ジョブである標準ジョブ C が実行されます。標準ジョブ A が正常終了した場合は，標準ジョブ B が実行されます。

なお，実行間隔制御ジョブで指定する［待ち時間］は，ジョブが実行される所要時間ではなく，実行間隔制御プロセスでの待ち時間を意味します。通信の状態などによって，指定した待ち時間と誤差が生じる場合があります。

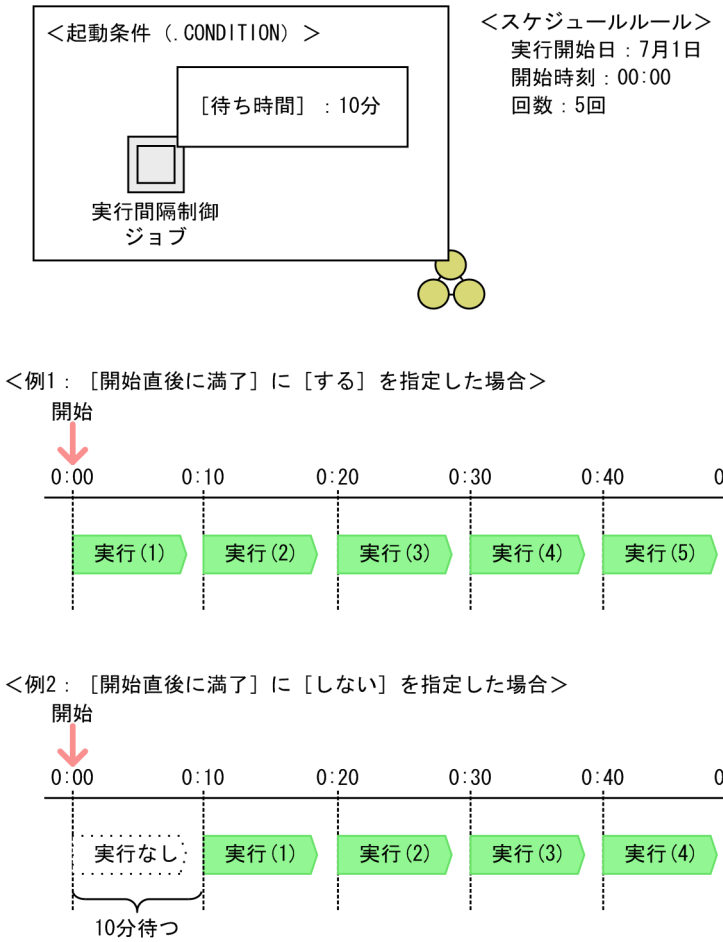
(a) 実行間隔制御ジョブを起動条件に定義する場合

実行間隔制御ジョブを起動条件に定義する場合，実行間隔制御ジョブの実行開始直後に起動条件を成立させるかどうかを選択できます。

実行開始直後に起動条件を成立させる場合は，実行間隔制御ジョブの［開始直後に満了］に［する］を指定します。実行開始直後に起動条件を成立させないで，［待ち時間］に指定した時間を待つ場合は，実行間隔制御ジョブの［開始直後に満了］に［しない］を指定します。

スケジュールルールで「7 月 1 日の 0:00 から 5 回処理を実行する」と設定した場合を例にして，それぞれの動作を説明します。

図 2-84　起動条件に実行間隔制御ジョブを定義する例



例 1 では，［開始直後に満了］に［する］を指定しています。その場合，7 月 1 日 0:00 に起動条件の監視が始まり，すぐに起動条件が成立してジョブネットの最初の実行が開始されます。その後，待ち時間に指定した 10 分ごとにジョブネットが実行されます。

例 2 では，［開始直後に満了］に［しない］を指定しています。その場合，7 月 1 日 0:00 に起動条件の監視が始まり，待ち時間に指定した 10 分が経過した 0:10 にジョブネットの最初の実行が開始されます。その後も，10 分ごとにジョブネットが開始されます。

［開始直後に満了］に［しない］を指定する場合は，最初にジョブネットを実行する時刻から待ち時間の分だけ早い時刻を開始時刻に指定する必要があります。

注意事項

- ［開始直後に満了］は，実行間隔制御ジョブの実行先の JP1/AJS3 - Manager または JP1/AJS3 - Agent のバージョンが 10-00 以降，および JP1/AJS3 - View のバージョンが 10-00 以降の場合に指定できます。JP1/AJS3 - Manager がバージョン 10-00 以降で，実行先の JP1/AJS3 - Agent のバージョンが 10-00 より前の場合は，実行エージェントとして，JP1/AJS3 - Manager 上のエージェントであるデフォルト実行エージェント「@SYSTEM」を指定してください。実行先の JP1/AJS3 - Manager または JP1/AJS3 - Agent のバージョンが 10-00 より前の場合，実行間隔制御ジョブは「異常検出終了」状態になります。

- JP1/AJS3 サービスまたはスケジューラーサービスを強制終了した場合，［開始直後に満了］に［する］を指定していても，次に起動したときには，待ち時間に指定した時間が経過してからジョブネットが実行されます。

(6) イベントジョブの受信情報の引き継ぎ

イベントジョブで受信したイベント情報を変数（マクロ変数）として定義しておき，後続ユニットに引き継ぐことができます。後続ユニットに引き継ぐ情報を**引き継ぎ情報**といいます。

後続ユニットでイベント情報を引き継ぐには，イベントジョブ側に情報を引き継ぐためのマクロ変数を定義しておき，後続ユニット側でマクロ変数名を指定します。

イベントジョブ側に定義しておくマクロ変数は，次のように設定します。

マクロ変数の定義
　「**マクロ変数名:引き継ぎ情報名**」

マクロ変数名は，「?AJS2xxxxx?」という形式の 64 バイト以内の文字列で指定します。xxxxx の部分に指定できる文字列は，A～Z（大文字の英字），0～9（アラビア数字），「.（ピリオド）」です。また，「**引き継ぎ情報名**」に設定できる情報は，イベントジョブで引き継ぐ情報として用意されています。詳細については，「付録 B イベントジョブで引き継ぐ情報」を参照してください。

マクロ変数はイベントジョブの後続ユニットの実行先のエージェントホスト上で，コマンドラインに引き継いだ情報を展開した形で実行されます。受信情報を引き継ぐ場合には，その内容を意識して，実行時にコマンドの引数として扱える形の情報を引き継ぐ場合だけ定義してください。

マクロ変数の詳細については，「2.2.6 マクロ変数の使用を検討する」を参照してください。

ここでは，イベントジョブで受信した情報を後続ユニットにどのように引き継ぐかを，後続ユニットとして定義したジョブの種類ごとに説明します。

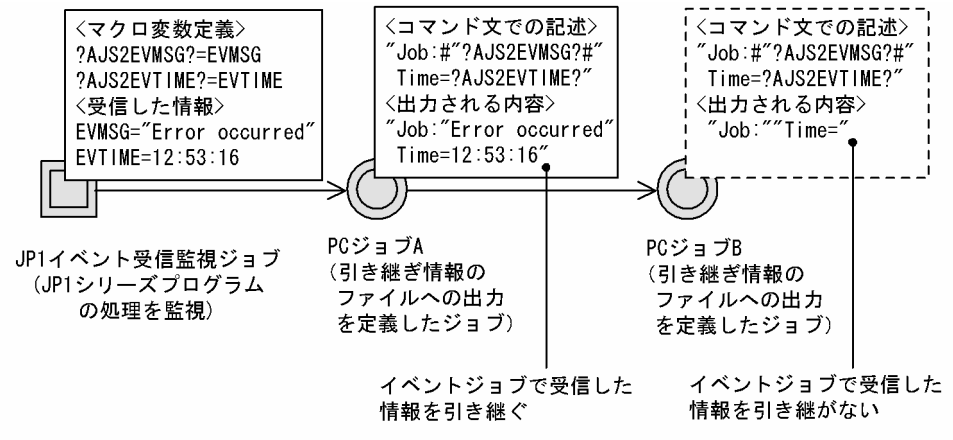
なお，イベントジョブの受信情報をジョブネット全体で参照する場合には，起動条件を使用してください。起動条件については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.4 起動条件の定義」を参照してください。

(a) 後続に標準ジョブ，HTTP 接続ジョブ，またはアクションジョブを定義した場合

イベントジョブの後続ジョブとして標準ジョブ（PC ジョブ，UNIX ジョブ，フレキシブルジョブ），HTTP 接続ジョブ，またはアクションジョブを定義した場合は，コマンドテキスト，スクリプトファイル名，パラメーター名，環境変数など，任意の文字列が設定できる項目でマクロ変数名を指定しておくと，受信したイベント情報をイベントジョブの直後のジョブに引き継げます。

イベントジョブの後続に PC ジョブを定義した例を次の図に示します。

図 2-85　イベントジョブの後続に PC ジョブを定義した例

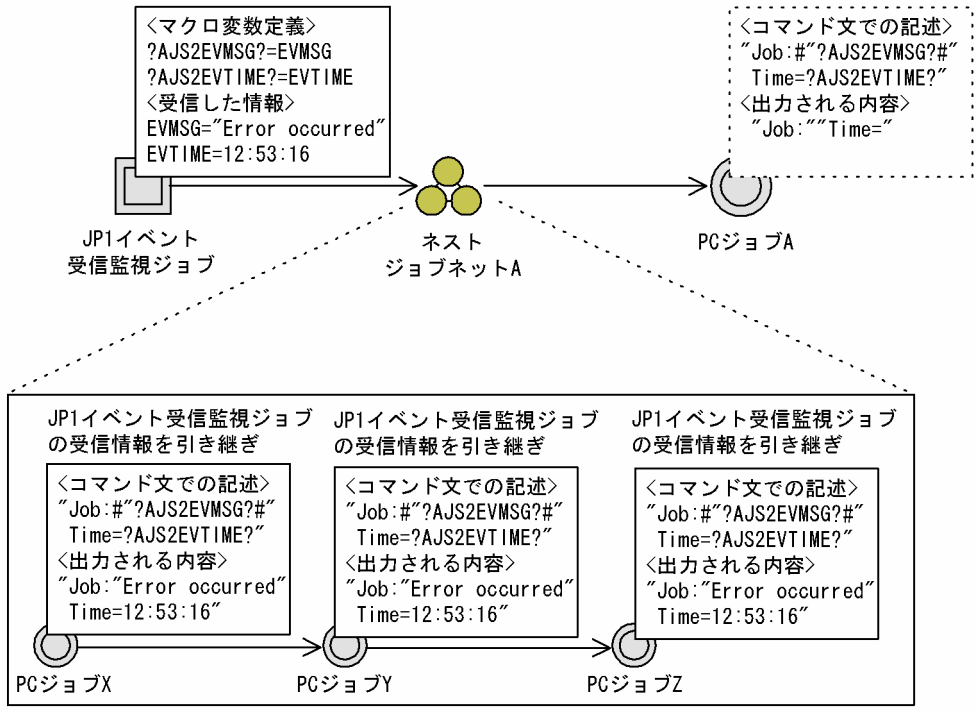


この例の場合，JP1 イベント受信監視ジョブで受信したイベント情報は，PC ジョブ A に引き継がれます。PC ジョブ A の後続ジョブである PC ジョブ B には引き継がれません。PC ジョブ A が異常終了した場合でも，受信情報は PC ジョブ B には引き継がれません。

(b)　後続にネストジョブネットを定義した場合

イベントジョブの後続ユニットとしてネストジョブネットを定義した場合は，ネストジョブネットでマクロ変数名を指定しておくと，ネストジョブネット中の標準ジョブ（PC ジョブ，UNIX ジョブ，フレキシブルジョブ），HTTP 接続ジョブ，およびアクションジョブに受信したイベント情報を引き継げます。イベントジョブの後続にネストジョブネットを定義した例を次の図に示します。

図 2-86　イベントジョブの後続にネストジョブネットを定義した例

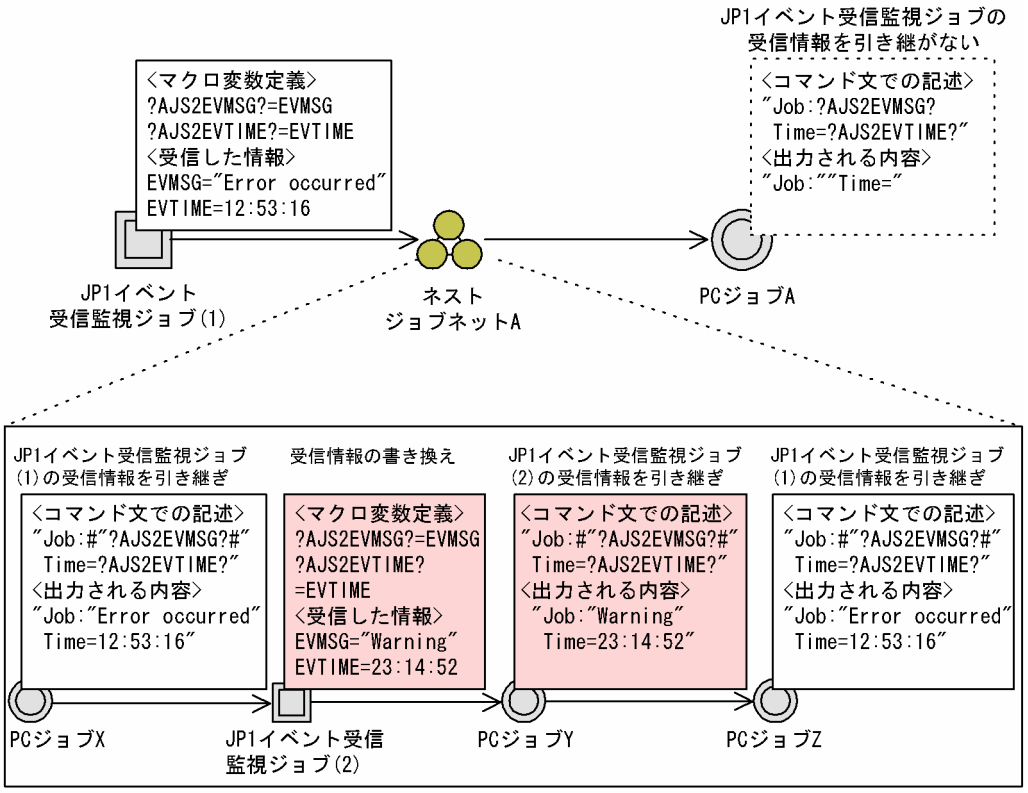


注　各標準ジョブでは，マクロ変数を引き継いでファイルに出力するジョブが定義されていることとします。

この例の場合，JP1 イベント受信監視ジョブで受信したイベント情報は，ネストジョブネット A に定義されている PC ジョブ X，PC ジョブ Y，および PC ジョブ Z に引き継がれます。

なお，ネストジョブネット中に同じマクロ変数を定義したイベントジョブがある場合は，ネストジョブネット中のイベントジョブの情報が優先されます。ネストジョブネット中に同じマクロ変数を定義したイベントジョブがある場合の例を次の図に示します。

図 2-87　ネストジョブネット中に同じイベントジョブを定義した例



注　各PCジョブでは，マクロ変数を引き継いでファイルに出力するジョブが定義されていることとします。

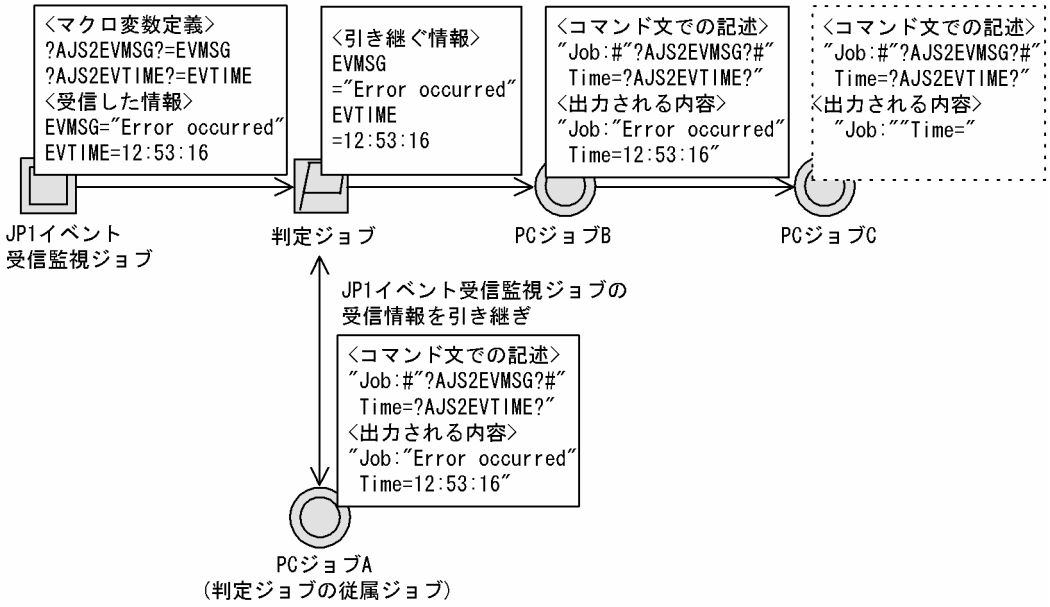
この例の場合，PC ジョブ X には JP1 イベント受信監視ジョブ(1)のイベント情報が引き継がれますが，PC ジョブ Y には JP1 イベント受信監視ジョブ(2)の受信情報が引き継がれます。なお，PC ジョブ Z には JP1 イベント受信監視ジョブ(1)のマクロ変数が指定されているため，JP1 イベント受信監視ジョブ(1)の受信情報が引き継がれます。

(c)　後続に判定ジョブを定義した場合

イベントジョブの後続ジョブとして判定ジョブを定義した場合は，判定ジョブの従属ジョブにマクロ変数名を指定しておくことで，従属ジョブと判定ジョブの後続ジョブとの両方に受信情報を引き継げます。

イベントジョブの後続に判定ジョブを定義した例を次の図に示します。

図 2-88　イベントジョブの後続に判定ジョブを定義した例



注　各PCジョブでは，マクロ変数を引き継いでファイルに出力するジョブが定義されていることとします。

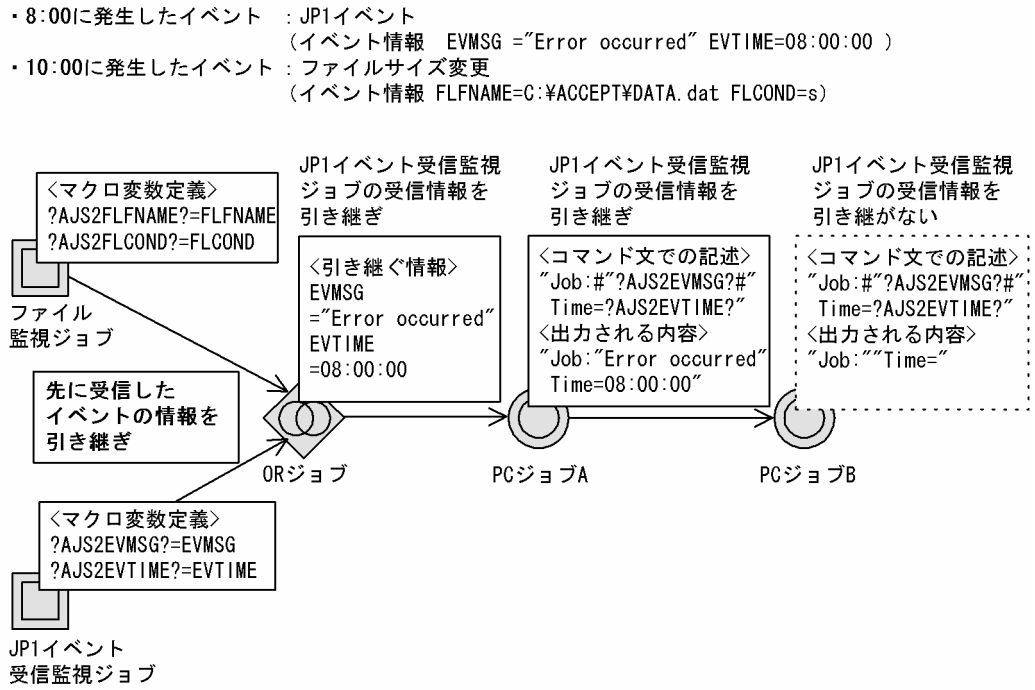
イベントジョブの後続ジョブとして判定ジョブを定義した場合は，判定ジョブの従属ジョブおよび判定ジョブの後続ジョブに受信情報が引き継がれます。したがってこの例の場合，JP1 イベント受信ジョブで受信したイベント情報は，判定ジョブの従属ジョブである PC ジョブ A と判定ジョブの後続ジョブである PC ジョブ B に引き継がれます。

(d)　後続に OR ジョブを定義した場合

イベントジョブの後続ジョブとして OR ジョブを定義した場合は，OR ジョブの後続ジョブにマクロ変数名を指定しておくと，OR ジョブの後続ジョブに受信情報を引き継げます。

イベントジョブの後続に OR ジョブを定義した例を次の図に示します。

図 2-89　イベントジョブの後続に OR ジョブを定義した例



注　各PCジョブでは，マクロ変数を引き継いでファイルに出力するジョブが定義されていることとします。

イベントジョブの後続ジョブとして OR ジョブを定義した場合は，OR ジョブの後続ジョブに受信情報が引き継がれます。この例では，ファイル監視ジョブと JP1 イベント受信監視ジョブの二つのイベントジョブでイベントの発生を監視していますが，8:00 に発生した JP1 イベントを監視している JP1 イベント受信監視ジョブの受信情報が OR ジョブの後続ジョブに引き継がれます。なお，この場合，ファイル監視ジョブは「計画未実行」状態となります。

注意事項

- 引き継ぎ情報は，イベントジョブの実行先のエージェントホストと後続ジョブの実行先のエージェントホストが異なる場合でも引き継げます。ホスト間の文字コードが異なる場合は，文字コードが変換されて引き継がれます。ただし，変換する文字コードに該当する文字がない場合は，正しく変換されません。
- マクロ変数を後続ジョブのコマンドラインに指定する場合，引き継ぐ情報に空白または「'（シングルクォーテーションマーク）」が含まれていると情報を正しく引き継げません。
- コマンドラインには，エスケープシーケンスを含むデータを引き継がないでください。また，空白を含んだデータを引き継いだ場合は，意図しない動作をすることがあるため，マクロ変数を「"（ダブルクォーテーションマーク）」で囲んで定義してください。
- 引き継ぎ情報をジョブのコマンドラインで使う場合に，例えば「AB"C」のように引き継ぐ情報に「"（ダブルクォーテーションマーク）」が入っているとき，「"（ダブルクォーテーションマーク）」は無視され，「ABC」という形で後続ジョブに引き継がれます。このように，各 OS のコマンドラインの制約によってジョブが正しく実行されないことがあります。特殊な文字が入っている情報を引き継ぐ場合には，コマンドラインに直接展開しないで環境変数に引き継ぐようにしてください。なお，「"（ダブルクォーテーションマーク）」を有効にする機能によって，「"（ダブルクォーテー

ションマーク)」を含む情報を引き継ぐことができます。「"（ダブルクォーテーションマーク)」を有効にする機能については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 4.3.1(5) 「"（ダブルクォーテーションマーク)」を含むデータをイベントジョブの引き継ぎ情報に使用する」，およびマニュアル「JP1/Automatic Job Management System 3 構築ガイド 6.3.4 「"（ダブルクォーテーションマーク)」を含むデータをイベントジョブの引き継ぎ情報に使用するための設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド 15.3.4 「"（ダブルクォーテーションマーク)」を含むデータをイベントジョブの引き継ぎ情報に使用するための設定」（UNIX の場合）を参照してください。ただし，JP1/AJS2 - Manager 06-71 以前を使用している場合，この機能は使用できません。

補足事項

- 複数のイベントジョブに対して後続ジョブが一つだけの場合でも，すべてイベントジョブで受信した情報を後続ジョブに引き継げます。ただし，複数のイベントジョブに同じマクロ変数名が定義されている場合は，受信したイベントジョブの情報が上書きされるため，最後に受信したイベントジョブの情報を参照することになります。
- 一つのイベントジョブの引き継ぎ情報に，同じ名称のマクロ変数が複数定義された場合，最初に定義された情報が引き継がれます。例えば，「?AJS2111?:EVID」という定義（マクロ変数「?AJS2111?」にイベント ID を引き継ぐように指定）を一つ目に設定し，「?AJS2111?:EVMSG」という定義（マクロ変数「?AJS2111?」にメッセージ情報を引き継ぐように指定）を二つ目に設定した場合，「?AJS2111?」で引き継がれる情報はイベント ID（EVID）だけです。
- イベントジョブから引き継ぐ情報がない場合，またはイベントジョブが実行されなかった場合，後続ジョブで定義したマクロ変数にはマクロ変数名の文字列が渡されます。例えば，マクロ変数名を「?AJS2111?」と定義している場合は，文字列「?AJS2111?」が渡されます。
- JP1 イベント受信監視ジョブでは，監視条件として拡張正規表現で使用できる「(」「)」を用いることで，「(」「)」内に含まれる正規表現に該当する文字列だけを切り出して引き継ぎ情報EVENVn（n:0～9）として後続ジョブに引き継ぐことができます。例えば，次の定義の JP1 イベント受信監視ジョブの実行中に，メッセージ「Event Test」を含む JP1 イベントを受信して監視条件成立した場合，後続ジョブで「?AJS2EVENV1?」を指定したときは「Event」，「?AJS2EVENV2?」を指定したときは「Test」という文字列が引き継がれます。

  ・メッセージ：(.*) (.*)

  ・引き継ぎ情報：?AJS2EVENV1?:EVENV1，?AJS2EVENV2?:EVENV2

  なお，Windows 環境で拡張正規表現を使用する場合，JP1/Base で使用する正規表現として拡張正規表現を指定する必要があります。詳細は JP1/Base のマニュアルを参照してください。

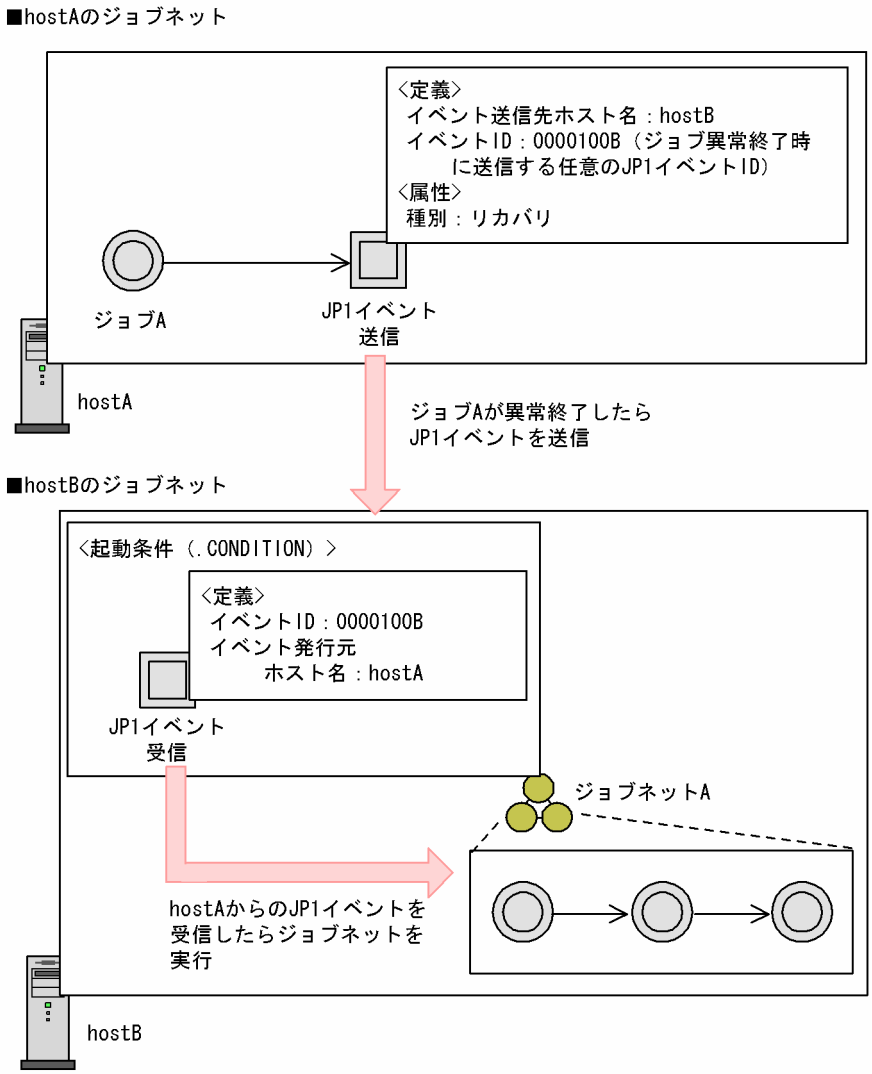
2.4.5 先行ジョブの終了や事象の発生を契機に JP1 イベントを送信する（JP1 イベント送信ジョブを使ったジョブネットの定義例）

先行ジョブの終了や事象の発生を契機に JP1 イベントを送信するというようなジョブネットの定義には，JP1 イベント送信ジョブを使用します。

なお，JP1 イベント送信ジョブを実行する際には，事前に送信元ホストと送信先ホストの双方で，JP1/Base イベントサービスを起動しておく必要があります。

「hostA で，あるジョブ（ジョブ A）が異常終了したときに hostB に JP1 イベントを送信する。hostB では，hostA から送信された JP1 イベントを受信すると，異常終了発生時のためのジョブネット（ジョブネット A）を実行する」というジョブネットを定義する例を次の図に示します。

図 2-90　JP1 イベント送信ジョブを使ったジョブネットの定義例



hostA には，ジョブ A の後続ジョブに JP1 イベント送信ジョブを定義します。JP1 イベント送信ジョブは，ジョブ A が異常終了したときに実行するジョブであるため，ジョブの［種別］を［リカバリ］と設定します。詳細については，「2.4.6 ジョブが異常終了したときに特定の処理を実行する（リカバリーユニットを使ったジョブネットの定義例）」を参照してください。また，ジョブ A が異常終了したときに JP1 イベントを hostB に送信するため，［イベント送信先ホスト名］を「hostB」と設定します。hostB に送信する JP1 イベント ID は，「0000100B」です。この JP1 イベント ID は任意に設定できます。

一方，hostB には，ジョブネット A を実行させる起動条件に hostA からの JP1 イベントを受信するための JP1 イベント受信監視ジョブを定義します。起動条件については，マニュアル「JP1/Automatic Job

Management System 3 導入ガイド　3.4 起動条件の定義」を参照してください。JP1 イベント受信監視ジョブには，［イベント発行元ホスト］に「hostA」，［イベント ID］に「0000100B」を設定します。

イベントの到達確認

JP1 イベント送信ジョブによって指定した送信先ホストに JP1 イベントが実際に送信されたかどうか確認できます。送信先ホストへの JP1 イベントの到達確認をする際，確認間隔（単位：秒）または確認回数を設定できます。

2.4.6　ジョブが異常終了したときに特定の処理を実行する（リカバリーユニットを使ったジョブネットの定義例）

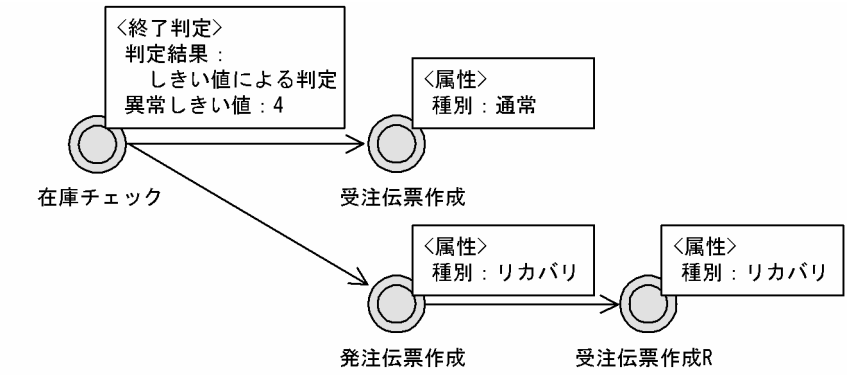
ジョブが異常終了したときにだけ処理を実行させるジョブネットの定義には，リカバリーユニットを使用します。

リカバリーユニットは，ジョブまたはジョブネットの属性定義で［種別］を［リカバリ］とすることで設定できます。また，リカバリー属性は，すべてのジョブおよびジョブネットに設定できます。

(1)　リカバリージョブを使ったジョブネットの定義例

「在庫チェックを実行して，在庫数が十分であれば受注伝票を作成する。在庫数が不足している場合はジョブを異常終了して発注伝票を作成したあと，受注伝票を作成する」というジョブネットを，リカバリージョブ使って定義する例を次の図に示します。

図 2–91　リカバリージョブを使ったジョブネットの定義例



この例では，「在庫チェック」は，実行の結果，在庫数が不足している場合は戻り値 5 以上を返すものとします。

まず，「在庫チェック」の終了判定を［しきい値による判定］とし，［異常しきい値］を 4 とします。つまり，戻り値が 4 以下ならば正常終了，戻り値が 5 以上ならば異常終了します。「在庫チェック」が正常終了した場合に次に実行する「受注伝票作成」は，［種別］は［通常］（デフォルト）のままにします。異常終了した場合に実行される「発注伝票作成」は，属性定義の［種別］を［リカバリ］とします。さらに「発注伝票作成」のあとに実行される「受注伝票作成 R」も同様に［リカバリ］を設定します。リカバリーユニットの後続ユニットには，リカバリーユニットしか定義できません。

これによって，「在庫チェック」の実行結果，以降のジョブは次のように実行されます。

- 戻り値が 4 以下の場合

  在庫数が十分であるとして「在庫チェック」は正常終了し，後続ジョブの「受注伝票作成」が実行されます。

- 戻り値が 5 以上の場合

  在庫数不足として「在庫チェック」は異常終了し，リカバリージョブの「発注伝票作成」が実行されます。さらに「発注伝票作成」が正常終了すると，リカバリージョブの後続ジョブである「受注伝票作成 R」が実行されます。

なお，先行ジョブの異常終了によって実行されたリカバリーユニットが正常終了しても，それらが定義されたジョブネット自体は「異常終了」になります。また，リカバリーユニットに先行ジョブが定義されていない場合，それらが実行されることはなく，「未実行終了」となります。

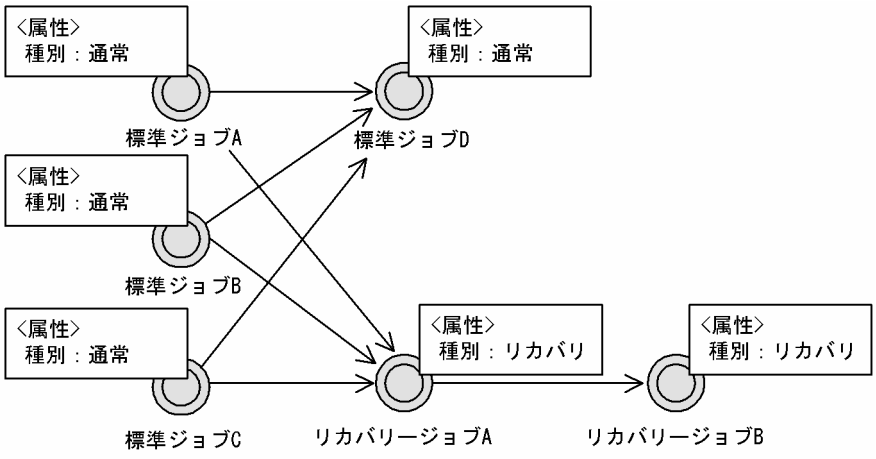
補足事項

OR ジョブをリカバリージョブとして定義する場合は，OR ジョブの先行ジョブとなるイベントジョブも後続ジョブもリカバリージョブとして定義します。

(2) 一つのリカバリーユニットに複数の先行ジョブを定義した例

一つのリカバリーユニットに対して複数の先行ジョブを定義できます。この場合，複数ある先行ジョブのすべてが異常終了したときにだけリカバリーユニットが実行されます。一つのリカバリージョブに複数の先行ジョブを定義した例を次の図に示します。

図 2-92 複数の先行ジョブを定義した例

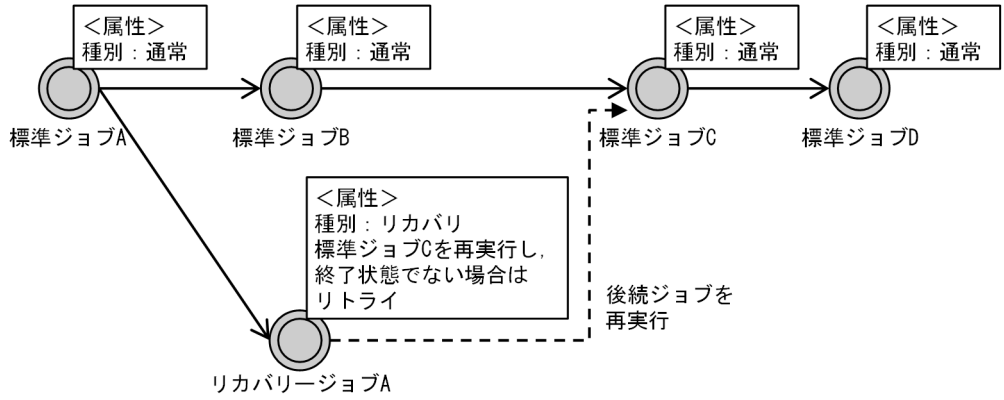


この場合，リカバリージョブ A は，標準ジョブ A，標準ジョブ B，および標準ジョブ C すべてが異常終了した場合にだけ実行されます。また，リカバリージョブ B はリカバリージョブ A が正常終了したあとに実行されます。

(3) 異常終了したユニットの後続ユニットを再実行する定義例

異常終了したユニットの後続ユニットを再実行する場合，タイミングによっては再実行に失敗することがあります。ajsrerun コマンドの-rr，-rn および-ri オプションを指定し，リカバリーユニットからコマンドを実行することで，対象ユニットが終了状態でない場合に再実行をリトライできます。異常終了したユニットの後続ユニットを再実行する定義例を次の図に示します。

図 2-93　異常終了したユニットの後続ユニットを再実行する定義例



この場合，標準ジョブ A が異常終了すると，リカバリージョブ A がajsrerun コマンドで標準ジョブ C を再実行します。標準ジョブ C が終了状態でない場合は，再実行をリトライします。

リカバリーユニットから再実行する場合の詳細については，「JP1/Automatic Job Management System 3 導入ガイド　4.5.11 実行が終了したジョブネットやジョブを再実行する」を参照してください。

ajsrerun コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajsrerun」を参照してください。

注意事項

異常終了したユニットの後続ユニットを再実行する場合，短い間隔で大量のリトライが実行されると，ジョブの実行に影響を与えるおそれがあります。リカバリーユニットで再実行を定義する場合は，次の点に注意してください。

- 再実行のリトライ回数およびリトライ間隔は，デフォルト値で運用することを推奨します。再実行のリトライ回数およびリトライ間隔を変更する場合は，使用している環境で事前に処理時間を測定および検証し，余裕のある値を設定してください。なお，ジョブネットは配下のユニットがすべて終了してから終了します。このため，異常終了したユニットの後続ユニットが「未実行終了」状態になるまでの時間を測定する場合は，ユニットが異常終了してから後続ユニットがすべて「未実行終了」状態となり上位ジョブネットが異常終了するまでの時間を測定してください。そのあと，再実行のリトライ回数およびリトライ間隔を運用に合わせて検証してください。
- 連続または同時にリトライを実行する場合は，リトライの間隔を空けてください。
- -rr，-rn および-ri を指定したajsrerun コマンドを大量に実行しないように，-rr，-rn および-ri を指定したajsrerun コマンドを実行するリカバリージョブの実行エージェントに，ジョブの実行多重度の設定を検討してください。

2.4.7　ルートジョブネットの実行順序を制御する（ジョブネットコネクタを使ったジョブネットの定義例）

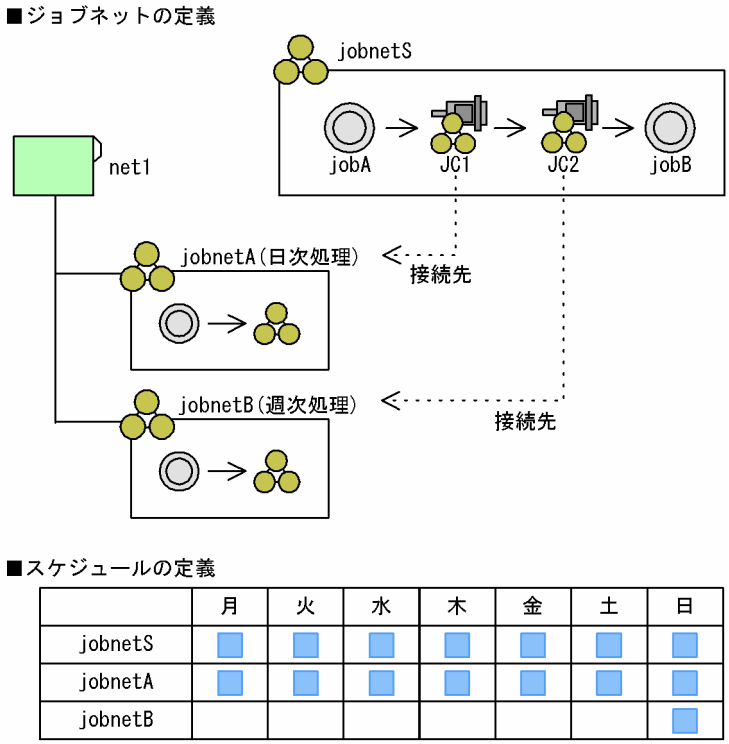
次のようなジョブネットを，ジョブネットコネクタを使って定義します。

- 日次処理のルートジョブネット「jobnetA」を，毎日ジョブ「jobA」のあとに実行する
- ルートジョブネット「jobnetA」のあとにジョブ「jobB」を実行する
- 週次処理のルートジョブネット「jobnetB」を，日曜日だけ jobnetA のあとに実行する

なお，ルートジョブネット「jobnetA」および「jobnetB」のスケジュールは，それぞれ日次，週次で実行するように定義済みであるとします。

この場合のジョブネットコネクタの定義例を，次の図に示します。

図 2-94　ジョブネットコネクタの定義例



この場合，まず，ジョブネットコネクタを定義するルートジョブネット「jobnetS」を作成し，「jobA－jobnetA－jobnetB－jobB」の順序で実行するようにジョブフローを作成します。jobnetA および jobnetB はルートジョブネットであるため，ジョブネットコネクタを適用します。jobnetA に対応するジョブネットコネクタを「JC1」，jobnetB に対応するジョブネットコネクタを「JC2」とし，「jobA－JC1－JC2－jobB」というジョブフローを定義します。JC1 の接続先のジョブネットには「jobnetA」を，JC2 の接続先のジョブネットには「jobnetB」を指定します。また，このルートジョブネット「jobnetS」のスケジュールを，1 日単位で実行するように設定します。

次に，接続先のジョブネット（jobnetA および jobnetB）に，それぞれに対応するジョブネットコネクタ名や実行順序制御方式などの情報を設定します。実行順序制御を「する」とし，jobnetA のジョブネット

コネクタ名には「JC1」を，jobnetB には「JC2」を指定します。また，それぞれの実行順序制御方式の設定で，ジョブネットコネクタと同期して実行させるか，非同期で実行するかを選択します。

ジョブネットコネクタを定義したジョブネット，および接続先のジョブネットの定義が完了したら，それぞれを実行登録します。実行登録をすると，接続ルールに従ってそれぞれ同一実行日にある世代が接続関係を結び，定義した実行順序で処理されます。

2.4.8 異なるジョブネットにあるユニット同士の実行順序を制御する（待ち合わせ条件を使ったジョブネットの定義例）

(1) 異なるジョブネットにあるユニット同士を連携する

異なるジョブネットにあるユニット同士の実行順序を制御するには，待ち合わせ条件を使用してユニットを定義します。

次のようなジョブネットを，待ち合わせ条件を使って定義することを想定します。

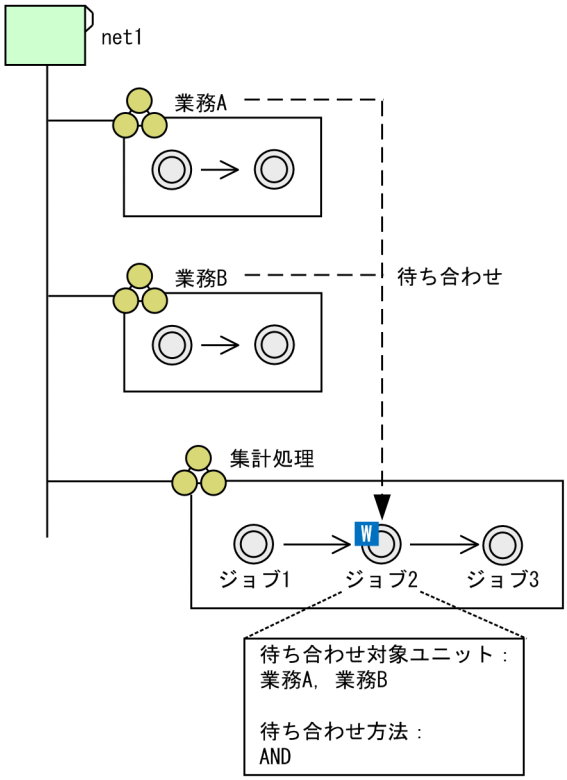
- 集計処理をするルートジョブネット「集計処理」は，「ジョブ 1」「ジョブ 2」「ジョブ 3」の順に実行する。
- 「ジョブ 2」の実行には，ルートジョブネット「業務 A」とルートジョブネット「業務 B」の出力するデータが必要となるため，「ジョブ 2」はルートジョブネット「業務 A」および「業務 B」の実行終了後に実行する。

なお，「業務 A」および「業務 B」は，1 日 1 回，月曜日～金曜日に実行するように定義済みであるとします。

このようなジョブネットを，待ち合わせ条件を使って定義する例を，次の図に示します。

図 2-95　待ち合わせ条件を使用したユニットの定義例

■ジョブネットの定義



■スケジュールの定義

	月	火	水	木	金	土	日
業務A	■	■	■	■	■		
業務B	■	■	■	■	■		
集計処理	■	■	■	■	■		

この例の場合，まず，集計処理をするルートジョブネット「集計処理」を作成し，「ジョブ 1－ジョブ 2－ジョブ 3」の順序で実行するようにジョブを定義します。ジョブ 2 は「業務 A」および「業務 B」の実行終了後に実行を開始する必要があるため，ジョブ 2 には待ち合わせ条件を定義します。待ち合わせ対象ユニットには「業務 A」および「業務 B」を，待ち合わせ方法には「AND」を指定します。待ち合わせ対象ユニットに実行予定世代がない場合の設定は，運用に合わせて決定します。

次に，「集計処理」のスケジュールを定義します。「集計処理」は「業務 A」および「業務 B」と同じ日に実行する必要があるため，スケジュールルールは「業務 A」および「業務 B」に合わせて定義します。この例の場合，「業務 A」および「業務 B」は月曜日～金曜日に実行するように定義されているため，「集計処理」も月曜日～金曜日に実行するように定義します。

ユニットの定義，待ち合わせ条件の設定，およびスケジュールの定義が完了したら，「業務 A」，「業務 B」，および「集計処理」を実行登録します。実行登録すると，ジョブ 2 は「業務 A」および「業務 B」が実行終了するのを待ち合わせ，「業務 A」および「業務 B」が実行終了すると実行を開始します。

実行開始後の待ち合わせ条件付きユニット，および待ち合わせ対象ユニットの運用方法については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　8.3 待ち合わせ条件に関する運用方法」を参照してください。

(2)　ジョブフローに繰り返し処理を組み込む（起動条件と待ち合わせ条件を組み合わせた定義例）

繰り返し処理を行うユニットをジョブフローに組み込むには，起動条件と待ち合わせ条件を組み合わせてユニットを定義します。

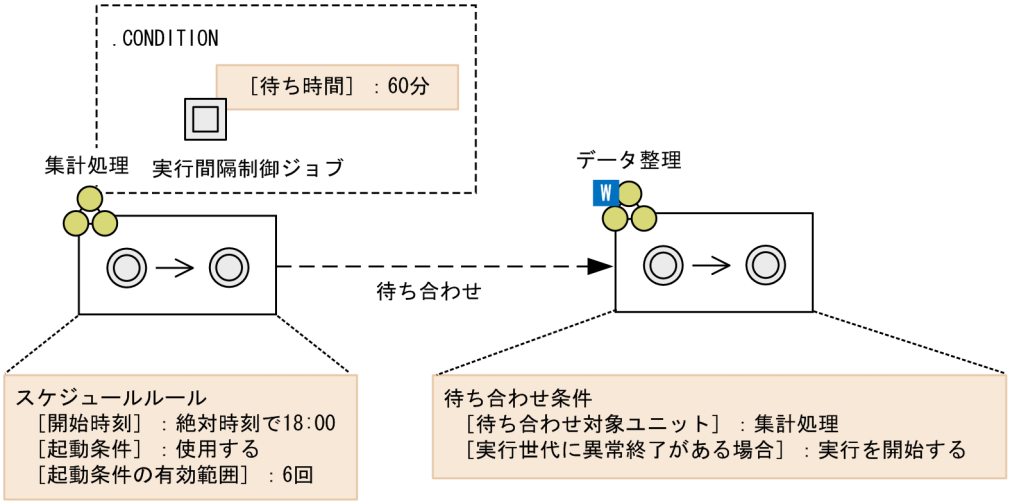
次のような業務を，待ち合わせ条件を使って定義することを想定します。

- データの集計処理業務を，ルートジョブネット「集計処理」として定義する。ルートジョブネット「集計処理」は，18:00 から 1 時間おきに 6 回実行する。
- 集計したデータを整理する業務を，ルートジョブネット「データ整理」として定義する。ルートジョブネット「データ整理」は，6 回の集計業務が正常終了しているかどうかに関係なく，実行する。

なお，スケジュールはそれぞれのルートジョブネットで定義済みであるとします。

このような業務を，起動条件と待ち合わせ条件を使って定義する例を，次の図に示します。

図 2-96　起動条件と待ち合わせ条件を組み合わせたユニットの定義例



この例の場合，まず，ルートジョブネット「集計処理」を定義します。「集計処理」で実行するジョブを定義したあと，ルートジョブネットが一定間隔で実行するように設定します。設定手順を次に示します。

1. 起動条件を作成し，実行間隔制御ジョブを定義する。
2. 実行間隔制御ジョブの［待ち時間］に 60 分を設定する。
3. スケジュールルールを作成し，次のように定義する。
   - ［開始時刻］：絶対時刻で 18:00
   - ［起動条件］：使用する

- ［起動条件の有効範囲］の［回数］：6 回

次に，ルートジョブネット「データ整理」を定義します。「データ整理」で実行するジョブを定義したあと，「集計処理」の実行終了を待ち合わせるために，待ち合わせ条件を定義します。待ち合わせ条件の設定項目を次に示します。

- ［待ち合わせ対象ユニット］：集計処理
- ［実行世代に異常終了がある場合］：実行を開始する

ユニットの定義，スケジュールの定義，および待ち合わせ条件の設定が完了したら，「集計処理」および「データ整理」を実行登録します。実行登録すると，「集計処理」は 18:00 から 1 時間おきに 6 回実行します。「集計処理」のすべての実行世代が実行終了すると，実行世代が異常終了したかどうかに関係なく，「データ整理」が実行を開始します。

実行開始後の待ち合わせ条件付きユニット，および待ち合わせ対象ユニットの運用方法については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　8.3 待ち合わせ条件に関する運用方法」を参照してください。

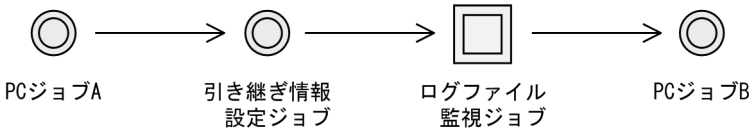
2.4.9　動的に変わる情報を後続ユニットに引き継ぐ（引き継ぎ情報設定ジョブを使ったジョブネットの定義例）

動的に変化する情報を使った処理を実行させるようなジョブネットの定義には，引き継ぎ情報設定ジョブを使用します。

引き継ぎ情報設定ジョブでは，先行ジョブの標準出力ファイルから必要な情報を切り出してグローバルマクロ変数に設定し，後続ユニットへ引き継ぎます。グローバルマクロ変数とは，マクロ変数の一種で，引き継ぎ情報設定ジョブによって設定されるマクロ変数のことです。マクロ変数の詳細については，「2.2.6 マクロ変数の使用を検討する」を参照してください。

引き継ぎ情報設定ジョブを使用した情報の引き継ぎの例を次に示します。

図 2-97　引き継ぎ情報設定ジョブを使用した情報の引き継ぎ例



各ユニットは，次のように定義されているものとします。

- PC ジョブ A
  - 実行ファイル名：日付を標準出力に出力するバッチファイル
  - 標準出力ファイル名：ファイル名
- 引き継ぎ情報設定ジョブ

- 正規表現：「DATE=(.*)」
- 出力マクロ変数：「?AJS2FILEDATE?」

- ログファイル監視ジョブ
  - ログファイル名：「?AJS2FILEDATE?.txt」
  - トラップデータ：「Error」
- PC ジョブ B
  - 実行ファイル名：エラー対処のバッチファイル

PC ジョブ A および PC ジョブ B の実行ファイル名に指定するバッチファイルは，ジョブ実行前に作成しておく必要があります。

このジョブネットを実行すると，次のように実行されます。

1. PC ジョブ A が，日付情報を標準出力ファイルに出力する。
   標準出力ファイルに「DATE=20XX1010」と出力されます。
2. 引き継ぎ情報設定ジョブが，PC ジョブ A が出力した標準出力ファイルから日付部分を切り出す。
   マクロ変数「?AJS2FILEDATE?」に「20XX1010」が設定されます。
3. ログファイル監視ジョブが，ファイルを監視する。
   「20XX1010.txt」という名前のファイルに「Error」という文字列が書き込まれたら，条件が成立します。
4. PC ジョブ B が，エラー対処の処理を実行する。

(1) 定義方法

引き継ぎ情報設定ジョブを使用するための，先行ジョブ，引き継ぎ情報設定ジョブ，および後続ユニットの定義方法について説明します。

(a) 引き継ぎ情報設定ジョブの先行ジョブ

引き継ぎ情報設定ジョブの先行ジョブは，次の条件をすべて満たしている必要があります。

- 先行ジョブが一つ以上ある
- 先行ジョブのうち，次の条件をすべて満たすものが一つだけある
  - PC ジョブ※1，UNIX ジョブ※1，フレキシブルジョブ※2，HTTP 接続ジョブ，およびカスタムジョブのどれか
  - 標準出力ファイル名に，ファイル名または「$JP1AJS2_JPQSTDOUTTEMP$」の指定がある※3，※4
  - 実行したときに結果を標準出力ファイルに出力する

注※1

キューレスジョブを除きます。

注※2

フレキシブルジョブの後続の引き継ぎ情報設定ジョブが引き継げる標準出力ファイルの内容は，4,096 バイトまでです。4,096 バイトを超える値を出力しても，使用できるのは 4,096 バイト分までです。4,096 バイトの制限は，中継エージェントおよび宛先エージェントの両方で実施します。このため，中継エージェントと宛先エージェントの両方の文字コードで，4,096 バイト以内となるようにしてください。なお，4,096 バイトから 4,097 バイト目のデータがマルチバイト文字の場合は，「?」に置き換えられます。

注※3

ジョブの標準出力ファイルをエージェントホストに残したい場合は，ファイル名を指定します。指定したファイルがマネージャーホストに転送されても，エージェントホストに残ります。

ジョブの標準出力ファイルをエージェントホストに残す必要がない場合は，「$JP1AJS2_JPQSTDOUTTEMP$」を指定します。この場合，標準出力ファイルは，エージェントホストの一時ファイルに出力され，マネージャーホストに転送されます。追加書きを指定しても無効です。マネージャーホストに転送されたファイルのファイル名は，ajsshow コマンドの-i オプションに 2 バイトフォーマット指示子の「%so」を指定すると確認できます。

注※4

フレキシブルジョブの場合，「$JP1AJS2_JPQSTDOUTTEMP$」の指定は不要です。

注意事項

- 引き継ぎ情報設定ジョブを判定ジョブの従属ジョブとして定義した場合は，判定ジョブの先行ジョブが引き継ぎ情報設定ジョブの先行ジョブとして扱われます。そのため，判定ジョブの先行ジョブに標準出力ファイル名の指定が必要です。
- ファイル受信制限の設定によって標準出力ファイルの受信処理が中止された場合，標準出力ファイルが完全に作成されません。このような場合，不完全な標準出力ファイルでも引き継ぎ情報設定ジョブで情報が切り出せるかどうか確認してください。なお，JP1/AJS3 を新規インストールした場合，5 メガバイトを超えるデータを破棄するように設定されます。この設定によって情報の切り出しに失敗する場合は，環境設定パラメーターLimitReceiveFileSize の値を適切に設定してください。
- ファイル送信制限の設定によって標準出力ファイルの送信処理が中止された場合，標準出力ファイルが完全に作成されません。このような場合，不完全な標準出力ファイルでも引き継ぎ情報設定ジョブで情報が切り出せるかどうか確認してください。なお，JP1/AJS3 を新規インストールした場合，3 メガバイトを超えるデータを破棄するように設定されます。この設定によって情報の切り出しに失敗する場合は，環境設定パラメーターLimitSendFileSize の値を適切に設定してください。

(b) 引き継ぎ情報設定ジョブ

引き継ぎ情報設定ジョブには，次の指定をします。

正規表現

先行ジョブが出力した標準出力ファイルから必要な情報を切り出すための，正規表現を指定します。

出力マクロ変数

正規表現に従って切り出した情報を設定する，マクロ変数名を指定します。

先行ジョブが出力した標準出力ファイルのどの行にも正規表現が一致しない場合，出力マクロ変数にはNULL 文字列が設定されます。

引き継ぎ情報設定ジョブが実行終了したときの戻り値を，次の表に示します。

表 2-20　引き継ぎ情報設定ジョブの戻り値

先行ジョブの戻り値	引き継ぎ情報設定ジョブの戻り値	条件
0	0	すべての正規表現が一致した場合
	1	一致しない正規表現が一つ以上あった場合
	20 以上	その他のエラーが発生した場合
0 以外	先行ジョブの戻り値	環境設定パラメーターPassingInfoUsePreRc が「1」の場合
	0	環境設定パラメーターPassingInfoUsePreRc に「0」を設定し，かつ，すべての正規表現が一致した場合
	1	環境設定パラメーターPassingInfoUsePreRc に「0」を設定し，かつ，一致しない正規表現が一つ以上あった場合
	20 以上	環境設定パラメーターPassingInfoUsePreRc に「0」を設定し，かつ，その他のエラーが発生した場合

一致しない正規表現があっても後続ユニットの実行を継続したい場合は，次に示す方法で対処してください。

- 環境設定パラメーターPassingInfoUsePreRc が「0」の場合
  引き継ぎ情報設定ジョブの異常しきい値に「1」に設定してください。
- 環境設定パラメーターPassingInfoUsePreRc が「1」の場合
  先行ジョブの戻り値が 0 になるようにして，引き継ぎ情報設定ジョブの異常しきい値を「1」に設定してください。

実行終了した引き継ぎ情報設定ジョブを再実行すると，出力マクロ変数には，再実行時の標準出力ファイルおよび正規表現に従って切り出した情報が再設定されます。

引き継ぎ情報設定ジョブで指定した正規表現は，使用する正規表現を JP1/Base で拡張しているかどうかに関係なく，拡張正規表現として扱われます。拡張正規表現の詳細については，マニュアル「JP1/Base 運用ガイド」の，拡張正規表現について記載されている個所を参照してください。

拡張正規表現の指定方法を次に示します。

- 切り出したい部分を「(」と「)」で囲む。
- 「.（ピリオド）」は，改行文字以外の任意の 1 文字を意味する。
- 「$」は，末尾指定（後方一致）を意味する。

注意事項

- 複数組の「(」および「)」を指定した場合は，いちばん左の組での指定と一致する文字列が切り出されます。
- 「(」および「)」を指定しない場合は，拡張正規表現全体と一致する文字列が切り出されます。
- 行単位に，指定した拡張正規表現で切り出します。複数行にわたる切り出しはできません。
- 「(」に対応する「)」がないなど，文法が正しくない拡張正規表現を指定した場合，引き継ぎ情報設定ジョブは異常終了します。
- 標準出力ファイルの 1 行当たりの長さは，最大 1,024 バイトです。1,024 バイトを超える行に対して切り出し処理を実行すると，引き継ぎ情報設定ジョブが異常終了します。
- 拡張正規表現に一致する文字列が複数ある場合，ファイルの先頭から最初に一致した文字列を切り出します。

拡張正規表現で切り出される情報の例を，次の表に示します。

表 2-21　拡張正規表現の指定例

切り出し対象の文字列	パターン	拡張正規表現	切り出される情報
abcdefghi	一組の「(」「)」を指定	abc(...)ghi	def
	複数組の「(」「)」を指定	abc(.)(.)(.)ghi	d
	「(」「)」を指定しない	...$	ghi

(c)　引き継ぎ情報設定ジョブの後続ユニット

引き継ぎ情報設定ジョブの出力マクロ変数に指定した，マクロ変数名を指定します。

(2)　引き継ぎ情報の有効範囲

引き継ぎ情報の有効範囲は，ルートジョブネット配下です。ただし，引き継ぎ情報設定ジョブで切り出した引き継ぎ情報がマクロ変数に設定されるのは，引き継ぎ情報設定ジョブの実行後です。引き継ぎ情報設定ジョブの実行前は，マクロ変数に設定された値を参照できません。引き継ぎ情報設定ジョブが設定したマクロ変数は，ジョブの再実行時にもそのまま使用されます。

次の条件を例に，実行したジョブとマクロ変数の値について説明します。

- ジョブ A およびジョブ B に，マクロ変数「?AJS2PRM?」を定義
- 引き継ぎ情報設定ジョブに，「?AJS2PRM?」に「PARAM」を指定

図 2-98　引き継ぎ情報設定ジョブの例

この図の場合で，実行したジョブとその時点でのマクロ変数の値について次の表に示します。

表 2-22　実行したジョブとマクロ変数の値

実行したジョブ	マクロ変数の値
ジョブ A	?AJS2PRM?
引き継ぎ情報設定ジョブ	－
ジョブ B	PARAM
ジョブ A（再実行）	PARAM

(3)　引き継ぎ結果の参照

引き継ぎ情報設定ジョブが含まれるジョブネットの定義によって，引き継ぎ結果の参照可否が異なります。

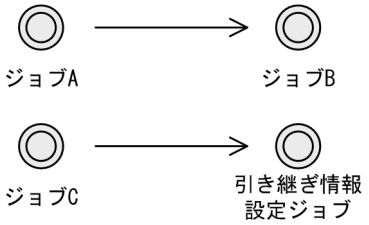
引き継ぎ結果の確認方法については，「2.2.6(2) マクロ変数の確認」を参照してください。

引き継ぎ結果の参照可否について説明します。

(a)　関連線がない

引き継ぎ情報設定ジョブと関連線がないジョブの場合，引き継ぎ情報設定ジョブが先に実行されていれば参照できます。

図 2-99　関連線がない例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

表 2-23　関連線がない場合の参照可否

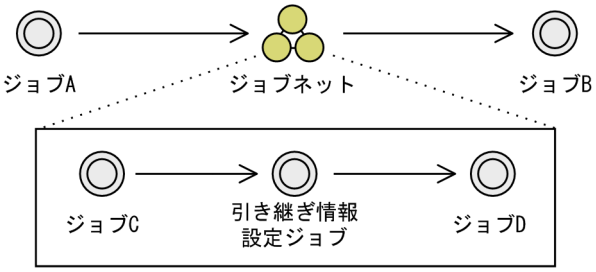
ジョブ	引き継ぎ結果の参照可否
ジョブ A	引き継ぎ情報設定ジョブが先に実行されていれば，参照できる。再実行時も同様。 引き継ぎ情報設定ジョブがあとで実行された場合は，参照できない。

ジョブ	引き継ぎ結果の参照可否
ジョブ B	引き継ぎ情報設定ジョブが先に実行されていれば，参照できる。再実行時も同様。 引き継ぎ情報設定ジョブがあとで実行された場合は，参照できない。
ジョブ C	ルートジョブネット実行時は参照できない。 ジョブ C の再実行時は参照できる。

(b)　ネストジョブネット

ネストジョブネット中の引き継ぎ情報設定ジョブの引き継ぎ結果は，ルートジョブネット中のジョブでも参照できます。

図 2-100　ネストジョブネットの例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

表 2-24　ネストジョブネットの場合の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	ルートジョブネット実行時は参照できない。 ジョブ A の再実行時は参照できる。
ジョブ B	参照できる。
ジョブ C	ルートジョブネット実行時は参照できない。 ジョブ C の再実行時は参照できる。
ジョブ D	参照できる。

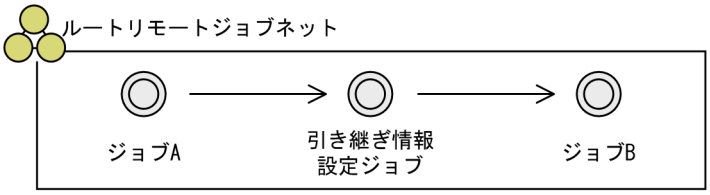
(c)　リモートジョブネット

リモートジョブネットを含むジョブネットに引き継ぎ情報設定ジョブを定義している場合の，引き継ぎ結果の参照可否について説明します。

■ ルートリモートジョブネットの場合

ルートリモートジョブネットに引き継ぎ情報設定ジョブがある場合，ルートリモートジョブネット内で引き継ぎ結果を参照できます。

図 2-101　ルートリモートジョブネットの例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

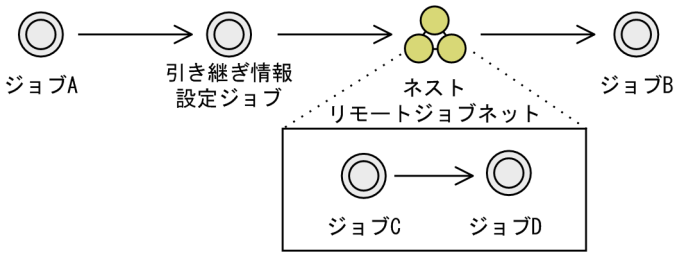
表 2-25　ルートリモートジョブネットの場合の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	ルートジョブネット実行時は参照できない。 ジョブ A の再実行時は参照できる。
ジョブ B	参照できる。

■ ネストリモートジョブネットの場合（ルートジョブネット中に引き継ぎ情報設定ジョブがある）

ルートジョブネット中に引き継ぎ情報設定ジョブがある場合，ルートジョブネット内のジョブだけ引き継ぎ結果を参照できます。ネストリモートジョブネット内のジョブでは参照できません。

図 2-102　ネストリモートジョブネット（ルートジョブネット中に引き継ぎ情報設定ジョブがある）の例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

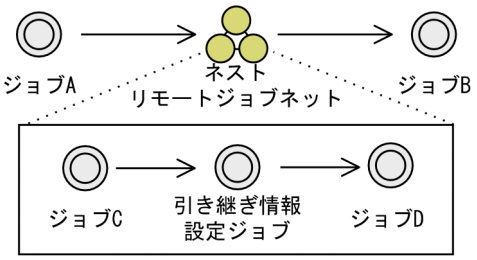
表 2-26　ネストリモートジョブネット（ルートジョブネット中に引き継ぎ情報設定ジョブがある）の場合の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	ルートジョブネット実行時は参照できない。 ジョブ A の再実行時は参照できる。
ジョブ B	参照できる。
ジョブ C	参照できない。
ジョブ D	参照できない。

■ ネストリモートジョブネットの場合（ネストリモートジョブネット中に引き継ぎ情報設定ジョブがある）

ネストリモートジョブネット中に引き継ぎ情報設定ジョブがある場合，ネストリモートジョブネット内のジョブだけ引き継ぎ結果を参照できます。ルートジョブネット内のジョブでは参照できません。

図 2-103　ネストリモートジョブネット（ネストリモートジョブネット中に引き継ぎ情報設定ジョブがある）の例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

表 2-27　ネストリモートジョブネット（ネストリモートジョブネット中に引き継ぎ情報設定ジョブがある）の場合の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	参照できない。
ジョブ B	参照できない。
ジョブ C	ルートジョブネット実行時は参照できない。 ジョブ C の再実行時は参照できる。ただし，ネストリモートジョブネットを再実行した場合は，再実行後でも参照できない。
ジョブ D	参照できる。

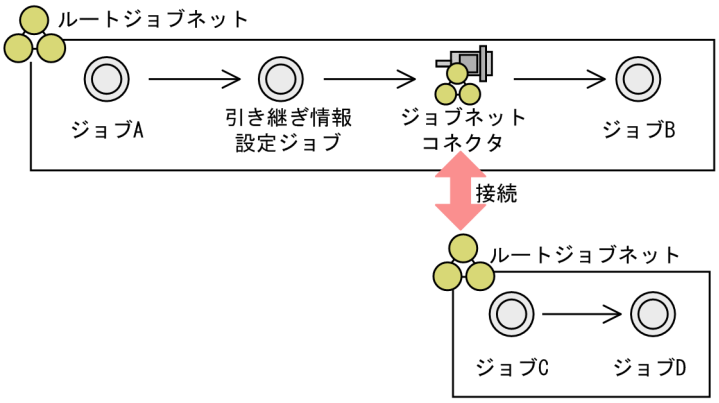
(d)　ジョブネットコネクタ

ジョブネットコネクタを定義しているジョブネットまたは接続先ジョブネットで定義した引き継ぎ情報設定ジョブの引き継ぎ結果は，引き継ぎ情報設定ジョブがある側のジョブネットだけ参照できます。

■ ジョブネットコネクタを定義しているジョブネットに引き継ぎ情報設定ジョブがある場合

ジョブネットコネクタを定義しているジョブネットだけ引き継ぎ結果を参照できます。

図 2-104　ジョブネットコネクタを定義しているジョブネットに引き継ぎ情報設定ジョブがある例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

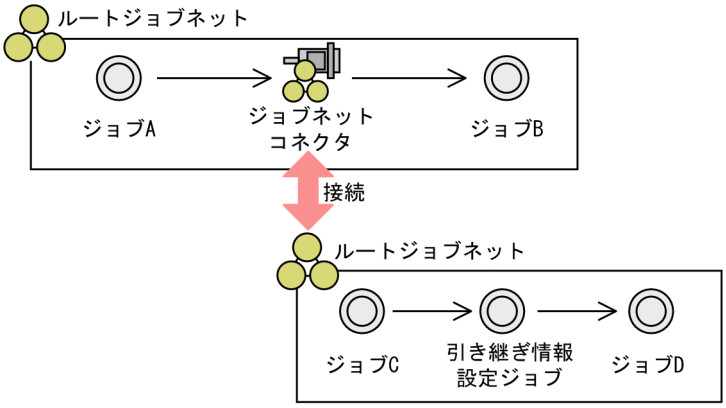
表 2-28　ジョブネットコネクタを定義しているジョブネットに引き継ぎ情報設定ジョブがある場合の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	ルートジョブネット実行時は参照できない。 ジョブ A の再実行時は参照できる。
ジョブ B	参照できる。
ジョブ C	参照できない。
ジョブ D	参照できない。

■ 接続先のジョブネットに引き継ぎ情報設定ジョブがある場合

接続先のジョブネットだけ引き継ぎ結果を参照できます。

図 2-105　ジョブネットコネクタの接続先ジョブネットに引き継ぎ情報設定ジョブがある例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

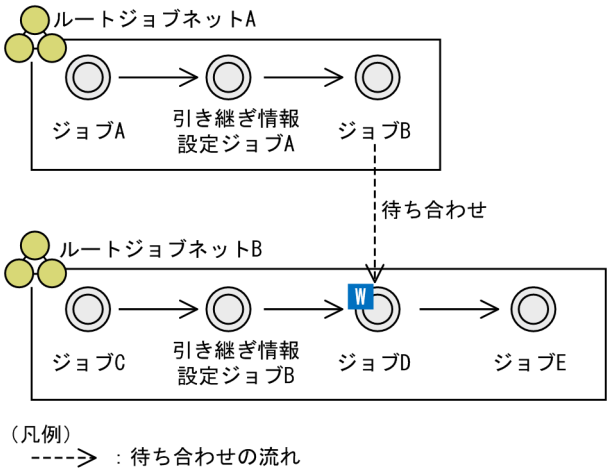
表 2-29　ジョブネットコネクタの接続先ジョブネットに引き継ぎ情報設定ジョブがある場合の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	参照できない。
ジョブ B	参照できない。
ジョブ C	ルートジョブネット実行時は参照できない。 ジョブ C の再実行時は参照できる。
ジョブ D	参照できる。

(e)　待ち合わせ条件

待ち合わせ条件付きユニットと，その待ち合わせ対象ユニットが別のルートジョブネット配下にある場合，引き継ぎ結果はそれぞれのルートジョブネット配下だけ参照できます。

図 2-106　待ち合わせ条件を使用した例



この図の場合の，引き継ぎ結果の参照可否について次の表に示します。

表 2-30　引き継ぎ情報設定ジョブ A の引き継ぎ結果の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	ルートジョブネット A 実行時は参照できない。 ジョブ A の再実行時は参照できる。
ジョブ B	参照できる。
ジョブ C	参照できない。
ジョブ D	参照できない。
ジョブ E	参照できない。

表 2-31　引き継ぎ情報設定ジョブ B の引き継ぎ結果の参照可否

ジョブ	引き継ぎ結果の参照可否
ジョブ A	参照できない。
ジョブ B	参照できない。
ジョブ C	ルートジョブネット B 実行時は参照できない。 ジョブ C の再実行時は参照できる。
ジョブ D	参照できる。
ジョブ E	参照できる。

(4)　注意事項

引き継ぎ情報設定ジョブを使用する場合の注意事項を，次に示します。

- 引き継ぎ情報設定ジョブを，JP1/AJS3 - Manager のバージョンが 09-50 より前の環境にコピーして使用した場合，引き継ぎ情報設定ジョブの実行時にエラーになります。
- 引き継ぎ情報設定ジョブには，実行エージェントを指定できません。実行登録されたマネージャーホストのデフォルト実行エージェント「@SYSTEM」で実行されます。リモートジョブネット配下の引き継ぎ情報設定ジョブは，そのリモートジョブネットの転送先ホスト上のデフォルト実行エージェントで実行されます。
- 一つのルートジョブネット配下に，同じ名称のマクロ変数を指定した引き継ぎ情報設定ジョブが複数ある場合，あとから実行した引き継ぎ情報設定ジョブの引き継ぎ情報が有効です。マクロ変数使用時の注意事項については，「2.2.6(5) 注意事項」を参照してください。
- 引き継ぎ情報設定ジョブを使用する場合は，ジョブが出力したデータが結果ファイルに出力されないことを防止するための設定を有効にしてください。詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.2.26 ジョブが出力したデータが結果ファイルに出力されないことを防止するための設定」を参照してください。

2.4.10　クラウド環境でジョブを実行する（フレキシブルジョブを使ったジョブネットの定義例）

クラウド上のホスト名や IP アドレスが固定されていない環境でジョブを実行させるジョブネットの定義には，フレキシブルジョブを使用します。フレキシブルジョブを使用すると，マネージャーホストで直接管理していないエージェントホストで処理を実行できます。そのため，次のようにオートスケール環境でジョブを実行する場合に有効です。

- ロードバランサーを利用して，ジョブの実行先エージェントを振り分ける場合
- 同一のジョブを，オートスケール環境にあるすべての実行エージェントで一斉に実行する場合

ただし，フレキシブルジョブは次のような場合に利用することを推奨します。

- ホスト名や IP アドレスが固定されていない環境でジョブを実行する場合

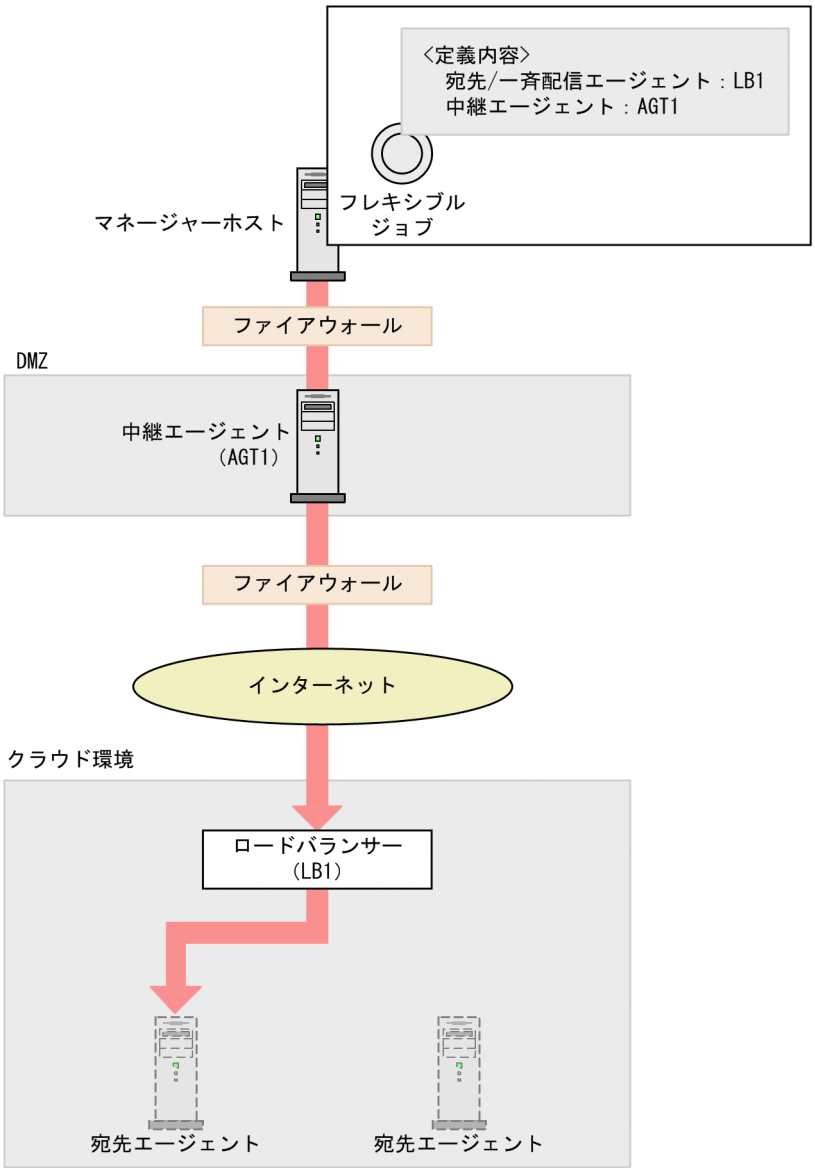
  ホスト名や IP アドレスが固定されている場合，PC ジョブや UNIX ジョブの使用を推奨します。

- 実行に失敗してもエラーの要因分析が不要で，再実行すればよいジョブを実行する場合

  フレキシブルジョブは，オートスケール環境ではスケールインによって宛先エージェントが削除されて実行に失敗することがあり，エラーの要因分析が困難です。したがって，オートスケール環境で実行する処理は，実行に失敗してもエラーの要因分析が不要で，再実行すればよい処理とすることを推奨します。

ロードバランサーを利用したフレキシブルジョブの定義例を次の図に示します。なお，フレキシブルジョブを実行する場合のシステム構成や検討事項などについては，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 2.9 クラウド環境でのジョブ実行について検討する」を参照してください。また，フレキシブルジョブを一斉実行する場合の定義例については，「2.4.11 一つのジョブを複数の実行エージェントで一斉に実行する（フレキシブルジョブを使ったジョブネットの定義例）」を参照してください。

図 2-107　ロードバランサーを利用したフレキシブルジョブの定義例



この図では，次のようなジョブネットを定義しています。

- 中継エージェント（AGT1）でジョブの実行要求を中継する。
- ジョブをロードバランサー（LB1）で振り分け，宛先エージェントで実行する。

フレキシブルジョブ使用時の注意事項については，「7.8 フレキシブルジョブ使用時の注意事項」を参照してください。

フレキシブルジョブを実行する場合に必要な設定については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　21.3 フレキシブルジョブを実行するための設定」を参照してください。

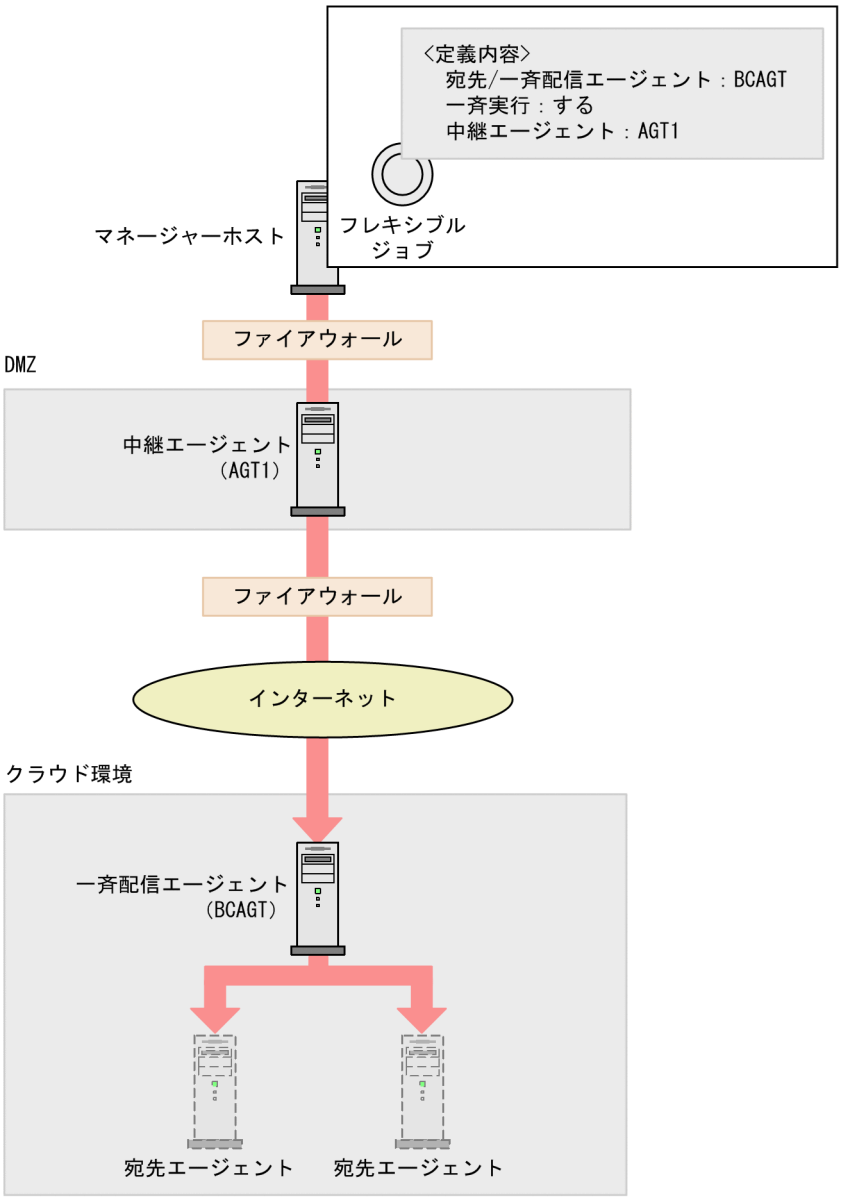
2.4.11　一つのジョブを複数の実行エージェントで一斉に実行する（フレキシブルジョブを使ったジョブネットの定義例）

一つのジョブを複数の実行エージェントで一斉に実行する機能を**一斉実行**と呼びます。一斉実行するジョブネットの定義には，フレキシブルジョブを使用します。

フレキシブルジョブの一斉実行を使用すると，複数の実行エージェントに同じ処理を実行させたい場合に，一つのジョブを定義するだけで済みます。また，実行エージェントを追加・削除した場合，ジョブを一斉に配信するエージェント（一斉配信エージェント）が自動で配信する範囲を拡張・縮小するため，ジョブを追加・削除する必要がありません。

フレキシブルジョブの一斉実行を使ったジョブネットを定義する例を次の図に示します。なお，フレキシブルジョブを実行する場合のシステム構成や検討事項などについては，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 2.9 クラウド環境でのジョブ実行について検討する」およびマニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 2.10 ジョブの一斉実行について検討する」を参照してください。

図 2-108　フレキシブルジョブの一斉実行を使ったジョブネットの定義例



この図では，次のようなジョブネットを定義しています。

- 中継エージェント（AGT1）でジョブの実行要求を中継する。
- 一斉配信エージェント（BCAGT）でジョブを一斉に配信する。
- クラウド環境にある複数の宛先エージェントでジョブを実行する。

フレキシブルジョブの一斉実行を使用する場合，宛先エージェントがオートスケールによって追加・削除されたときでも，ジョブを追加・削除する必要はありません。また，セットアップ済みのマシンイメージを基にオートスケールで宛先エージェントを複製させれば，宛先エージェントをセットアップする必要はありません。

同期モードと非同期モードについて

フレキシブルジョブの一斉実行では，同期モードで実行するか，非同期モードで実行するかによって，後続ジョブの処理のタイミングを選べます。同期モードと非同期モードについて次に説明します。

同期モード

一斉配信エージェントがすべての宛先エージェントから実行完了の通知を受け取ったあと，後続ジョブを実行します。

非同期モード

一斉配信エージェントがすべての宛先エージェントへの実行要求を完了した時点で，後続ジョブを実行します。

フレキシブルジョブの一斉実行使用時の注意事項については，「7.8 フレキシブルジョブ使用時の注意事項」を参照してください。

一斉実行を使用する場合に必要な設定については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　21.3.2 ジョブを一斉実行する場合の設定手順」を参照してください。

2.4.12　Web 上の業務システムと連携する（HTTP 接続ジョブを使ったジョブネットの定義例）

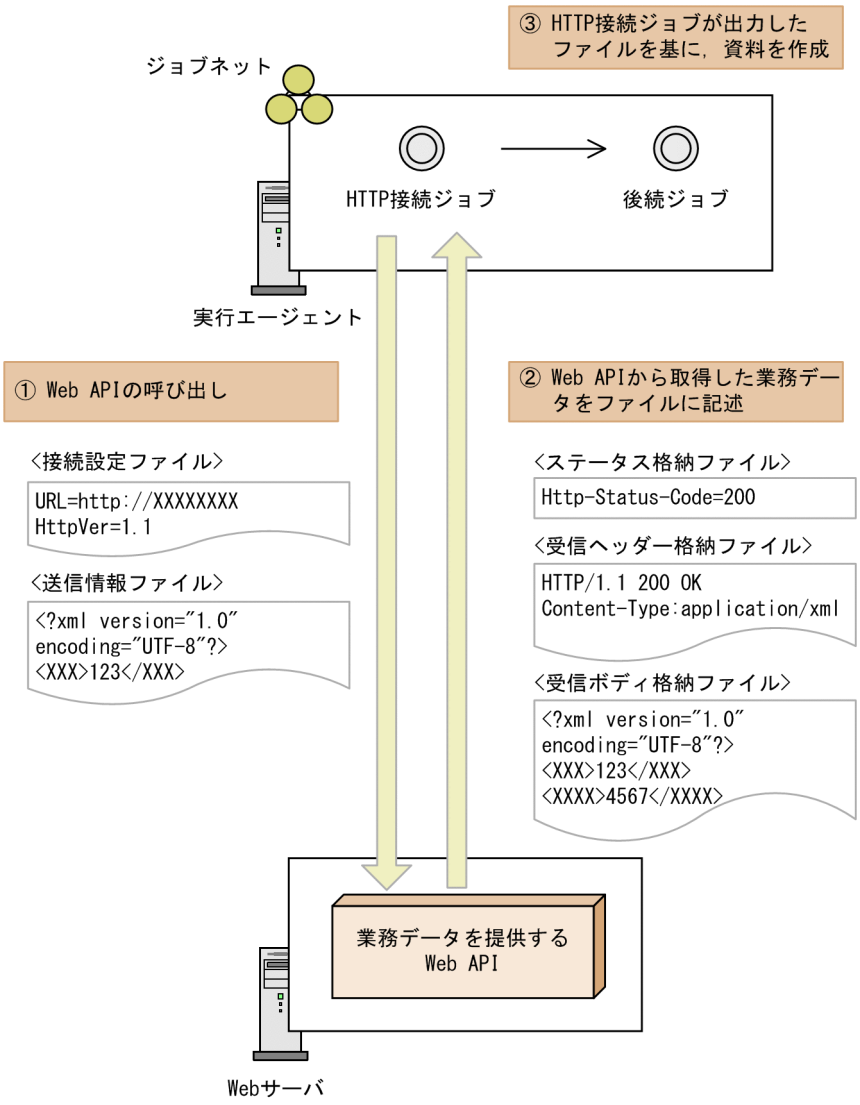
JP1/AJS3 で構築された業務システムと Web 上の業務システムとを連携させるジョブネットの定義には，HTTP 接続ジョブを使用します。

HTTP 接続ジョブを使用すると，HTTP プロトコルでリクエストの送信およびレスポンスの受信ができます。クラウド上や Web サーバ上で公開されている Web API（例えば REST API）を，JP1/AJS3 で呼び出すことができます。

(1)　HTTP 接続ジョブを使ったジョブネットの定義例

「Web 上に公開されている Web API から業務データを取得し，資料を作成する」というジョブネットを，HTTP 接続ジョブを使って定義する例を次の図に示します。

図 2-109　HTTP 接続ジョブを使ったジョブネットの定義例



(2)　HTTP 接続ジョブの定義内容

HTTP 接続ジョブ固有の定義項目について説明します。

HTTP 接続ジョブでは，HTTP リクエストを送信するために，リクエスト種別やリクエスト情報を記載するファイルを定義します。なお，JP1/AJS3 - Manager と JP1/AJS3 - Agent のバージョンが 12-50 以降で，拡張モードを使用している場合，HTTP リクエストの送信時に URL パラメーターとメッセージボディを同時に送信できます。

HTTP リクエストの送信に使用する定義項目を次の表に示します。

表 2-32　HTTP リクエストの送信に使用する定義項目

項番	定義項目	内容
1	リクエスト種別	Web API へのリクエスト種別を指定します。指定できるリクエスト種別を次に示します。

項番	定義項目	内容
1	リクエスト種別	• GET（取得） • POST（追加） • PUT（更新） • DELETE（削除）
2	接続設定ファイル名	Web API を呼び出すための HTTP 接続に関する情報を設定した接続設定ファイルの名称を指定します。
3	送信情報ファイル名※1	Web API を呼び出す HTTP リクエスト情報を設定した送信情報ファイルの名称を指定します。
4	送信情報ファイル名（URL パラメーター）※2	Web API を呼び出す HTTP リクエスト情報の，URL パラメーターを設定した送信情報ファイルの名称を指定します。
5	送信情報ファイル名（メッセージボディ）※2	Web API を呼び出す HTTP リクエスト情報の，メッセージボディを設定した送信情報ファイルの名称を指定します。

注※1

JP1/AJS3 - Manager と JP1/AJS3 - Agent のバージョンが 12-10 以前の場合，または拡張モードを使用しない場合に指定するファイルです。

注※2

JP1/AJS3 - Manager と JP1/AJS3 - Agent のバージョンが 12-50 以降で，拡張モードを使用する場合に指定するファイルです。

また，HTTP レスポンスを受信するためには，受信データを記述するファイル名や受信データの格納方式を定義します。

HTTP レスポンスの受信に使用する定義項目を次の表に示します。

表 2-33　HTTP レスポンスの受信に使用する定義項目

項番	定義項目	内容
1	ステータス格納ファイル名	Web API から取得した受信データの HTTP ステータスコードを記述させるステータス格納ファイルの名称を指定します。
2	受信データ格納方式	Web API から取得した受信データのヘッダーとボディを同じファイルに出力するか，別のファイルに出力するかを指定します。 ヘッダーとボディを同じファイルに出力する場合，受信データのボディ部分も受信ヘッダー格納ファイルに記述されます。
3	受信ヘッダー格納ファイル名	Web API から取得した受信データのヘッダー部分を記述させる受信ヘッダー格納ファイルの名称を指定します。
4	受信ボディ格納ファイル名	Web API から取得した受信データのボディ部分を記述させる受信ボディ格納ファイルの名称を指定します。

なお，HTTP レスポンスでチャンク形式のデータを受信する場合，データをデコードしてファイルに出力できます。チャンク形式のデータをデコードしたい場合は，環境設定パラメーターTRANSFER_DECODING に

「y」を指定してください。環境設定パラメーターTRANSFER_DECODINGの詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.14.2(1) TRANSFER_DECODING」を参照してください。

HTTP接続ジョブで使用するファイルについては，「付録C HTTP接続ジョブで使用するファイル」を参照してください。

また，受信するHTTPステータスコードの値ごとに，ジョブの終了コードを0～9で指定できます。終了コードを指定しない場合はデフォルトの終了コードになります。ジョブの終了コード一つに対して，複数のHTTPステータスコードを指定できます。ただし，一つのHTTPステータスコードを複数のジョブの終了コードに対して指定することはできません。ジョブの終了コードの詳細については，「7.10.3 ジョブの終了コードを確認する」を参照してください。

(3)　HTTP接続ジョブが設定するHTTPヘッダー

HTTP接続ジョブでは，リクエスト種別に応じてHTTPヘッダーが自動的に設定されます。自動的に設定されるHTTPヘッダーを次の表に示します。

表2-34　HTTP接続ジョブが設定するHTTPヘッダー

項番	リクエスト種別	HTTPヘッダー	設定内容
1	GET	User-Agent	「JP1/AJS3 11-00 HTTP Connection Job」固定です。
2		Host	接続設定ファイルで指定した内容を基に，接続先のホスト名およびポート番号を設定します。
3		Accept	「*/*」固定です。
4	• POST • PUT • DELETE	User-Agent	「JP1/AJS3 11-00 HTTP Connection Job」固定です。
5		Host	接続設定ファイルで指定した内容を基に，接続先のホスト名およびポート番号を設定します。
6		Accept	「*/*」固定です。
7		Content-Length	HTTPボディのデータサイズを算出し，設定します。 送信情報ファイルが指定されていない，または送信情報ファイルの中身が存在しない場合は，このヘッダーは設定されません。
8		Expect	次の条件をどちらも満たす場合に，「100-continue」を設定します。 • 接続設定ファイルで指定しているHTTPバージョンが，1.1の場合 • 送信情報ファイルが指定されていて，かつ送信情報ファイルにリクエスト情報が記載されている場合

HTTP接続ジョブで自動的に設定するHTTPヘッダーに追加でヘッダーを設定したい場合は，接続設定ファイルの「Header=追加ヘッダー」で指定します。詳細については，「付録C.1 接続設定ファイル」を参照してください。

(4) HTTP 接続ジョブのセキュリティ対策について

HTTP 接続ジョブで使用する HTTPS 通信とサーバ認証について説明します。

(a) HTTPS 通信

HTTP 接続ジョブは，HTTPS 通信ができます。HTTP 接続ジョブの HTTPS の対応について，次の表に示します。

表 2-35 HTTP 接続ジョブの HTTPS の対応について

項番	項目	内容
1	通信プロトコル	TLSv1.0，TLSv1.1 および TLSv1.2 に対応しています。
2	サーバ認証	対応しています。
3	クライアント認証	対応していません。
4	ホスト名確認	対応しています。 接続設定ファイルの URL に指定されたホスト名と，サーバから返却されたサーバ証明書に含まれる CN（Common Name）または SAN（subject Alt Name）が一致するかどうかを検証します。
5	証明書失効リスト確認	対応しています。 証明書の失効リストを確認する場合，接続設定ファイルで証明書失効リストファイル名を指定する必要があります。

接続設定ファイルの詳細については，「付録 C.1 接続設定ファイル」を参照してください。

(b) サーバ認証

サーバ認証には，Basic 認証を使用できます。

また，接続設定ファイルの追加ヘッダーを設定すれば，Basic 認証以外の認証方式も使用できます。

接続設定ファイルの詳細については，「付録 C.1 接続設定ファイル」を参照してください。

(5) HTTP 接続ジョブの出力情報

ジョブの実行終了時に，HTTP 接続ジョブが標準出力ファイルに出力する情報を次に示します。

```
Http-Status-Code=HTTPステータスの値※1
Http-Status-File=ステータス格納ファイル名
Http-Response-Header=受信ヘッダー格納ファイル名
Http-Response-Body=受信ボディ格納ファイル名※2
```

注※1

HTTP ステータスコードが 1 以上の場合は，HTTP ステータスが出力されます。HTTP ステータスコードが 0 以下の場合は，JP1/AJS3 が設定する保守情報を出力します。

注※2

HTTP 接続ジョブの定義項目の受信データ格納方式で，ヘッダーとボディを同じファイルに出力すると指定した場合，「Http-Response-Body=」だけ出力され，「**受信ボディ格納ファイル名**」は出力されません。

補足事項

- 標準出力ファイルに出力した情報は，引き継ぎ情報設定ジョブを使用し，後続ジョブに任意の情報を引き継くことができます。
- HTTP 接続ジョブが標準出力ファイルに出力する情報は，終了コードに関係なく必ず 4 行出力されます。HTTP 接続ジョブが異常終了すると，値（「=」以降の値）が出力されない場合があります。

(6) 注意事項

- HTTP 接続ジョブが実行ホストとして指定できるマネージャーホストおよびエージェントホストの OS は，Windows と Linux だけです。
- HTTP 接続ジョブが強制終了またはタイムアウトによって異常終了した場合は，ジョブ自体は終了状態になりますが，Web API から実行した処理は終了されません。Web API から実行した処理も取り消したい場合は，API の仕様に従って取り消してください。
- HTTP 接続ジョブで使用する次のファイルに，セキュリティ情報が含まれる場合は，ユーザー責任で管理するようにしてください。
  - 接続設定ファイル
  - 送信情報ファイル
  - 受信ヘッダー格納ファイル
  - 受信ボディ格納ファイル

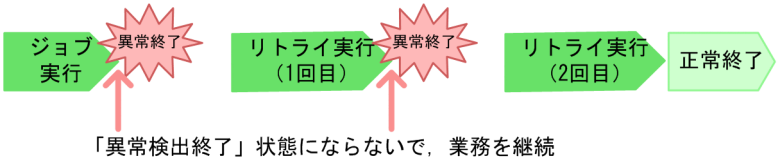
2.4.13 ジョブの異常終了時に自動でリトライする

ジョブに定義した実行ファイルが異常終了した場合に，ジョブをリトライすることで一時的なエラーを回復できることがあります。リトライによって回復できるジョブには，ジョブの異常終了時に自動的にジョブをリトライさせるように定義することで，実行ファイルに一時的なエラーが発生しても業務を継続できます。

ジョブに指定した実行ファイルが異常終了したときに，自動的にリトライすることを**自動リトライ**といい，自動リトライによってジョブを実行することを**リトライ実行**といいます。

自動リトライした場合の，ジョブの動作を次の図に示します。

図 2-110　自動リトライした場合のジョブの動作



自動リトライする場合，ジョブに定義した実行ファイルが異常終了しても，ジョブは「異常検出終了」状態になりません。一定の間隔を待ってから，ジョブが自動リトライされます。

(1)　自動リトライの概要

自動リトライの概要について説明します。

(a)　自動リトライの条件

次の条件を満たすジョブの実行ファイルがエラーになった場合，ジョブは「異常検出終了」状態にならないで，自動リトライします。

- ［終了判定］に［しきい値による判定］を指定している。
- ［異常終了時リトライ］に［する］を指定している。
  ［異常終了時リトライ］を指定できるのは，次のジョブです。
  - UNIX ジョブ
  - PC ジョブ
  - QUEUE ジョブ
  - フレキシブルジョブ
  - HTTP 接続ジョブ
  - 標準カスタムジョブ
  - カスタム PC ジョブ
  - カスタム UNIX ジョブ

注意事項

自動リトライに関する設定は，JP1/AJS3 - View および JP1/AJS3 - Manager のバージョンが 10-00 以降の場合に使用できます。ただし，10-00 以降の JP1/AJS3 - Manager を使用していても，データベース構成が互換用 ISAM 構成の場合は使用できません。

(b)　自動リトライの実行方法に関する設定

自動リトライの実行方法に関する設定を，**リトライ設定**と呼びます。

リトライ設定の各項目について，次に示します。

表 2-36　自動リトライの実行方法に関する設定項目

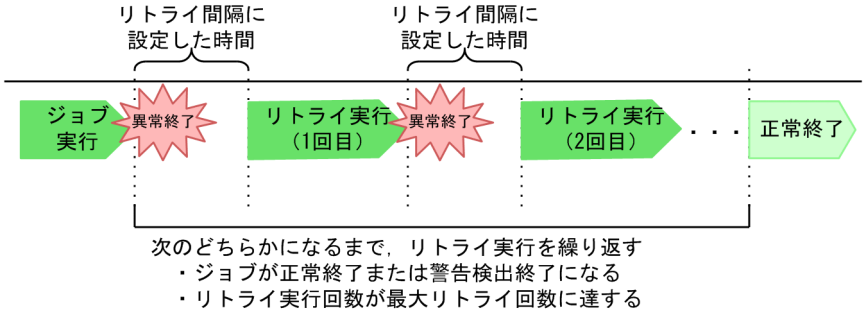
項番	設定項目	設定内容
1	異常終了時リトライ	ジョブに指定した実行ファイルがエラーになったときに，自動リトライするかどうかを指定します。
2	終了コード	リトライ実行する終了コードの範囲を指定します。
3	最大リトライ回数	リトライ実行する最大回数を指定します。
4	リトライ間隔	ジョブの実行ファイルがエラーになってから，リトライ実行するまでの間隔を指定します。

注意事項

終了コードは，必要最低限の範囲を設定してください。自動リトライする終了コードの範囲を必要以上に広く設定すると，リトライ実行しても回復する見込みのない終了コードもリトライ実行し，ジョブの実行数が増加してジョブ実行性能に影響を与えるおそれがあります。

リトライ設定をしているジョブが異常終了したときの動作を，次の図に示します。

図 2-111　リトライ設定をしているジョブが異常終了したときの動作



リトライ設定をしているジョブに指定した実行ファイルが異常終了すると，リトライ間隔で設定した時間が経過してから，リトライ実行されます。リトライ実行は，ジョブが正常終了または警告終了するか，もしくはリトライ実行回数が最大リトライ回数に達するまで繰り返されます。

リトライ設定の内容は，JP1/AJS3 - View の画面およびコマンドで確認できます。確認できる JP1/AJS3 - View の画面およびコマンドと，確認できるリトライ設定を次に示します。

表 2-37　リトライ設定を確認できる JP1/AJS3 - View の画面およびコマンド

項番	確認できる JP1/AJS3 - View の画面およびコマンド	確認できるリトライ設定
1	［ジョブネットエディタ］ウィンドウ	異常終了時リトライ
2	［検索］ウィンドウ	• 最大リトライ回数 • リトライ間隔
3	ajsprint コマンド	すべてのリトライ設定

(c)　自動リトライの実行状況に関する情報

自動リトライの実行状況に関する次の情報を，**リトライ情報**と呼びます。

- リトライ状態
- リトライ実行回数
- リトライ登録日時
- リトライ開始日時

リトライ情報について，次に説明します。

リトライ状態

自動リトライが実行された場合の，自動リトライの進行状況です。

リトライ状態を次に示します。

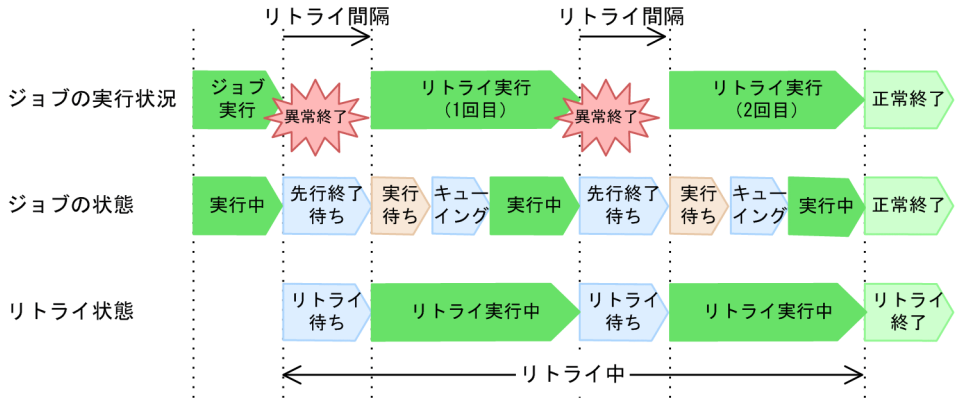
表 2-38　リトライ状態一覧

項番	リトライ状態	リトライ状態の意味	対応するジョブの状態
1	リトライ待ち	ジョブの実行ファイルがエラーになって，リトライ間隔に設定した時間が経過するのを待っている。	• 先行終了待ち • 保留中
2	リトライ実行中	自動リトライによって，ジョブが「実行待ち」，「キューイング」，または「実行中」状態である。	• 実行待ち • キューイング • 実行中
3	リトライ終了	自動リトライが終了した。	• 正常終了 • 警告検出終了 • 異常検出終了 • 起動失敗 • 終了状態不明 • 計画未実行 など終了している状態

「リトライ待ち」，および「リトライ実行中」の状態をまとめて**リトライ中**といいます。

自動リトライが実行されたときの，状態遷移の例を次の図に示します。

図 2-112　自動リトライが実行されたときの状態遷移



リトライ設定をしているジョブに指定した実行ファイルが異常終了すると，リトライ間隔で設定した時間，リトライ状態は「リトライ待ち」になります。このとき，ジョブは「先行終了待ち」状態になり，「異常検出終了」状態にはなりません。リトライ間隔で設定した時間が経過したら，リトライ状態は「リトライ実行中」になります。

リトライ実行回数が最大リトライ回数で設定した回数に達する前に，ジョブに指定した実行ファイルが正常終了または警告終了した場合，ジョブは「正常終了」または「警告検出終了」状態になります。

なお，最大リトライ回数で設定した回数まで実行しても，ジョブに指定した実行ファイルが正常終了または警告終了しない場合は，ジョブは「異常検出終了」状態になります。

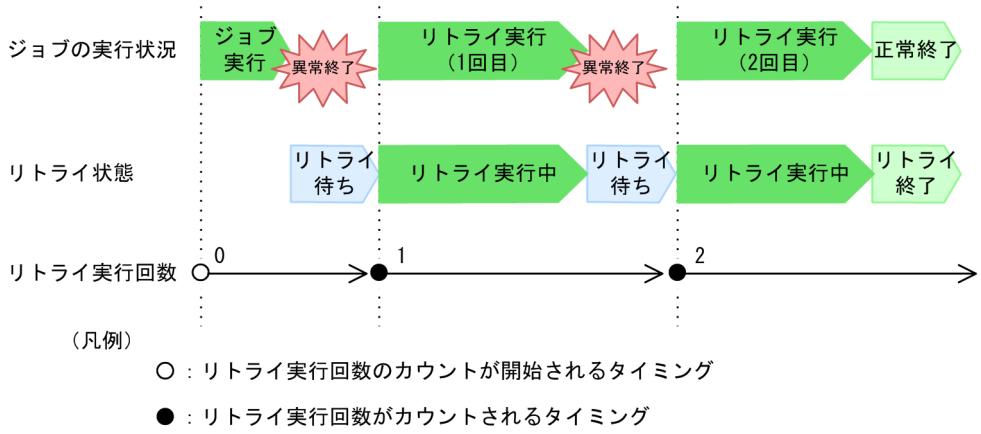
ジョブが「正常終了」，「警告検出終了」，または「異常検出終了」状態になると，リトライ状態は「リトライ終了」になります。

リトライ実行回数

リトライ実行した回数です。

リトライ実行回数がカウントされるタイミングを，次の図に示します。

図 2-113　リトライ実行回数のカウント



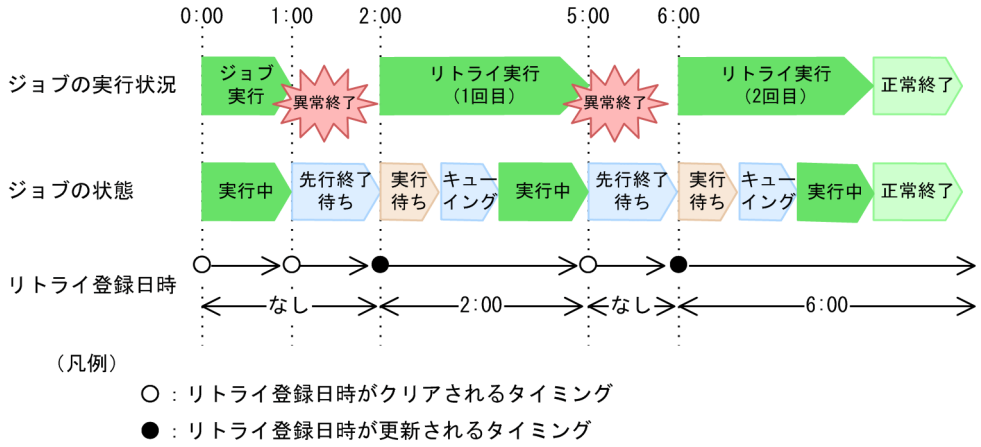
リトライ登録日時

リトライ実行して，ジョブが「実行待ち」状態になった日時です。

リトライ登録日時は，ジョブに指定した実行ファイルが異常終了して「先行終了待ち」状態になるといったんクリアされ，「実行待ち」状態になるとその日時に更新されます。複数回リトライ実行された場合は，最後にジョブが「実行待ち」状態になった日時に更新されます。

リトライ登録日時が更新されるタイミングを，次の図に示します。

図 2-114　リトライ登録日時の更新



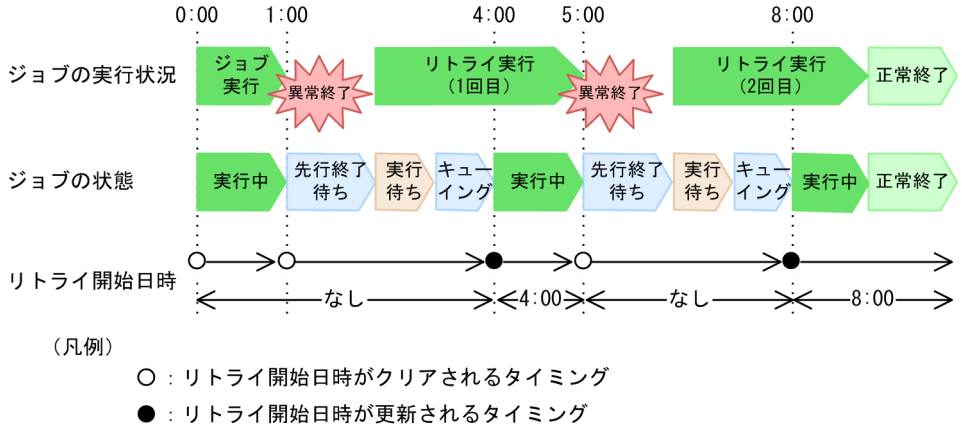
リトライ開始日時

リトライ実行して，ジョブが「実行中」状態になった日時です。

リトライ開始日時は，ジョブに指定した実行ファイルが異常終了して「先行終了待ち」状態になるといったんクリアされ，「実行中」状態になるとその日時に更新されます。複数回リトライ実行された場合は，最後にジョブが「実行中」状態になった日時に更新されます。

リトライ開始日時が更新されるタイミングを，次の図に示します。

図 2-115　リトライ開始日時の更新



リトライ情報は，JP1/AJS3 - View の画面およびコマンドで確認できます。確認できる JP1/AJS3 - View の画面およびコマンドと，確認できるリトライ情報を次に示します。

表 2-39　リトライ情報を確認できる JP1/AJS3 - View の画面およびコマンド

項番	確認できる JP1/AJS3 - View の画面およびコマンド	確認できるリトライ情報
1	• ［モニタ詳細］ダイアログボックス • ［詳細スケジュール］ダイアログボックス • ajsshow コマンド	すべてのリトライ情報
2	• ［デイリースケジュール］ウィンドウ	• リトライ状態

項番	確認できる JP1/AJS3 - View の画面およびコマンド	確認できるリトライ情報
2	• ［マンスリースケジュール］ウィンドウ • ［ジョブネットモニタ］ウィンドウ	• リトライ実行回数
3	［検索］ウィンドウ	リトライ実行回数

複数回リトライ実行された場合，JP1/AJS3 - View の画面およびajsshow コマンドで確認できるのは，最後のリトライ実行の結果です。リトライ実行ごとの結果を知りたい場合は，JP1 イベント，スケジューラーログ，または［実行結果詳細］ダイアログボックスで確認してください。

(2)　リトライ設定のあるジョブの監視

ここでは，リトライ設定のあるジョブの，打ち切り時間および終了遅延監視について説明します。

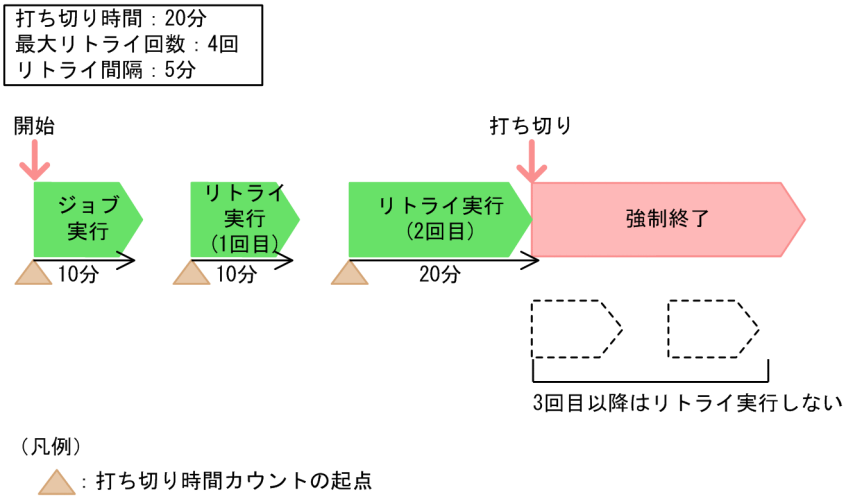
(a)　リトライ中の打ち切り時間

打ち切り時間とリトライ設定の両方を設定した場合，打ち切り時間を監視するための経過時間は，リトライ実行ごとにカウントし直されます。

複数回のリトライ実行を含む，ジョブ全体の経過時間を監視したい場合は，ジョブネットの実行所要時間による終了遅延監視で監視してください。

リトライ中の打ち切り時間の監視を，次の図に示します。

図 2-116　リトライ中の打ち切り時間監視



この例では，1 回目のリトライ実行中に，ジョブの実行開始から 20 分が経過しています。しかし，打ち切り時間を監視するための経過時間は，リトライ実行開始時点からカウントし直されるため，「強制終了」状態にはなりません。2 回目のリトライ実行中にリトライ実行開始から 20 分が経過すると，「強制終了」状態になります。

なお，打ち切り時間が経過してジョブが「強制終了」状態になった場合，最大リトライ回数に達していなくても，それ以降はリトライ実行しません。

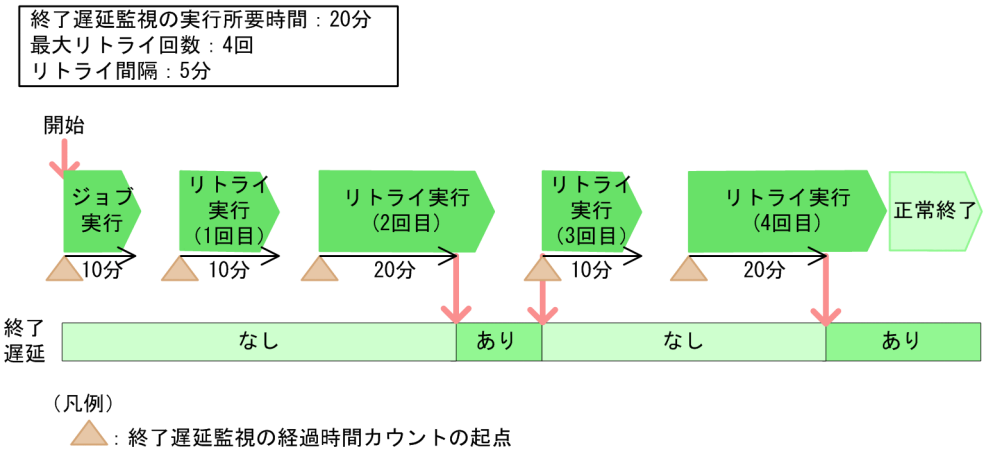
(b)　リトライ中の終了遅延監視

終了遅延監視とリトライ設定の両方を設定した場合，終了遅延を監視するための経過時間はリトライ実行ごとにカウントし直されます。

複数回のリトライ実行を含む，ジョブ全体の経過時間を監視したい場合は，ジョブネットの実行所要時間による終了遅延監視で監視してください。

リトライ中の終了遅延監視を，次の図に示します。

図 2-117　リトライ中の終了遅延監視



この例では，1 回目のリトライ実行中に，ジョブの実行開始から 20 分が経過しています。しかし，終了遅延を監視するための経過時間は，リトライ実行開始時点からカウントし直されるため，終了遅延は「あり」にはなりません。2 回目のリトライ実行中にリトライ実行開始から 20 分が経過すると，終了遅延が「あり」になります。終了遅延が「あり」になったあとにリトライ実行されると，終了遅延は「なし」になります。

最後のリトライ実行の結果，終了遅延が「あり」だった場合，ジョブが終了したあとも遅延情報が表示されます。

注意事項

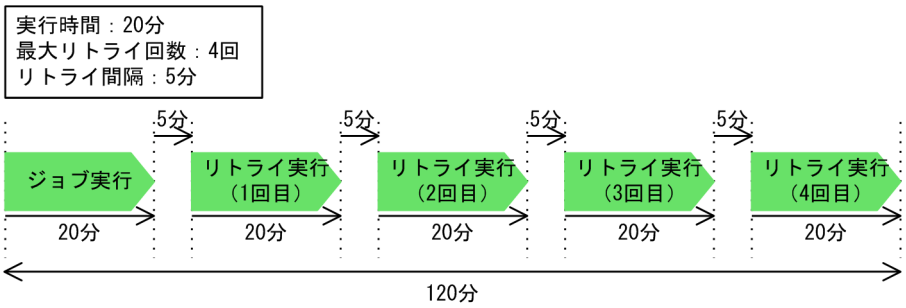
終了遅延が「あり」になったジョブが，次のリトライ実行の実行によって終了遅延が「なし」になっても，上位ジョブネットは「ネスト終了遅延」状態になります。

(3)　リトライ設定のあるジョブの実行シミュレーション

リトライ設定のあるジョブを実行シミュレーションする場合，自動リトライによるリトライ待ちの時間や，リトライ実行の時間を含めてシミュレーションされます。

リトライ設定のあるジョブを実行シミュレーションした例を，次の図に示します。

図 2-118　リトライ設定のあるジョブの実行シミュレーション



この例では，ジョブの実行時間を「20 分」とします。ジョブ定義時には，最大リトライ回数に「4 回」，リトライ間隔に「5 分」を指定しています。この場合のシミュレーション結果は，最初のジョブ実行時間，4 回分のリトライ実行の実行時間，およびそれぞれのリトライ実行前のリトライ間隔をすべて合計した 120 分です。

(4)　リトライ設定のあるユニットの操作

リトライ設定のあるジョブ，リトライ設定のあるジョブの先行ユニット，またはリトライ設定のあるジョブの上位ユニットに対しての操作について説明します。

(a)　リトライ設定のあるジョブまたは先行ユニットの再実行

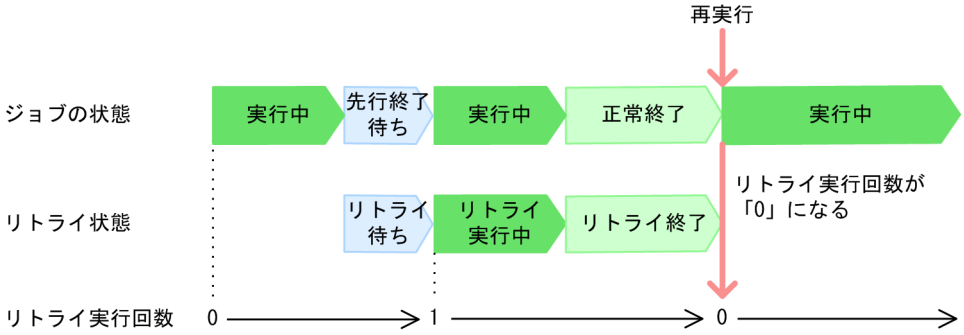
自動リトライが終了したジョブを再実行した場合，リトライ中に先行ジョブから再実行した場合，およびリトライ中に先行ユニットだけを再実行した場合の動作について説明します。

■ 自動リトライが終了したジョブを再実行した場合の動作

自動リトライが終了したジョブを再実行すると，再実行した時点でリトライ実行回数がリセットされます。

自動リトライが終了したジョブを再実行した場合の動作を，次の図に示します。

図 2-119　自動リトライが終了したジョブの再実行



ジョブが再実行されると，リトライ実行回数は「0」になります。

■ リトライ中に先行ユニットから再実行した場合の動作

リトライ中に先行ユニットから再実行すると，実行回数がリセットされ，リトライ中のジョブは「先行終了待ち」状態に遷移して先行ユニットの終了を待ちます。

リトライ中のジョブが「先行終了待ち」状態で先行ユニットの終了を待つタイミングは，先行ユニットを再実行した時点での，リトライ中のジョブの状態によって異なります。

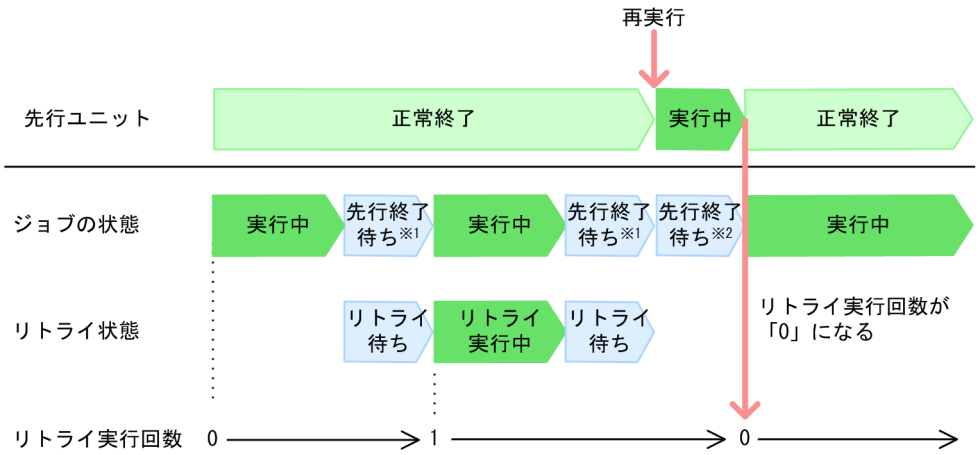
リトライ中に先行ユニットから再実行したときの動作を説明します。

リトライ中のジョブが「先行終了待ち」または「保留中」状態の場合

リトライ中のジョブは，先行ユニットを再実行した時点で「先行終了待ち」状態になり，先行ユニットが終了するのを待ちます。先行ユニットが終了したら，リトライ実行回数はリセットされ，再度ジョブが実行されます。

リトライ中のジョブが「先行終了待ち」状態の場合に，先行ユニットから再実行したときの動作を次の図に示します。

図 2-120　リトライ中のジョブが「先行終了待ち」状態の場合に先行ユニットから再実行したときの動作



注※1
　リトライ間隔が経過するのを待ちます。
注※2
　先行ユニットが終了するのを待ちます。

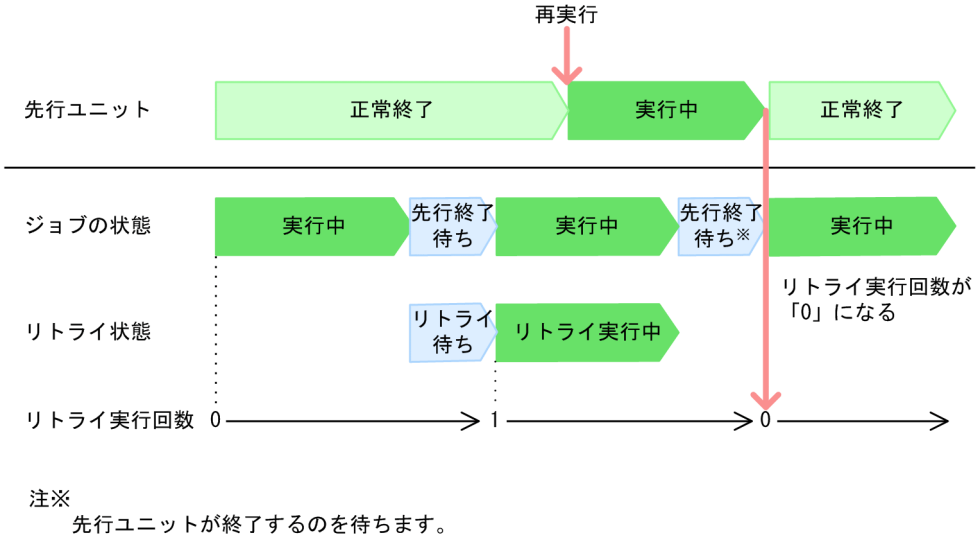
先行ユニットが再実行されると，リトライ実行回数は「0」になります。

リトライ中のジョブが「実行待ち」，「キューイング」，または「実行中」状態の場合

実行中のリトライ実行は最後まで実行され，ジョブが正常終了したかどうかに関係なく，リトライ実行が終了した時点でジョブは「先行終了待ち」状態になります。そのあと，先行ユニットが終了したら，リトライ実行回数はリセットされ，再度ジョブが実行されます。

リトライ中のジョブが「実行中」状態の場合に，先行ユニットから再実行したときの動作を次の図に示します。

図 2-121　リトライ中のジョブが「実行中」状態の場合に先行ユニットから再実行したときの動作



先行ユニットが再実行されると，リトライ実行回数は「0」になります。

■ リトライ中に先行ユニットだけ再実行した場合の動作

リトライ中に先行ユニットだけを再実行すると，リトライ実行が終了してから「先行終了待ち」状態に遷移してリトライ間隔が経過するのを待ちます。リトライ間隔が経過しても先行ユニットが終了していないときは，先行ユニットの終了を待ち，そのあとリトライ実行回数に達するまでリトライ実行されます。

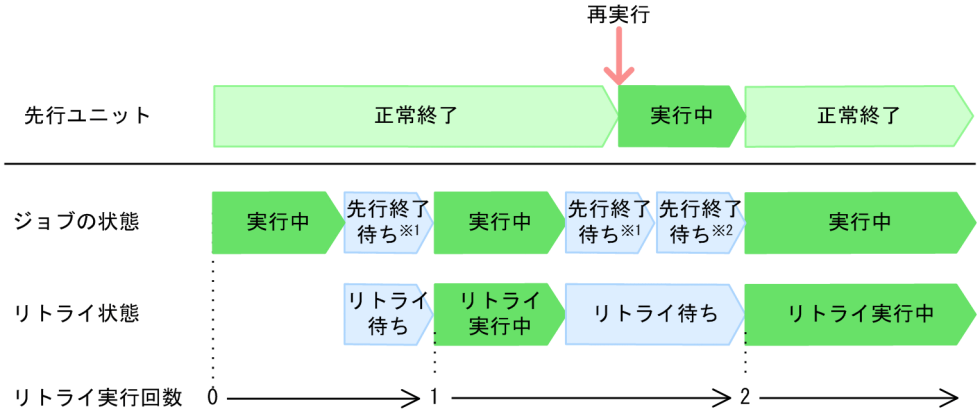
リトライ中に先行ユニットだけ再実行したときの動作を説明します。

リトライ中のジョブが「先行終了待ち」または「保留中」状態の場合

リトライ中のジョブは，リトライ間隔が経過したら先行ユニットが終了するのを待ちます。そのあと，先行ユニットが終了したらリトライ実行されます。

リトライ中のジョブが「先行終了待ち」状態の場合に，先行ユニットだけ再実行したときの動作を次の図に示します。

図 2-122　リトライ中のジョブが「先行終了待ち」状態の場合に先行ユニットだけ再実行したときの動作



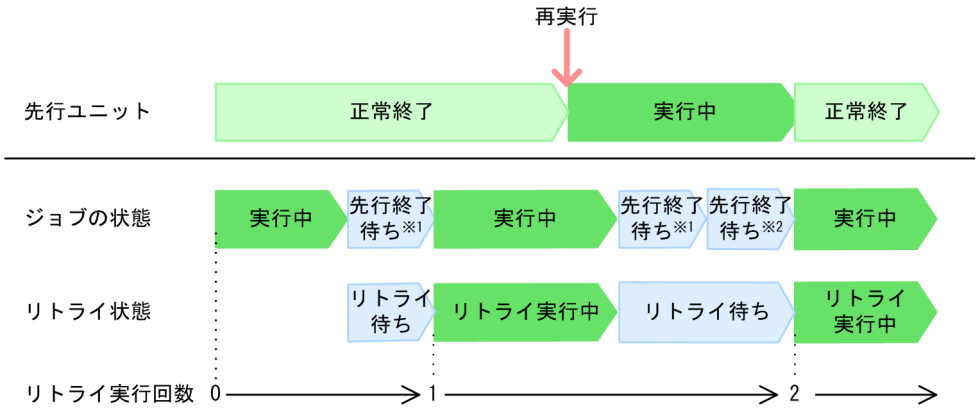
注※1
　リトライ間隔が経過するのを待ちます。
注※2
　先行ユニットが終了するのを待ちます。

リトライ中のジョブが「実行待ち」，「キューイング」，または「実行中」状態の場合

実行中のリトライ実行は最後まで実行されます。実行中のリトライ実行で実行ファイルが異常終了すると，リトライ間隔が経過してから先行ユニットが終了するのを待ちます。そのあと，先行ユニットが終了したらリトライ実行されます。

リトライ中のジョブが「実行中」状態の場合に，先行ユニットだけ再実行したときの動作を次の図に示します。

図 2-123　リトライ中のジョブが「実行中」状態の場合に先行ユニットだけ再実行したときの動作



注※1
　リトライ間隔が経過するのを待ちます。
注※2
　先行ユニットが終了するのを待ちます。

(b)　リトライ中のサスペンド

リトライ中のジョブを含むルートジョブネットは，サスペンド状態にしたり，サスペンド状態を解除したりできます。

サスペンド中は，ジョブが異常終了してもリトライ実行されません。サスペンドを解除すると，リトライ間隔に設定した時間を待ってからリトライ実行されます。

サスペンド操作の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.5.17 ジョブネットの実行登録を解除しないでジョブネットやジョブの定義を変更する」を参照してください。

(c)　リトライ中のジョブを含むルートジョブネットの中断

リトライ中のジョブを含むルートジョブネットを中断した場合，ルートジョブネットは「中断」状態に，リトライ設定のあるジョブは「未実行終了」状態になります。

リトライ中のジョブが「未実行終了」状態になるタイミングは，リトライ中のジョブの状態によって異なります。

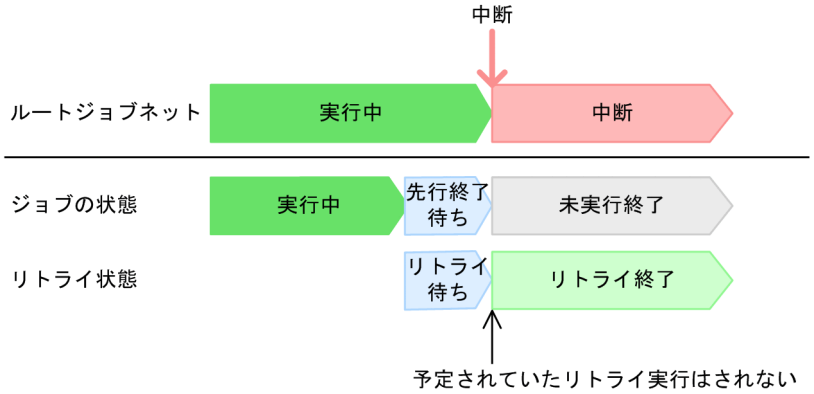
ルートジョブネットを中断した場合の，リトライ中のジョブの動作を次に示します。

リトライ中のジョブが「先行終了待ち」または「保留中」状態の場合

予定されていたリトライ実行はされないまま，ジョブは「未実行終了」状態に，リトライ状態は「リトライ終了」になります。

リトライ中のジョブが「先行終了待ち」状態の場合に，ルートジョブネットを中断したときの動作を，次の図に示します。

図 2-124　リトライ中のジョブが「先行終了待ち」状態の場合にルートジョブネットを中断したときの動作

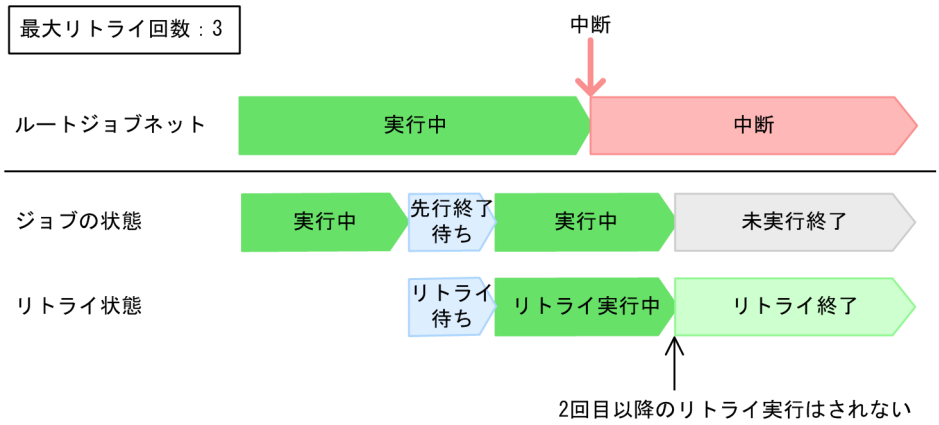


リトライ中のジョブが「実行待ち」，「キューイング」，または「実行中」状態の場合

実行中のリトライ実行は最後まで実行されます。ジョブが正常終了したかどうかに関係なく，終了したジョブは「未実行終了」状態に，リトライ状態は「リトライ終了」になります。

リトライ中のジョブが「実行中」状態の場合に，ルートジョブネットを中断したときの動作を，次の図に示します。

図 2-125　リトライ中のジョブが「実行中」の場合にルートジョブネットを中断したときの動作



リトライ中のジョブを含むルートジョブネットを中断すると，ジョブに定義した実行ファイルのエラーが解消されないまま，ジョブは「未実行終了」状態に，リトライ状態は「リトライ終了」になります。

(d)　リトライ中の状態変更または強制終了

自動リトライによって，状態が「実行中」，「キューイング」，または「実行待ち」（実行先サービスがキューレスに設定されているジョブの場合）になったジョブを状態変更で終了状態にしたり，すでに終了状態のジョブを任意の終了状態に変更したりできます。また，強制終了することもできます。状態変更または強制終了した場合，状態変更後の終了状態や終了コードに関係なく，終了状態になったジョブは自動リトライされません。

(e)　リトライ中のジョブ定義内容変更

リトライ中にジョブの定義内容を変更した場合，環境設定パラメーターUNITDEFINERELOADに「yes」を設定していると，リトライ実行のたびにジョブの定義が再読み込みされます。そのため，リトライ中にジョブの定義を変更すると，変更後に実行されるリトライ実行から，新しい定義内容が有効になります。

定義内容を変更したときの動作については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　7.4 実行登録中にユニット定義情報を変更する」を参照してください。

なお，リトライ状態が「リトライ待ち」のときにリトライ設定を削除した場合，一回だけリトライ実行します。リトライ実行が終了したら，それ以降はリトライ実行しません。

注意事項

リトライ中にジョブの定義内容を変更すると，定義を変更した時点のジョブ状態によって，その後の動作が変わります。意図しない動作をすることがありますので，リトライ中には定義変更しないことを推奨します。

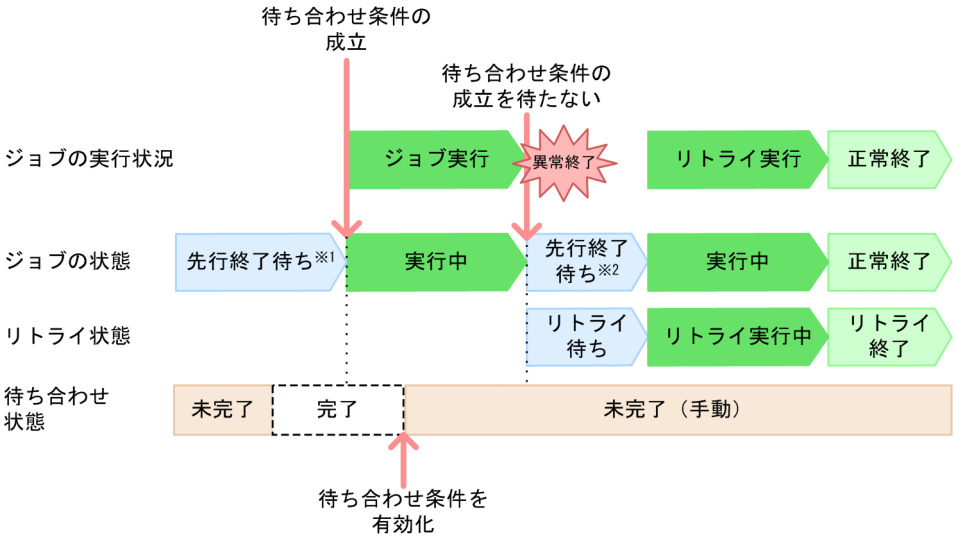
(f)　リトライ中の待ち合わせ

リトライ設定のあるジョブに待ち合わせ条件を設定している場合，一度待ち合わせ状態が「完了」になれば，リトライ実行のときには待ち合わせません。リトライ状態が「リトライ待ち」になる前に，待ち合わ

せ条件を有効にして待ち合わせ状態を「未完了（手動）」に変更しても，待ち合わせ条件の成立を待たないですぐにリトライ間隔の経過を待ちます。

リトライ設定のあるジョブに待ち合わせ条件を設定した場合の動作を，次の図に示します。

図 2-126　リトライ設定のあるジョブに待ち合わせ条件を設定した場合の動作



注※1
　待ち合わせ条件が成立するのを待ちます。
注※2
　リトライ間隔が経過するのを待ちます。

(5)　リトライ実行によって更新される情報

リトライ実行されると，リトライ情報以外にも更新される情報があります。ジョブ実行中にリトライ実行で更新される情報を参照する場合，参照するタイミングによって参照結果が異なることがあるので，影響を考慮してジョブを定義してください。

リトライ実行によって更新される情報を，次に示します。

表 2-40　リトライ実行によって更新される情報一覧

項番	更新情報	更新内容
1	環境変数JP1JobID	リトライ実行時のジョブ ID に更新されます。
2	標準出力ファイル	ジョブの詳細定義で［追加書き］を指定していない場合 　リトライ実行のたびに，ファイルの内容が上書きされます。 ジョブの詳細定義で［追加書き］を指定している場合 　リトライ実行のたびに，ファイルに追加書きされます。
3	標準エラー出力ファイル	ジョブの詳細定義で［追加書き］を指定していない場合 　リトライ実行のたびに，ファイルの内容が上書きされます。 ジョブの詳細定義で［追加書き］を指定している場合 　リトライ実行のたびに，ファイルに追加書きされます。

項番	更新情報	更新内容
4	転送ファイル	リトライ実行のたびに，ファイルは転送されます。
5	実行結果詳細	リトライ実行のたびに，追加書きされます。

注意事項

標準出力ファイルおよび標準エラー出力ファイルは，［追加書き］を指定すると，ファイルのサイズが大きくなり高い負荷が掛かります。そのため，［追加書き］を指定しないことを推奨します。［追加書き］を指定する場合は，出力情報を少なくしたり，定期的にファイルをクリアしたりして運用してください。

(6) リトライ中のスケジューラーサービスの再起動

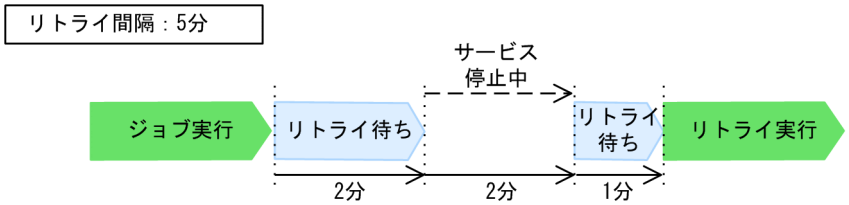
スケジューラーサービスを停止して再起動した場合，スケジューラーサービスの起動モードに応じて，リトライ中のジョブも通常のジョブと同様の状態になります。ただし，次の点が異なります。

- ウォームスタートまたはディザスターリカバリースタートでスケジューラーサービスを起動した場合
  自動リトライを継続しません。起動モードごとのジョブ状態に従って終了状態になります。
- ホットスタートでスケジューラーサービスを起動した場合
  自動リトライを継続します。リトライ間隔およびリトライ実行回数は，次のように扱われます。

リトライ間隔

スケジューラーサービスが停止している期間も，リトライ間隔の経過時間に含みます。例えば，5分間隔でリトライ実行するジョブの場合，「リトライ待ち」になってから2分経過した時点で2分間スケジューラーサービスを停止してホットスタートで再起動すると，再起動してから1分後にリトライ実行を開始します。

図 2-127　ホットスタートでスケジューラーサービスを起動した場合の動作



なお，JP1/AJS3サービスまたはスケジューラーサービスを，リトライ間隔以上の時間停止させてからホットスタートで再起動した場合，スケジューラーサービスの起動と同時にジョブが実行を開始します。そのため，一時的にジョブ量が増えて，ジョブ実行性能が一時的に低下するおそれがあります。

リトライ実行回数

リトライ実行回数は，初期化されません。リトライ実行回数を継続して，リトライ実行します。例えば，最大リトライ回数に「5回」を指定しているジョブの場合，リトライ実行が2回終了した時点でスケジューラーサービスを再起動すると，再起動後，最大で残りの3回分のリトライを実行します。

起動モード別のジョブの状態については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　6.2.1 JP1/AJS3 起動時の動作を一時的に変更する」を参照してください。

(7)　自動リトライに関する注意事項

自動リトライを使用する場合の注意事項を，次に示します。

- リモートジョブネット配下のユニットにリトライ設定をした場合で，リモートジョブネットの実行先マネージャーが次のどちらかに該当するとき，エラーになります。
  - データベース構成が互換用 ISAM 構成
  - JP1/AJS3 - Manager のバージョンが 09-50 以前
- スケジューラーサービスのジョブ制限停止中およびジョブ強制終了停止中は，自動リトライは行いません。ジョブは終了コードやリトライ実行回数に関係なく，リトライ実行しないで終了状態になります。実行中のジョブがなくなれば，スケジューラーサービスは停止します。

  スケジューラーサービスの制限停止については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　7.5.2 スケジューラーサービスを停止する」を参照してください。
- ユーザーアプリケーションではなく，OS が終了コードを設定してジョブが終了する場合があります。その場合，自動リトライの対象になるため注意してください。
- キューレスジョブでは，スクリプトファイルがない場合やアクセスできない場合も，自動リトライの対象となります。定義内容の事前チェック機能で，スクリプトファイルにアクセスできることを確認してください。
- キューレスジョブを実行する場合，バージョンが 10-00 以降の JP1/AJS3 - Manager または JP1/AJS3 - Agent で実行してください。バージョンが 09-50 以前だと，何らかの要因でジョブプロセス起動用プロセスが異常終了した場合も，自動リトライの対象となります。

3 運用カレンダーと実行スケジュールの検討

自動化する業務の検討が終了したら，JP1/AJS3 運用上のカレンダーと業務の実行スケジュールについて検討します。

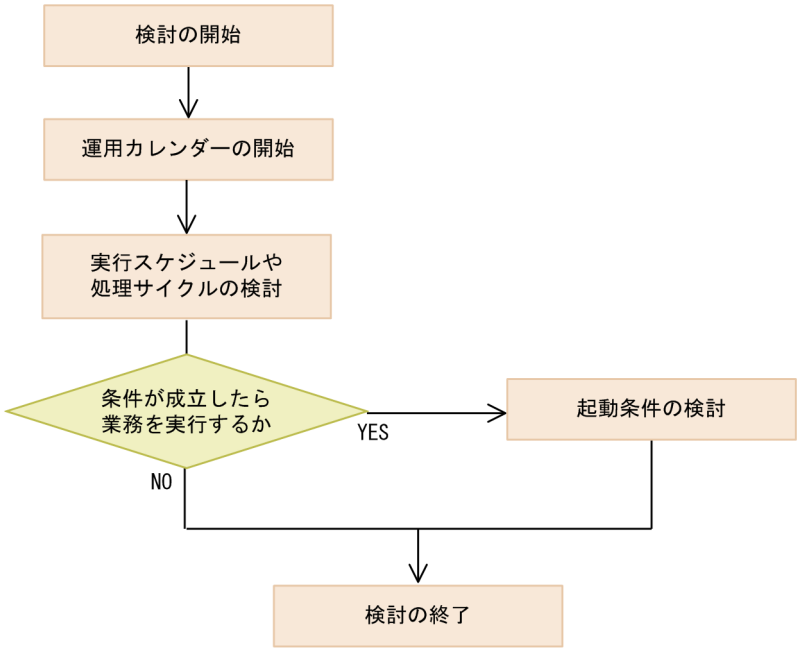
この章では，カレンダーや実行スケジュールの検討項目について説明します。

3.1　カレンダーやスケジュールの検討の流れ

ジョブネットの実行パターンによって，カレンダーやスケジュールの検討の流れが異なります。

ジョブネットの実行パターンを踏まえた，カレンダーやスケジュールの検討の流れを次の図に示します。

図 3-1　カレンダーやスケジュールの検討の流れ



3.2　JP1/AJS3 運用上のカレンダーの検討

暦に日曜日や祝祭日などの休日があるように，JP1/AJS3 でジョブネットを運用するためのカレンダーの作成について検討します。カレンダー作成の検討では，運用日および休業日のほか，基準日，基準時刻についても検討します。

表 3-1　カレンダー作成についての検討項目と内容

検討項目	内容
運用日と休業日	ジョブネットを実行する日（運用日）と実行しない日（休業日）を検討します。
基準日	月の開始日となる日を検討します。 日付または曜日（何週目の何曜日か）で指定します。 （例） 毎月 20 日が給与計算の締め日である場合，基準日を 21 日に設定することで，21 日から翌月の 20 日までを 1 か月分として処理する。
基準時刻	1 日の開始時刻となる時間を検討します。 （例） 基準時刻を 8:00 に設定することで，実際には翌日の 1:00 に実行される処理を当日分の処理として扱う。
排他スケジュール	ほかのジョブネットと自ジョブネットの実行予定が重なったときに自ジョブネットを実行させないようにする場合に検討します。

JP1/AJS3 運用上のカレンダー定義の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.2 JP1/AJS3 運用上のカレンダー定義」を参照してください。

3.3　ジョブネットの実行スケジュールの検討

開始日時や処理サイクルなど，ジョブネットの実行スケジュールを算出するためのルール（**スケジュールルール**）について検討します。

検討項目と内容を次の表に示します。

表 3-2　スケジュールに関する検討項目と内容

検討項目	内容
実行開始日時	ジョブネットを開始する日時を算出するために，次の項目について検討します。 • 開始する日をどのように指定するか（登録日，絶対日，相対日，運用日，休業日） • 開始する時刻をどのように指定するか（絶対時刻，相対時刻） • 基準日や基準時刻を設定するか
処理サイクル	ジョブネットを周期的に実行させるかを検討します。周期的に実行させる場合は，何日（週，月，または年）ごとに実行させるかを検討します。
休業日の振替方法	処理サイクルに従って算出された実行予定日が休業日だった場合，別の日に振り替えるかどうかを検討します。

スケジュールルールの詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.3 スケジュールの定義」を参照してください。また，さまざまな運用パターンに応じたスケジュールの設定方法について，「3.5 さまざまなスケジュール設定」に紹介しています。あわせて参考にしてください。

3.4　起動条件の検討

JP1/AJS3 では，実行する日時をあらかじめ指定できない不定期な業務でも，設定した条件の成立を契機に実行させることができます。この条件を**起動条件**といいます。起動条件にはイベントジョブを指定でき，次のような事象を契機にジョブネットを実行できます。起動条件には，複数のイベントジョブを指定できます。

- あるファイルが更新されたときに実行（ファイル監視ジョブ）
- 特定の JP1 イベントを受信したときに実行（JP1 イベント受信監視ジョブ）
- 指定した時間が経過したときに実行（実行間隔制御ジョブ）
- ログファイルに特定の文字列が出力されたときに実行（ログファイル監視ジョブ）
- メールを受信したときに実行（メール受信監視ジョブ※）

注※　メールシステムとの連携が必要です。

起動条件の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.4 起動条件の定義」を参照してください。

3.4.1　起動条件の検討内容

起動条件によってジョブネットを実行させる場合は，次の内容について検討してください。

(1)　起動条件の有効範囲

起動条件として設定したイベントを監視する範囲（回数または時間）について検討します。起動条件の有効範囲は，ジョブネットのスケジュールルールに設定します。

回数指定

起動条件の監視を開始してから，条件が成立してジョブネットが実行される回数が指定した回数に到達するまで監視を続けます。

時間指定

起動条件の監視を開始してから，指定した時刻に到達するまで監視を続けます。時間の指定には，絶対時刻で指定する方法と相対時刻で指定する方法があります。

(2)　複数のイベントジョブを起動条件に設定する場合の動作

複数のイベントジョブを起動条件として定義している場合，定義したすべてのイベントが発生したときに条件成立とするか（**AND 条件**），どれか一つでも発生すれば条件成立とするか（**OR 条件**）について検討します。

(3) 監視世代と実行世代の多重起動

起動条件を設定したジョブネットを実行すると，起動条件として設定したイベントの発生を監視する世代（**監視世代**）と，条件の成立によって実行される世代（**実行世代**）が生成されます。起動条件を設定したジョブネットでは，監視世代と実行世代のそれぞれの多重起動について検討します。

実行世代の多重起動

複数の条件成立によってジョブネットの実行が重なった場合に，多重に実行させるか，先に実行されたジョブネットの終了を待って次のジョブネットを実行させるかについて検討します。実行世代の多重起動については，ジョブネットに定義した多重起動の設定に基づきます。

また，実行世代を多重起動させない場合は，「(4) 実行世代の滞留」についても検討します。

監視世代の多重起動

起動条件を設定したジョブネットに処理サイクルを設定して実行する場合に検討が必要です。次の世代の開始時刻になった時点で前の監視世代が終了していない場合，監視世代を多重に起動するか，前の監視世代の終了を待つか，または次の監視世代を起動しないか検討します。

(4) 実行世代の滞留

実行世代を多重起動させない場合は，起動条件が成立した実行世代を滞留させるかどうかについて検討します。

- 繰り越して滞留させない

  「起動条件待ち」状態の実行世代を「繰り越し未実行」状態に遷移させて滞留させません。

- 繰り越さないで滞留させる

  「起動条件待ち」状態の実行世代をそのまま滞留させます。

(5) 異常終了したときの動作

起動条件を設定したジョブネットが異常終了したときの動作について検討します。異常終了したときの動作として，次のものを設定できます。

- ジョブネットの実行を開始する

  ジョブネットが異常終了しても，それ以降に条件が成立した場合はジョブネットを実行します。

- ジョブネットの開始を保留する

  ジョブネットが異常終了したら，それ以降に条件が成立した場合はジョブネットの実行を保留にするか，または条件成立を待つ状態のまま実行を抑止します。

- 起動条件の監視を停止する

  ジョブネットが異常終了したら，起動条件の監視を停止します。

(6) その他

あわせて次に示す内容も検討してください。

- ある時間帯にどのくらいの条件が成立するのか※（どのくらいの量のイベントジョブが実行されるのか）。
  注※　イベントジョブの性能見積もりで使用します。

3.5　さまざまなスケジュール設定

ジョブネットのスケジュールなどの機能を使うことで，さまざまな運用に対応できます。

3.5.1　2 日にまたがる業務のスケジュールを立てる

深夜に実行する業務など，2 日にまたがってジョブネットを実行する場合，日付が変わるとスケジュールに支障が出る場合があります。例えば，月曜日から金曜日までの毎日の売り上げ集計を翌日の午前 1:00 に実行し，さらに土曜日を休業日としている場合を考えます。集計の開始時刻を翌日の午前 1:00 とすると，金曜日の分の集計は土曜日の午前 1:00 にスケジュールされますが，土曜日は休業日であるため集計を実行できません。

集計の開始時刻を当日の 25:00 とすることで，土曜日を休業日としている場合にも集計を実行できます。しかし，ネストジョブネットにスケジュールを定義している場合，ルートジョブネットとネストジョブネットで実行開始日が異なることになり，ネストジョブネットが実行されないことがあります。

このような場合には，JP1/AJS3 上での「1 日」の範囲を変更し，2 日にまたがっていた処理が「1 日」の範囲内に収まるようにします。「1 日」の範囲を変更するには，次の 2 とおりの方法があります。

- 「1 日」を 48 時間としてスケジュールを計算する
  ルートジョブネットのスケジュールを 48 時間制にします。
- 「1 日」の始まりの時刻を「0:00」以外にする
  基準時刻に「0:00」以外の時刻を設定します。

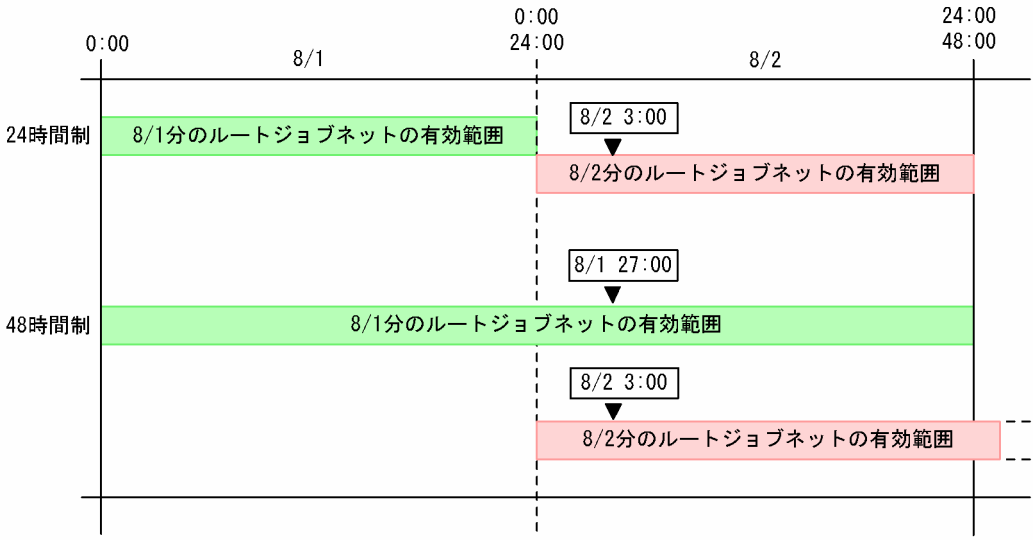
スケジュール構築が容易になるため，ルートジョブネットのスケジュールを 48 時間制にする方法を推奨します。次に，それぞれの方法について説明します。

(1)　48 時間制スケジュールによる日またがり業務の定義

ルートジョブネットのスケジュールを 48 時間制にすると，暦日での翌日 0:00～23:59 は，当日の 24:00～47:59 として扱われます。例えば，土曜日の午前 1 時は，金曜日の 25:00 と指定できます。先ほどの例のように土曜日が休業日の場合も，48 時間制にして実行開始時刻を金曜日の 25:00 と指定すれば，ジョブは実行されます。

24 時間制スケジュールと 48 時間制スケジュールの違いを次の図に示します。

図 3-2　24 時間制スケジュールと 48 時間制スケジュールの違い



48 時間制（基準時刻 0:00）で実行開始時刻を絶対時刻で指定する場合，午前 3 時に実行を開始するジョブは，「8/1 27:00」「8/2 3:00」の 2 とおりに表せます。どちらも実際の時刻は同じですが，実行予定日が異なるため，休業日／運用日，実行予定の振り替えなどが関係する場合，実行されない場合があります。

なお，24 時間制のままでも，24:00～47:59 の時刻を指定することはできます。しかし，24 時間制の場合，例えば「8/1 27:00」と指定すると，「8/2 3:00」と解釈されてスケジュールが設定されます。

(a)　24 時間制から 48 時間制へ変更する

48 時間制スケジュールを使ってジョブネットのスケジュールを設定するには，スケジューラーサービスの環境設定が必要になります。

24 時間制から 48 時間制スケジュールに変更する場合の手順を次に示します。

1. スケジューラーサービスの環境設定パラメーター ROOTJOBNETSCHEDULERANGE の値を「00000030」に変更する。
2. ジョブグループの詳細定義で，基準時刻に「0:00」を指定する。
   基準時刻をデフォルトから変更していない場合は，指定不要です。
3. ジョブネットの実行開始時刻に 24:00 以降の値を指定する。

環境設定パラメーターの設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　4.2 環境設定パラメーターの設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　14.2 環境設定パラメーターの設定」（UNIX の場合）を参照してください。

(b)　48 時間制に変更した場合のスケジュール見直し

ルートジョブネットのスケジュールを 24 時間制から 48 時間制に変更すると，ルートジョブネットのスケジュールの実行動作が変わります。そのため，48 時間制スケジュールに変更した場合は，既存のスケジュー

ルを見直してください。次に示す条件に該当する場合は，ジョブネットのスケジュールルールを再設定してください。

1. ルートジョブネットの実行開始時刻に 24:00 以降の時刻を定義している場合

   ルートジョブネットとネストジョブネットの実行開始日がずれるため，スケジュール定義内容を見直してください。

2. 1.に当てはまるジョブネットと同階層のジョブネット，またはプランニンググループに排他スケジュールを定義している場合

   排他スケジュールの対象となる実行開始日がずれるため，スケジュール定義内容を見直してください。

3. 1.および 2.のジョブネットの下に，上位ジョブネットに依存しないスケジュールが定義されたネストジョブネットがある場合

   ネストジョブネットの実行対象となる実行開始日がずれるため，ネストジョブネットのスケジュール定義内容を見直してください。

4. 基準時刻に 0:00 以外の時刻を設定している場合

   スケジュールルールの次の項目の時間の扱いが変わるため，ルートジョブネットとネストジョブネットのスケジュール定義内容を見直してください。

   - 実行開始日時
   - 開始遅延監視の時刻
   - 終了遅延監視の時刻
   - 起動条件の有効範囲の時間

   なお，基準時刻に 0:00 以外の時刻を設定した場合の時間の扱い方については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.3.2 スケジュール情報の定義」を参照してください。

(2)　基準時刻による日またぎ業務の定義

通常，JP1/AJS3 上での「1 日」は，0:00 から翌日の 0:00 までの 24 時間となっています。「1 日」の始まりの時刻をずらすことで，日付をまたぐ処理を 1 日に収めることができます。例えば，基準時刻を 8:00 とすると，JP1/AJS3 の運用上の「1 日」は，8:00 から翌日の 8:00 までとなります。当日の 0:00 から 7:59 までは前日扱いとなります。

基準時刻を変更すると，実行開始時刻や遅延時刻の設定などのスケジュール定義が複雑になります。例えば，実行開始日時の種別によって，時刻の指定方法が異なります。

基準時刻が 8:00 で，暦日の 8 月 5 日 1:00 を指定する場合の実行開始の指定時刻を次の表に示します。

表 3-3　実行開始時刻の指定

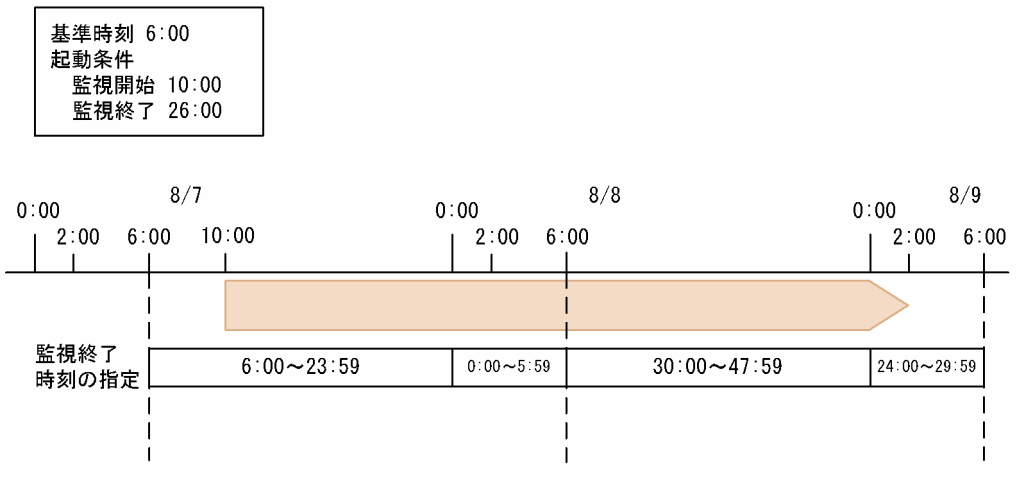
実行開始時刻の種別	実行開始日の種別	指定時刻
絶対時刻	登録日，絶対日，相対日	8/5　1:00（または 8/4　25:00）

実行開始時刻の種別	実行開始日の種別	指定時刻
絶対時刻	運用日，休業日	8/4　1:00
相対時刻	すべて	8/4　17:00

(a)　起動条件を設定した場合

24 時間制のスケジュールで基準時刻を「0:00」以外に設定している場合，ジョブネットに起動条件を設定し，起動条件の終了時刻として 24:00 以降で基準時刻より前の時刻を指定すると，起動条件の終了時刻は翌々日扱いとなります。起動条件の終了時刻が翌々日になる例を次の図に示します。

図 3-3　起動条件の終了時刻が翌々日になる場合



この例で示すように，基準時刻を「0:00」以外に設定している場合は，0:00（24:00）から基準時刻までの時刻の指定が複雑になるため，48 時間制スケジュールの使用を推奨します。

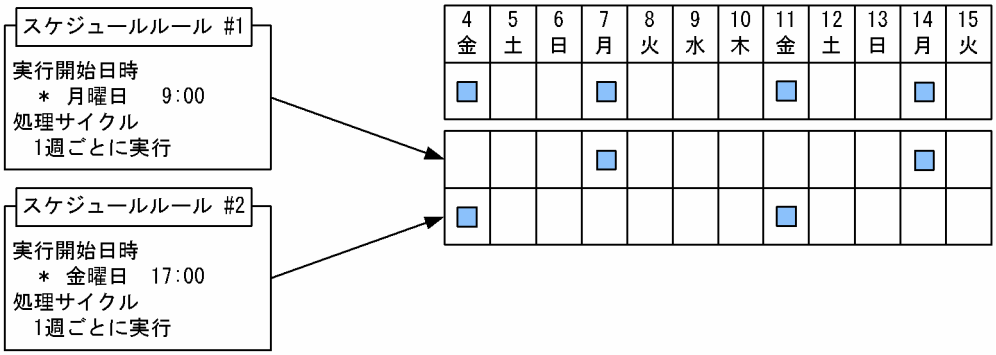
3.5.2　実行開始日時を複数設定する

一つのスケジュールルールでは設定できない複雑なスケジュールでジョブネットを運用する場合，スケジュールルールを複数作成できます。JP1/AJS3 では，一つのジョブネットに対し，最大 144 個のスケジュールルールを設定できます。複数のスケジュールルールを定義した場合，いちばん早い日時のスケジュールルールでスケジュールが設定されます。

例えば，1 日に複数回，決まった時刻にジョブネットを実行する場合や，曜日によって実行時刻を変える場合などは，複数のスケジュールルールを作ることで，スケジュールを定義できます。なお，複数のスケジュールルールを設定した結果として，同じ実行開始日時の実行予定が複数できた場合には，ジョブネットは 1 回だけしか実行されません。

スケジュールルールを複数作成する例を次の図に示します。

図 3-4　スケジュールルールを複数作成する例



この例では，月曜日の 9:00 と金曜日の 17:00 の週 2 回実行するジョブネットのスケジュールを定義しています。

「スケジュールルール#1」では，月曜日分のスケジュールを，「スケジュールルール#2」では，金曜日分のスケジュールを定義しています。これを実行登録すると，二つのスケジュールルールのそれぞれに基づいて実行予定が算出されます。

3.5.3　ジョブネットの一部のジョブにほかのジョブと異なるスケジュールを設定する

ジョブネット中の一部のジョブだけを，ほかとは異なるスケジュールで実行する場合，ネストジョブネットを使います。

ルートジョブネットと同じように，ネストジョブネットにもスケジュールルールやスケジュールオプションを設定できます。ネストジョブネットのスケジュールルールは，ルートジョブネットのスケジュールルールとリンクさせて定義します。ルートジョブネットのスケジュールルール一つに対して，ネストジョブネットの複数のスケジュールルールをリンクさせることができます。

なお，ネストジョブネットにスケジュールを設定しなかった場合は，ルートジョブネットと同じスケジュールでネストジョブネットが実行されます。

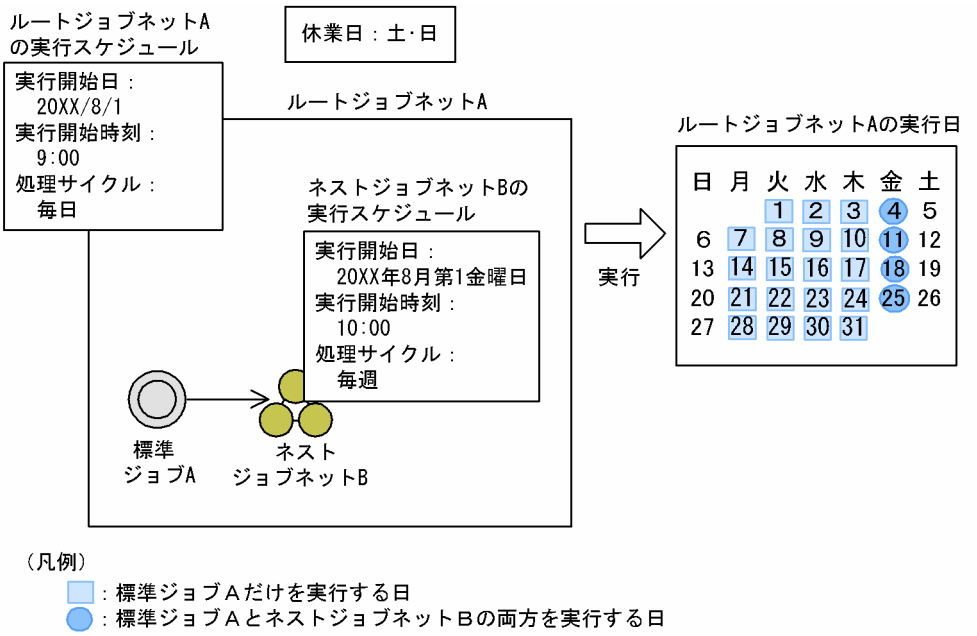
(1)　ネストジョブネットにスケジュールを設定する

ルートジョブネットの中にネストジョブネットを作成し，異なるスケジュールで実行するジョブをネストジョブネットの中に定義します。ネストジョブネットを定義したら，ネストジョブネットにスケジュールルールを設定します。

このように定義してルートジョブネットを実行登録すると，ネストジョブネットは定義されたスケジュールに従って運用されます。ただし，ネストジョブネットは，上位ジョブネットの実行条件が成立しているときだけ実行されます。ネストジョブネットのスケジュールが設定されていても，ルートジョブネットの実行予定がない日には，ネストジョブネットも実行されません。

ネストジョブネットのスケジュールを利用した例を次の図に示します。

図 3-5　ネストジョブネットのスケジュールを利用した例



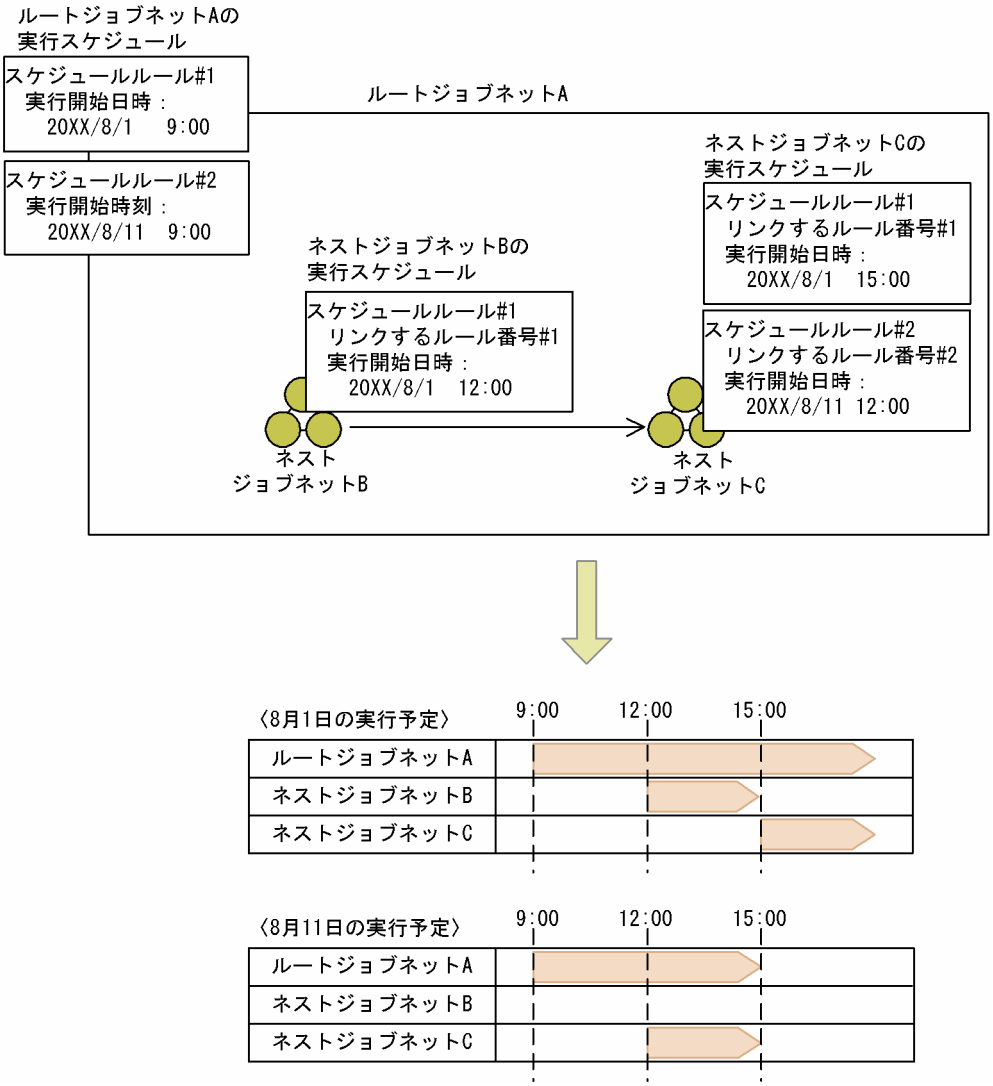
この例では，ネストジョブネット B は，金曜日にだけ実行するようにスケジュールが定義されています。そのため，月曜日～木曜日は，ジョブネット A が実行されても，ネストジョブネット B は実行されません。

(2)　ルートジョブネットのスケジュールルールとリンクさせる

ネストジョブネットのスケジュールルールは，ルートジョブネットのスケジュールルールとリンクさせて定義します。リンクしたルートジョブネットのスケジュールルールが有効になったときに，ネストジョブネットのスケジュールが有効になります。

スケジュールルールのリンクの例を次の図に示します。

図 3-6　ネストジョブネットとルートジョブネットのスケジュールルールをリンクする



この例では，ルートジョブネットのスケジュールルール#2 に対して，ネストジョブネット B のスケジュールルールはリンクしていません。そのため，ルートジョブネットのスケジュールルール#2 に従って実行される 8 月 11 日には，ネストジョブネット B は実行されません。

ネストジョブネットにスケジュールを定義するときの注意事項を次に示します。

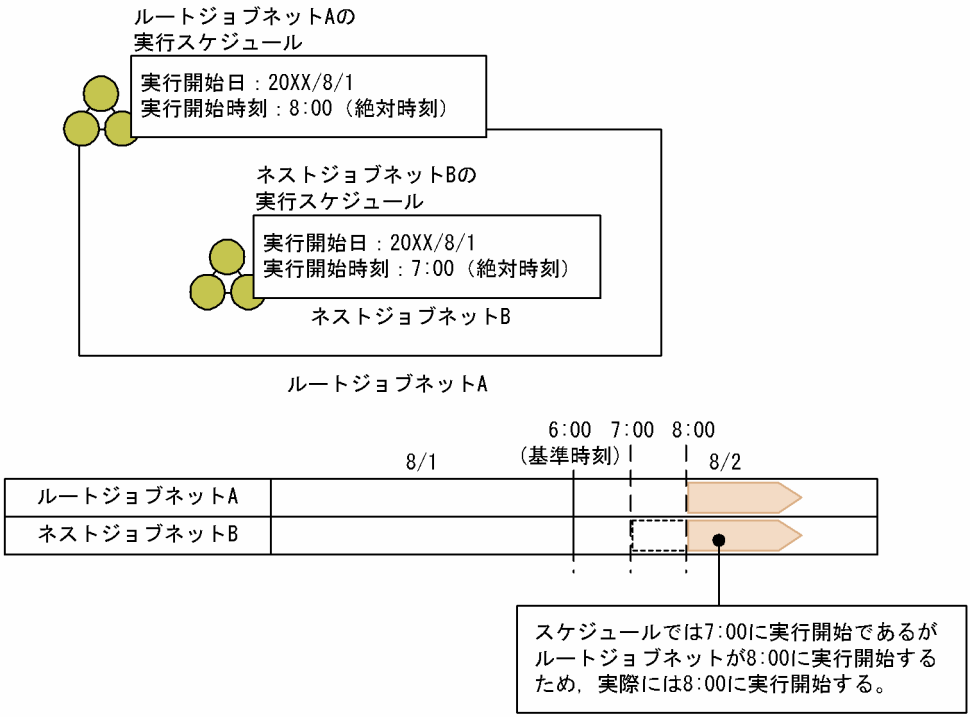
- 上位ジョブネットのスケジュールとネストジョブネットのスケジュールに重なる部分がまったくない場合，ネストジョブネットは実行されません。ネストジョブネットをコピーして使う場合などには，上位スケジュールを確認してから定義してください。
- ネストジョブネットに日またがりのスケジュールを定義することもできます。この場合，ネストジョブネットの実行開始時刻は 24:00～47:59 の間で指定します。

  例えば，ルートジョブネットの実行開始日時が「20XX/08/1　23:00」で，日付が変わった午前 2 時をネストジョブネットの実行開始時刻とする場合，「20XX/08/1　26:00」と指定します。これは，ルートジョブネットとネストジョブネットの実行開始日を同じ日にするためです。もし，ネストジョブネットの実行開始日時を「20XX/08/2　2:00」と指定すると，ルートジョブネットとネストジョブネットの実行開始日が異なるため，ネストジョブネットが実行されません。

- ルートジョブネットのスケジュールルール一つに対して，ルートジョブネットと実行開始日が重なるネストジョブネットのスケジュールルールを複数リンクさせた場合は，開始時刻が最も早いスケジュールルールが有効になります。また，実行予定日時が重なった場合は，重なったスケジュールルールの中で最も小さいルール番号のスケジュールルールが有効になります。有効になったスケジュールルール以外は無効になるため，1 日に複数回ジョブネットを実行する場合は，「3.5.4 1 日のうちに同じジョブネットを何回か定期的に実行する（サイクルジョブ定義）」を参照してください。
- ネストジョブネットの開始時刻がルートジョブネットの開始時刻より早い場合，ネストジョブネットの開始日時が基準時刻より前になっていると，ルートジョブネットとネストジョブネットの実行開始日が異なるため，ネストジョブネットが実行されません。

  ネストジョブネットの開始日時がルートジョブネットの開始日時より早い場合にネストジョブネットが実行される例を次の図に示します。

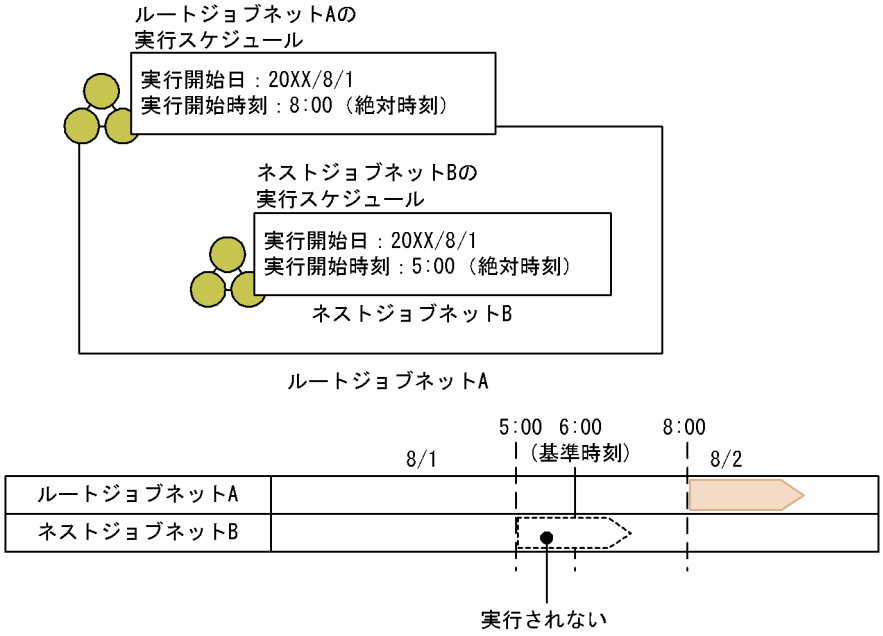
図 3-7　ネストジョブネットの開始日時がルートジョブネットの開始日時より早い場合（ネストジョブネットが実行される例）



この場合，ネストジョブネットの開始日時（7:00）が基準時刻（6:00）よりあとであり，ルートジョブネットとネストジョブネットの実行開始日が同じであるため，ネストジョブネットは実行されます。

ネストジョブネットの開始日時がルートジョブネットの開始日時より早い場合にネストジョブネットが実行されない例を次の図に示します。

図 3-8　ネストジョブネットの開始日時がルートジョブネットの開始日時より早い場合（ネストジョブネットが実行されない例）

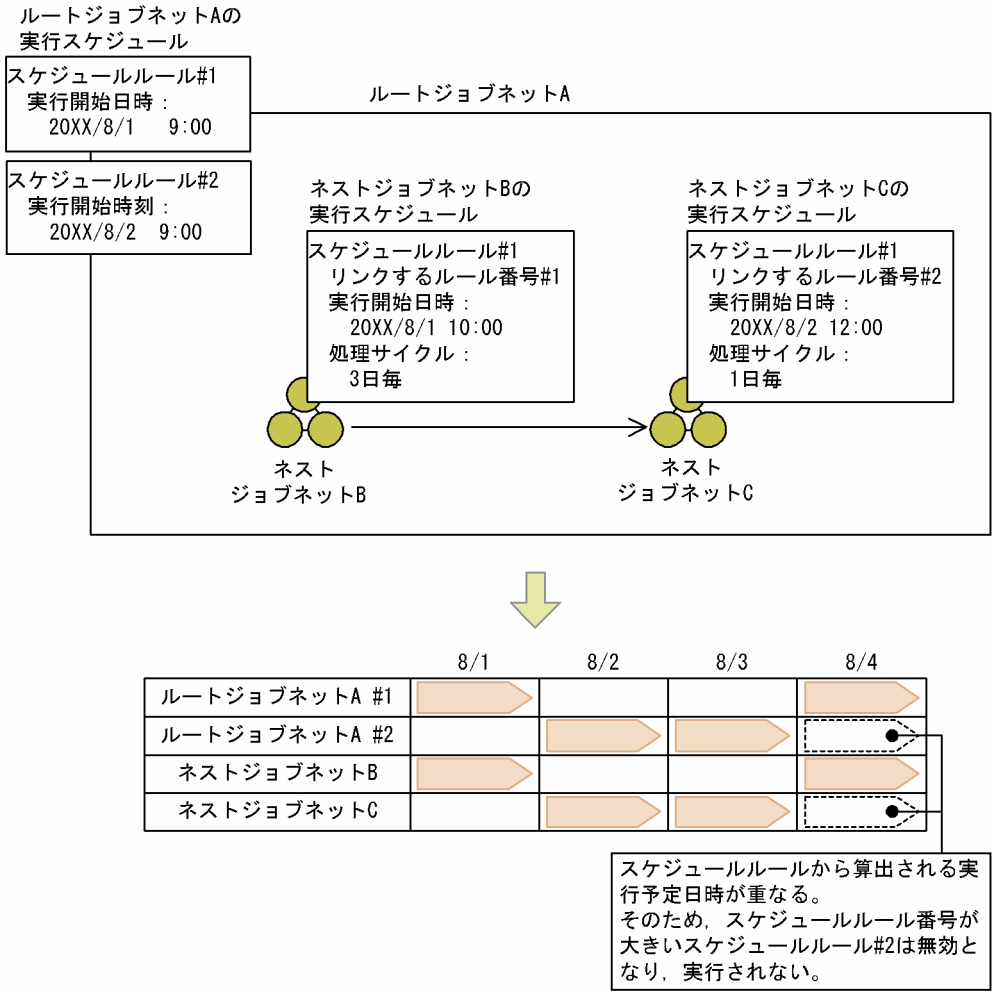


この場合，ネストジョブネットの開始日時（5:00）が基準時刻（6:00）より前になっていて，ルートジョブネットとネストジョブネットの実行開始日が異なるため，ネストジョブネットは実行されません。

- ルートジョブネットに複数のスケジュールルールを定義し，それぞれのスケジュールルールから算出される実行予定日時が重なった場合，重なったスケジュールルールの中で最も小さいルール番号のスケジュールが有効になり，それ以外のスケジュールは無効になります。なお，無効になったスケジュールルールをリンクしているネストジョブネットのスケジュールも同様に無効になります。

  スケジュールルールから算出される実行予定日時が重なった場合の例を次の図に示します。

図 3-9　スケジュールルールから算出される実行予定日時が重なった場合の例



この例では，スケジュールルール#1 とスケジュールルール#2 から算出されるルートジョブネット A の 8/4 の実行予定日時が重なります。この場合，スケジュールルール番号が小さいスケジュールルール#1 が有効となり，スケジュールルール#2 は無効となります。また，スケジュールルール#1 とリンクしているネストジョブネット B は実行され，無効になったスケジュールルール#2 とリンクしているネストジョブネット C は実行されません。

(3)　ネストジョブネットにスケジュールオプションを設定する

ルートジョブネットと同じように，ネストジョブネットにスケジュールオプションを設定できます。スケジュールオプションについては，マニュアル「JP1/Automatic Job Management System 3 導入ガイド 3.3.2 スケジュール情報の定義」を参照してください。

(a)　他のジョブグループのカレンダーを参照

ネストジョブネットに，上位ジョブネットと異なるカレンダー定義を参照させることができます。

上位ジョブネットと異なるカレンダー定義を参照させた場合，ネストジョブネットは，上位ジョブネットと自ジョブネットの実行日が重なるときに実行されます。

(b) 排他スケジュール

自ネストジョブネットの実行日が，同一階層に存在するネストジョブネットの実行日と重なる場合，自ネストジョブネットを実行させないようにできます。

排他スケジュールを設定したネストジョブネットの実行予定は，次の条件が重なる場合に取り消されます。

- 排他スケジュールに指定したネストジョブネットに，リンクするルール番号が同じスケジュールルールが存在する。
- リンクするルール番号が同じスケジュールルールから算出される実行日が重なる。

リンクするルール番号が異なるスケジュールルールから算出される実行日が重なるときは，排他スケジュールを設定したネストジョブネットの実行予定は取り消されません。

3.5.4 1 日のうちに同じジョブネットを何回か定期的に実行する（サイクルジョブ定義）

1 日の中で同じジョブネットを定期的に複数回実行する場合，次の三つの方法のどれかで定義できます。

- 方法 1：ジョブネットを一つ作成し，スケジュールルールを複数設定する。
- 方法 2：開始時刻ごとにジョブネットを作成する。
- 方法 3：実行間隔制御を起動条件として設定する。

(1) ジョブネットの定義方法

それぞれの方法について，朝 7 時から翌朝 6 時まで 1 時間おきに 24 回実行するジョブネットを例にして説明します。

方法 1：スケジュールルールを複数作成する

1. ジョブネットを一つ作成し，実行するジョブを定義する。
2. スケジュールルールを 24 個作成し，それぞれの開始時刻を 07:00，08:00，09:00・・・と定義する。

ジョブネットを実行登録すると，スケジュールルールで設定した開始時刻に従って，1 時間おきにジョブネットが実行されます。

なお，JP1/AJS3 - View を使ってスケジュールルールを定義する場合，一定間隔のスケジュールルールを，任意の時刻までまとめて定義することもできます。

方法 2：開始時刻ごとにジョブネットを作成する

1. ジョブネットを一つ作成する。
2. ネストジョブネットを作成し，実行するジョブを定義する。
3. 2.で作成したネストジョブネットをコピーし，同じ処理内容のジョブネットを全部で 24 個作成する。

4. 各ジョブネットの開始時刻に，07:00 から 06:00 までの異なる時刻を設定する。

ジョブネットを実行登録すると，開始時刻に従って各ジョブネットが実行されます。

方法 3：実行間隔制御を起動条件として設定する

1. ジョブネットを一つ作成し，実行するジョブを定義する。
2. 起動条件を作成し，実行間隔制御ジョブを貼り付ける。
3. 実行間隔制御ジョブの［待ち時間］に 60 分，［開始直後に満了］に［する］を設定する。
4. スケジュールルールを作成し，実行開始時刻を 7:00，起動条件の有効範囲を 24 回と設定する。

このジョブネットを実行登録すると，実行開始時刻の 7:00 に起動条件の監視が始まり，すぐに起動条件が成立してジョブネットの最初の実行が開始します。その後，起動条件に従って 60 分ごとにジョブネットが実行されます。

(2) 三つの方法の長所と短所

それぞれの方法の長所と短所について，次に説明します。

表 3-4 同じジョブネットを定期的に実行する方法の長所と短所

定義方法	長所	短所
方法 1	• ジョブネットの定義が簡単。 • ジョブネットの構成が単純。 • ルートジョブネットとして定義した場合は，多重起動やスケジュールスキップが設定できる。	• スケジュールルールを変更する場合，スケジュールルールすべてを変更しなければならない。
方法 2	• どのようなジョブネットにでも適用できる。 • いま実行しているのが何回目のジョブネットなのかが［ジョブネットモニタ］ウィンドウで確認できる。 • ジョブの再実行，実行中止などの操作が，［ジョブネットモニタ］ウィンドウでできる。	• 実行回数分のネストジョブネットを作成しなくてはならない。 • ネストジョブネットの数だけ，スケジュールを作らなくてはならない。 • スケジュールルールを変更する場合，すべてのネストジョブネットのスケジュールルールを変更しなければならない。 • 実行する処理を変更する場合に時間が掛かる。
方法 3	• ジョブネットの構成が単純。	• ルートジョブネットでしか定義できない。

3.5.5 算出したスケジュールを基に，実行予定日を前後にずらす（起算スケジュール）

カレンダー定義やスケジュールルールから算出された業務の実行予定日を起点として，そこからさらに「何日前」または「何日後」という形で実行予定日をずらす設定について説明します。

このような設定は，起算スケジュールを使って定義します。起算スケジュールについては，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.3.2 スケジュール情報の定義」を参照してください。

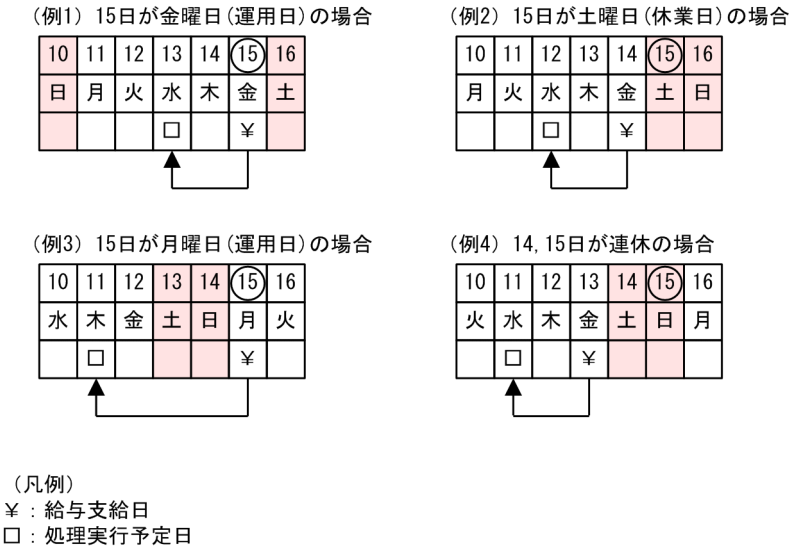
起算スケジュールについて，給与支給日の前に給与計算処理をするジョブネットを例に説明します。

(1)　給与支給日の 2 運用日前に給与計算処理をする場合

給与支給日が毎月 15 日で，その 2 運用日前に給与計算処理をするジョブネットを例に説明します。15 日が休業日の場合，給与支給日は，15 日よりも前で最も近い運用日に振り替えるものとします。

給与支給日の 2 運用日前に給与計算処理をする例を，次の図に示します。

図 3-10　給与支給日の 2 運用日前に給与計算処理をする例



（例 1）

15 日が金曜日（運用日）の場合，給与計算処理のジョブネットは 2 運用日前の 13 日に実行します。

（例 2）

15 日が土曜日（休業日）の場合，給与支給日は 15 日より前で最も近い運用日の 14 日になります。そのため，給与計算処理のジョブネットは 14 日の 2 運用日前の 12 日に実行します。

（例 3）

15 日が月曜日（運用日）で，13 日，14 日が休業日の場合，給与計算処理のジョブネットは 2 運用日前の 11 日に実行します。

（例 4）

15 日が日曜日（休業日）で，14 日も休業日の場合，給与支給日は 15 日より前で最も近い運用日の 13 日になります。そのため，給与計算処理のジョブネットは 13 日の 2 運用日前の 11 日に実行します。

このような運用をする場合，運用日による起算スケジュールを使用します。

運用日による起算スケジュールを使用すると，カレンダー定義や休業日の振り替えなどから算出された日を起点として，そこから n 運用日前に実行予定日をずらします。そのため，給与計算処理のジョブネットを，給与支給日の 2 運用日前に実行するように設定できます。

図に示したスケジュールを，起算スケジュールを使って設定するには，スケジュールルールを次のように定義します。

定義内容

実行開始日：絶対日　15 日

処理サイクル：1 月

休業日の振り替え：前の運用日に振り替え

起算スケジュール：開始日時から 2 運用日前に実行する

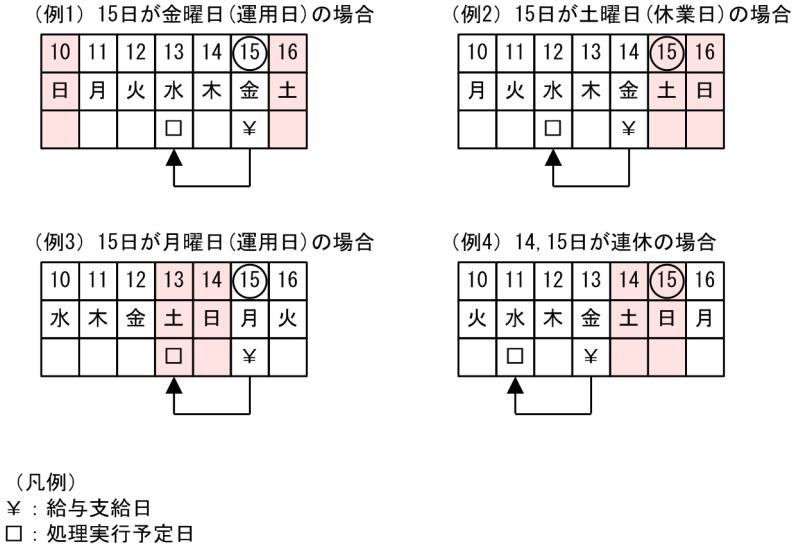
この定義を実行登録すると，まず，実行開始日，処理サイクル，および休業日の振り替えの設定に基づいてスケジュールの起点日（給与支給日）を算出します。そして，実際の実行予定日を起点日から 2 運用日前に設定します。

(2)　給与支給日の 2 日前に給与計算処理をする場合

給与支給日が毎月 15 日で，その 2 日前に給与計算処理をするジョブネットを例に説明します。15 日が休業日の場合，給与支給日は，15 日よりも前で最も近い運用日に振り替えるものとします。また，給与計算処理は，その日が休業日であっても運用日であっても実行します。

給与支給日の 2 日前に給与計算処理をする例を，次の図に示します。

図 3–11　給与支給日の 2 日前に給与計算処理をする例



（例 1）

15 日が金曜日（運用日）の場合，給与計算処理のジョブネットは 2 日前の 13 日に実行します。

（例 2）

15 日が土曜日（休業日）の場合，給与支給日は 15 日より前で最も近い運用日の 14 日になります。そのため，給与計算処理のジョブネットは 14 日の 2 日前の 12 日に実行します。

（例 3）

15 日が月曜日（運用日）で，13 日，14 日が休業日の場合，給与計算処理のジョブネットは 2 日前の 13 日に実行します。

（例 4）

15 日が日曜日（休業日）で，14 日も休業日の場合，給与支給日は 15 日より前で最も近い運用日の 13 日になります。そのため，給与計算処理のジョブネットは 13 日の 2 日前の 11 日に実行します。

このような運用をする場合，日数による起算スケジュールを使用します。

日数による起算スケジュールを使用すると，カレンダー定義や休業日の振り替えなどから算出された日を起点として，そこから n 日前に実行予定日をずらします。そのため，給与計算処理のジョブネットを，給与支給日の 2 日前に実行するように設定できます。

図に示したスケジュールを，起算スケジュールを使って設定するには，スケジュールルールを次のように定義します。

定義内容

実行開始日：絶対日　15 日

処理サイクル：1 月

休業日の振り替え：前の運用日に振り替え

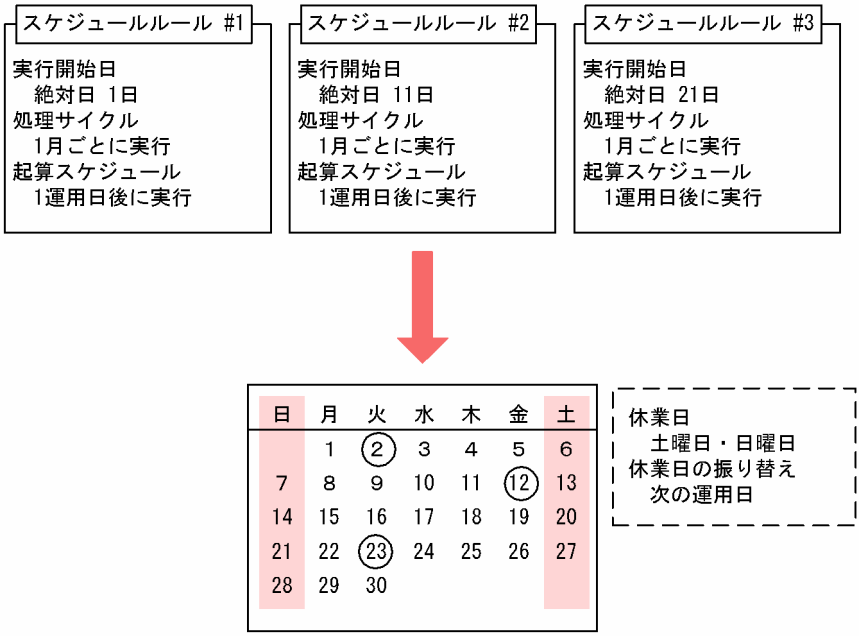
起算スケジュール：開始日時から 2 日前に実行する

この定義を実行登録すると，まず，実行開始日，処理サイクル，および休業日の振り替えの設定に基づいてスケジュールの起点日（給与支給日）を算出します。そして，実際の実行予定日を起点日から 2 日前に設定します。

3.5.6　月の上旬，中旬，下旬を意識したスケジュールを設定する

月の上旬，中旬，下旬にそれぞれジョブネットを実行する場合のスケジュール定義方法について説明します。このような定義をする場合，上旬・中旬・下旬のそれぞれの実行開始日を定義したスケジュールルールを作成します。例えば，上旬・中旬・下旬のそれぞれ最終の営業日に処理を実行する場合は，実行開始日として，10 日，20 日，月末を指定します。また，これらの日付からさらに実行日をずらす場合は起算スケジュールを使います。起算スケジュールを使って旬単位の処理を実行する例を次の図に示します。

図 3-12　旬単位の処理の実行例



この例では，上旬，中旬，下旬のそれぞれ最初の営業日を起点として，その次の営業日に処理を実行するようにスケジュールを組んでいます。下旬の最初の日は 21 日ですが，21 日が日曜日で休業日のため，次の営業日である 22 日にずらし，さらに起算スケジュールで 1 日後ろにずらして，23 日を実行予定日として算出しています。

3.5.7　業務ごとにカレンダーを定義する

業務の種類や担当部署によって，業務の運用日が異なる場合があります。このような場合に，業務ごとに異なるカレンダーを定義できます。

例として，東京本社，大阪支社，および名古屋支社で実行する業務があり，それぞれ運用日が異なる場合を考えてみます。それぞれの業務カレンダーは次のようになっています。

- 東京本社：毎日実行
- 大阪支社：月～土曜日に実行
- 名古屋支社：月～金曜日に実行

業務の構築の方法は 2 とおり考えられます。

- 同じ業務カレンダーのジョブネットをまとめてジョブグループを作る。
- 業務を一つのジョブグループに作り，別のジョブグループに設定したカレンダーを参照させる。

次にそれぞれの方法について説明します。

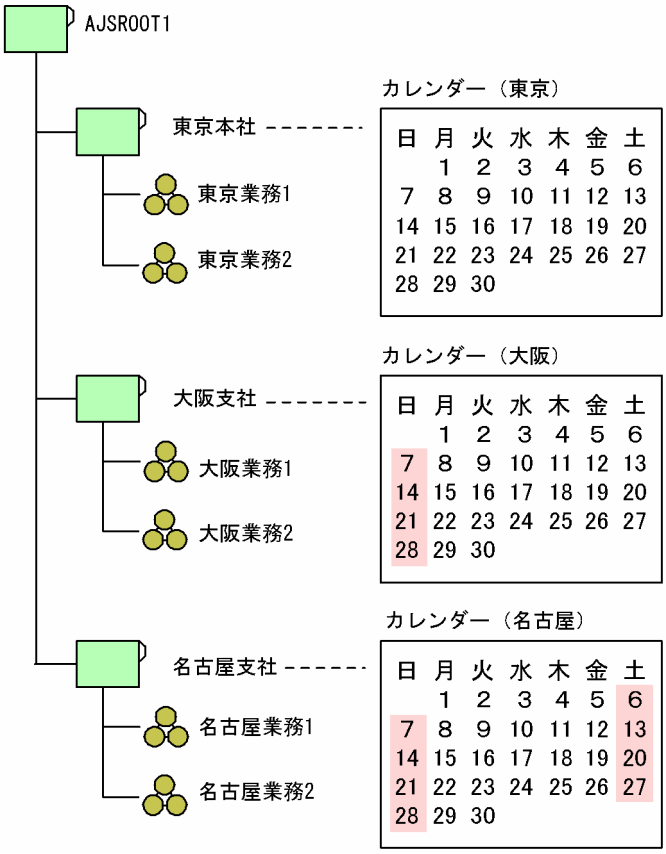
(1)　同じ業務カレンダーのジョブネットをまとめてジョブグループを作る

次の条件に当てはまるような業務には，この運用方法が適しています。

- 複数の業務グループがあるが，業務グループ間の依存関係はない。
- 業務グループごとにシステム管理者がいる。
- 複数の業務グループを 1 台のマネージャーで管理する。

業務構成例を次の図に示します。

図 3-13　同じ業務カレンダーのジョブネットをまとめてジョブグループを作る場合



スケジュールの設定方法の概要を次に示します。

1. スケジューラーサービスの下に，東京・大阪・名古屋のジョブグループを作成する。
2. 各ジョブグループの下にジョブネットを定義する。
3. 各ジョブグループにカレンダーを定義する。
4. ジョブネットを実行登録する。

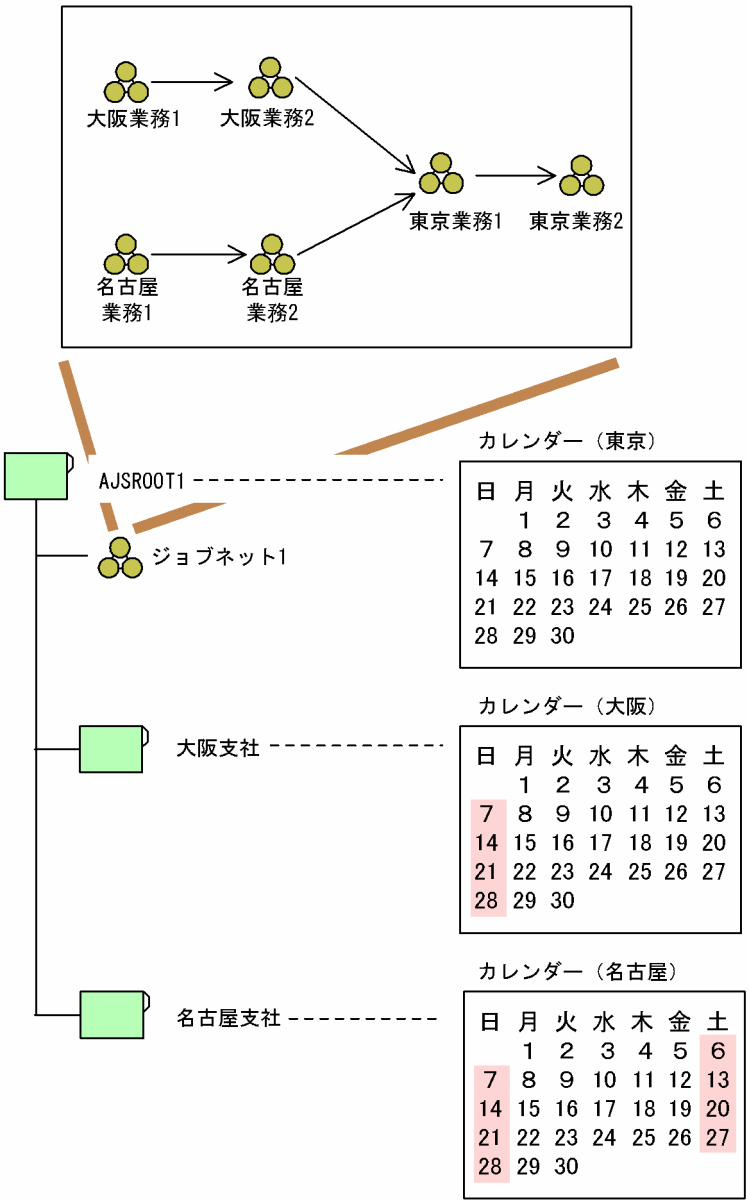
(2) 業務を一つのジョブグループに作り，別のジョブグループに設定したカレンダーを参照させる

次の条件に当てはまるような業務には，この運用方法が適しています。

- 複数の業務グループがあり，業務グループ間の依存関係がある。
- すべての業務を一人のシステム管理者が管理する。

業務構成例を次の図に示します。

図 3-14　別のジョブグループに設定したカレンダーを参照させる場合



スケジュールの設定方法の概要を次に示します。

1. スケジューラーサービスの下にジョブネット（ジョブネット 1）を一つ作成する。

2. ルートジョブネットの中に東京・大阪・名古屋で実行する業務をネストジョブネットとして定義する。
3. スケジューラーサービスに東京本社のカレンダーを定義する。
4. 大阪，名古屋のジョブグループを作る。
5. 大阪，名古屋のジョブグループにカレンダーを定義する。
   このジョブグループにはカレンダーだけを定義し，ジョブネットは作りません。
6. 大阪，名古屋で実行するネストジョブネットに，それぞれのジョブグループに定義したカレンダーを参照させる。
   ［スケジュールの設定］ダイアログボックスで，［他のジョブグループのカレンダーを参照］をチェックします。大阪業務 1，大阪業務 2 は，大阪支社ジョブグループのカレンダーを参照させます。名古屋業務 1，名古屋業務 2 は，名古屋支社ジョブグループのカレンダーを参照させます。
7. ルートジョブネットを実行登録する。

この方法の場合は，監視画面ですべてのジョブネットを一度に監視できます。

(3) カレンダーの適用例

カレンダーの適用例を紹介します。

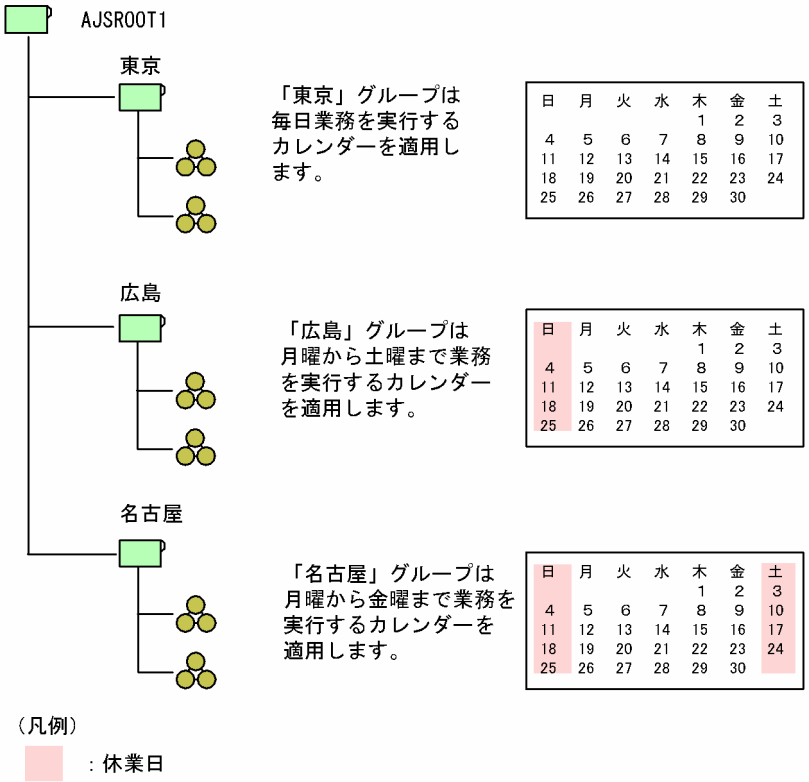
(例 1)

次に示す条件を前提とした場合のカレンダーの設定例を説明します。

- 複数の業務グループがあるが，お互いにカレンダー情報の依存関係を持たない。
- 業務グループごとにそれぞれのシステム管理者で管理する。
- 複数の業務グループを 1 台の業務管理ホストで管理する。

このような前提条件の場合，次の図のような業務グループの階層を作成し，カレンダーを定義します。

図 3-15　カレンダーの適用例 1



このような例で監視画面を使用すると，業務グループの下にある業務の実行予定や結果しか確認できません。例えば，一度に実行予定や結果を確認できるのは，「東京」グループの下にある業務だけです。そのため，複数のシステム管理者が，それぞれの業務グループを管理する場合，担当のシステム管理者に関係する業務の予定や結果を管理しやすくなります。

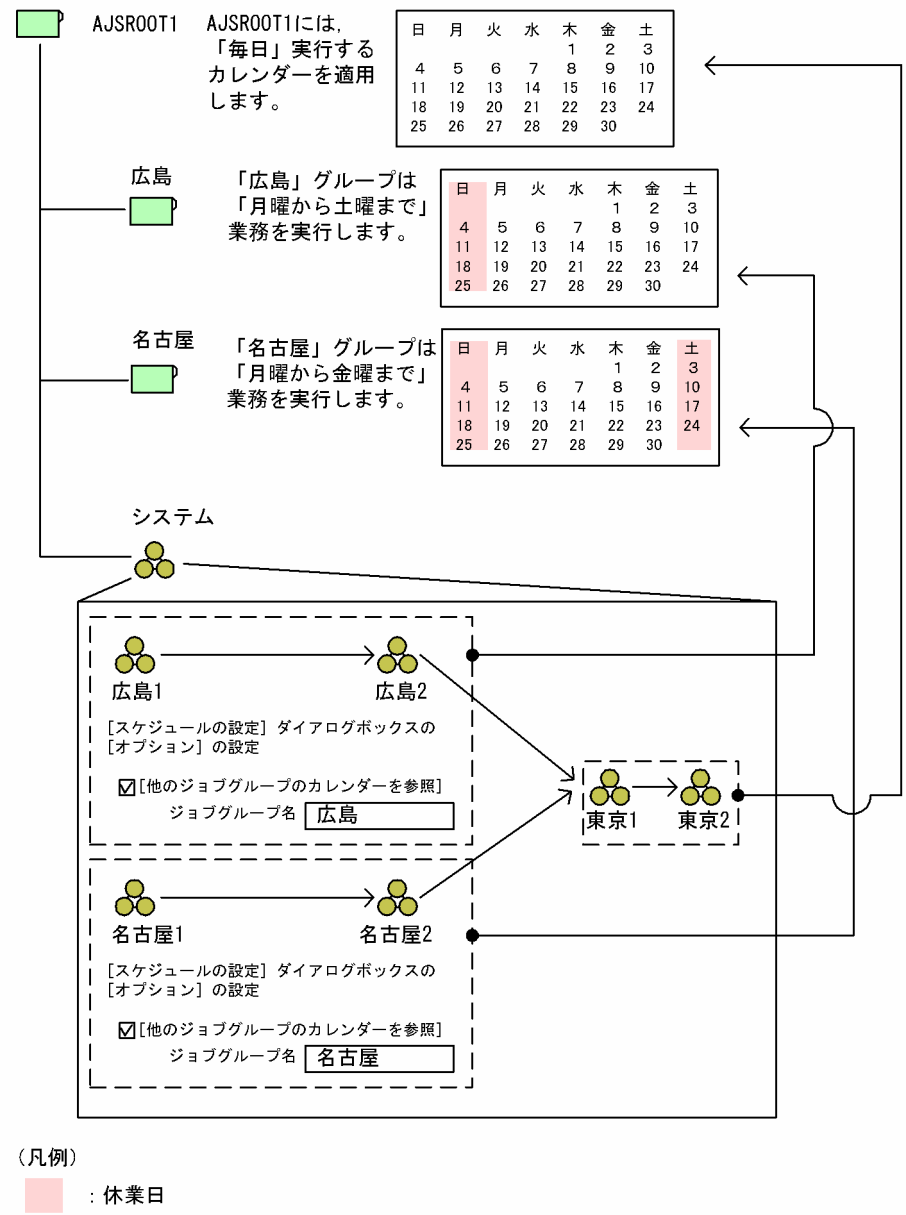
(例 2)

次に示す条件を前提とした場合のカレンダーの設定例を説明します。

- 複数の業務グループがあるが，お互いにカレンダー情報の依存関係を持つ。
- すべての業務を一人のシステム管理者で管理する。

このような前提条件の場合，次の図のような業務グループの階層を作成し，カレンダーを定義します。

図 3-16　カレンダーの適用例 2



この例のように設定すると，各業務は次のように実行されます。

東京の業務

毎日業務を実行するカレンダーを参照しているため，業務が毎日実行されます。

広島の業務

日曜日だけが休業日のカレンダーを参照しているため，業務が月曜日から土曜日まで実行されます。

名古屋の業務

土曜日と日曜日が休業日のカレンダーを参照しているため，業務が月曜日から金曜日まで実行されます。

このような例で，「AJSROOT1」というグループを監視画面で確認すると，業務グループの下にある，すべての業務の予定や結果を一度に確認できます。そのため，一人のシステム管理者がすべての業務を管理しやすくなります。

4 実行登録方法の検討

スケジュールルールを定義したジョブネットは，実行登録することで JP1/AJS3 にスケジュールされます。

この章では，JP1/AJS3 にどのような方法でジョブネットを実行登録するかの検討について説明します。

4.1　ジョブネットの実行登録方法

ジョブネットの実行登録方法には，次の三つがあります。

- 計画実行登録
- 確定実行登録
- 即時実行登録

それぞれの登録方法の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.1.1 実行登録の方法」を参照してください。

それぞれの実行登録方法の特徴を踏まえて，運用に適した方法を検討してください。

4.1.1　計画実行登録

カレンダー情報や，ジョブネットに設定されたスケジュールルールに従ってスケジュールを算出し，ジョブネットを実行します。ただし，算出するスケジュールは擬似予定（シミュレーションによる実行予定）として扱われます。

(1)　計画実行登録の特徴

計画実行登録の特徴は次のとおりです。

(a)　実行契機

カレンダー情報や，ジョブネットに設定したスケジュールルールに従ってスケジュールを算出し，ジョブネットを実行します。ジョブネットの実行開始時に，次の実行予定をスケジュールとして確定します。

(b)　スケジュールの一時変更

計画実行登録によって算出されるスケジュールの場合，次回実行予定に限って一時変更できます。ただし，それ以降のスケジュールは擬似予定であるため，変更できません。

(c)　カレンダー情報やスケジュールルールの変更

計画実行登録によって算出されるスケジュールの場合，カレンダー情報やジョブネットのスケジュールルールを変更すると，その内容に基づいてスケジュールが再計算されます。したがって，例えば年度が変わって運用日・休業日を再設定したり，スケジュールルールが変更になったりした場合でも，実行登録を解除しないでカレンダー情報やスケジュールルールを変更できます。

(2)　計画実行登録のベターユース

次のような場合に使用することを推奨します。

- カレンダー情報やスケジュールルールを変更することがある場合
- ファイルの更新などユーザーの行動を契機として（起動条件の設定が前提），1 日に決まった回数だけジョブネットを実行する場合
- 一時的なスケジュール変更が生じない場合

4.1.2　確定実行登録

カレンダー情報や，ジョブネットに設定されたスケジュールルールに従ってスケジュールを算出し，ジョブネットを実行します。実行登録と同時に，登録の際に指定された回数または期間のスケジュールを算出して確定します。

(1)　確定実行登録の特徴

確定実行登録の特徴は次のとおりです。

(a)　実行契機

カレンダー情報や，ジョブネットに設定したスケジュールルールに従ってスケジュールを算出し，ジョブネットを実行します。

(b)　スケジュールの一時変更

確定実行登録によって算出されるスケジュールの場合，実行登録時に指定した回数または期間分の確定スケジュールについては一時変更できます。また，実行予定日の追加もできます。

(c)　カレンダー情報やスケジュールルールの変更

確定実行登録によって算出されるスケジュールの場合，カレンダー情報やジョブネットのスケジュールルールを変更しても，登録時に指定した回数または期間分の確定スケジュールには反映されません。カレンダー情報やジョブネットのスケジュールルールの変更内容を反映させたい場合は，一度実行登録を解除し，登録し直す必要があります。

(2)　確定実行登録のベターユース

次のような場合に使用することを推奨します。

- 特定の期間や回数だけジョブネットを実行する場合
- 確定したスケジュールを一時的に変更したり，スケジュールを追加したり，実行を中止したりするおそれがある場合

4.1.3　即時実行登録

ジョブネットを実行登録すると，カレンダー情報やスケジュールルールに関係なく，登録と同時に 1 回だけジョブネットを実行します。

(1)　即時実行登録の特徴

即時実行登録の特徴は次のとおりです。

(a)　実行契機

実行登録と同時にジョブネットの実行を開始します。ジョブネットにスケジュールルールが設定されている場合でも，スケジュールを無視して即時に実行します。

(b)　スケジュールの一時変更

即時実行登録の場合，ジョブネットの次回実行予定がないため無効です。

(c)　カレンダー情報やスケジュールルールの変更

即時実行登録の場合，カレンダー情報やジョブネットのスケジュールルールは無視されるため無効です。

(2)　即時実行登録のベターユース

次のような場合に使用することを推奨します。

- コマンドや人手でジョブネットの実行を開始する場合
- アプリケーション中からジョブネットを実行する場合
- ファイル転送アプリケーション（JP1/FTP など）が正常終了したときに起動されるコマンドの実行を契機として，ジョブネットを実行する場合
- JP1/IM の自動アクションを契機として，ジョブネットを実行する場合
- スケジュールを定義しているジョブネットに対して，処理フローの整合性テストを実施する場合

5 監視方法の検討

この章では，JP1/AJS - View，Web GUI，またはユーザーアプリケーションを使って業務を監視するための検討項目について説明します。

5.1　JP1/AJS3 - View での監視方法の検討

JP1/AJS3 - View でジョブネットを監視するに当たって，検討しておく項目を説明します。

なお，JP1/AJS3 - View の環境設定の検討については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 4.5 JP1/AJS3 - View の環境設定について検討する」を参照してください。

5.1.1　サマリー監視画面での監視対象を検討する

[JP1/AJS3 - View］ウィンドウ（サマリー監視画面）では，任意のジョブネットを監視対象として設定しておき，進捗度や配下のジョブの状態分布など，ジョブネットの実行状況を監視できます。ジョブネットの終了予定日時を過去の実行所要時間からシミュレーションして表示することもでき，開始や終了の遅延が業務に大きな影響を与えるようなジョブネットを監視するのに適しています。

異なる階層のジョブネットや，異なるスケジューラーサービス配下のジョブネットでも，一括で監視できます。

5.1.2　ジョブネットの遅延監視について検討する

JP1/AJS3 では，ジョブネットの遅延を次の方法で監視できます。

(1)　日時指定による遅延監視

日時を指定し，ジョブネットが予定日時に開始または終了するかどうかを監視できます。日時指定による遅延監視には，開始遅延監視と終了遅延監視があります。

開始または終了の遅延が，そのあとのジョブネットの実行スケジュールに大きな影響を与えるようなジョブネットに設定します。

開始遅延の監視

開始が遅れている状態であるかどうかを判断する日時を設定します。ジョブネットの開始予定時刻から，ある時間以内に開始されれば正常と見なし，それ以上時間が過ぎても開始されない場合は，何らかの異常が発生していると見なします。

終了遅延の管理

終了が遅れている状態であるかどうかを判断する日時を設定します。ジョブネットの開始予定時刻から，ある時間以内に終了されれば正常と見なし，それ以上時間が過ぎても終了しない場合は，何らかの異常が発生していると見なします。

開始遅延と終了遅延の両方を指定することもできます。また，どちらか一方だけを指定することもできます。

(2) 所要時間指定による遅延監視

ジョブネットを開始してから終了するまでの所要時間を指定して遅延監視ができます。

指定した所要時間を超えたら，業務のアイコンの表示色が変化したり，イベントが発行されたりして遅延したことを通知します。

5.1.3 ジョブの実行所要時間による終了遅延監視

ジョブの実行所要時間による終了遅延監視では，マネージャーホスト上でジョブが実行中になっている時間を監視するため，エージェントホストでのジョブ実行所要時間と異なることがあります。このことから，終了遅延が検知されたり，されなかったりする場合があります。

一方，ジョブの打ち切り時間はエージェントホストで実行中になっている時間を監視します。このため，打ち切り時間と終了遅延監視では，検出するタイミングに差異があります。

例えば，次のような場合には，マネージャーホストとエージェントホストで，ジョブの実行所要時間が異なります。

- **スケジューラーサービス停止中にエージェントホストでジョブが開始した場合**

  スケジューラーサービスの停止中にエージェントホストでジョブの実行が開始された場合（マネージャーがエージェントからのジョブの開始通知を受け取る前にスケジューラーサービスが停止した場合），ジョブはスケジューラーサービスが起動されてから実行中の状態となります。したがって，実際にエージェントホストでジョブが実行されていた時間は，マネージャー上の実行時間より長くなります。この場合，例えばジョブ実行所要時間を 10 分としていて，エージェントホストで 20 分間実行中だったとしても，スケジューラーサービスの起動後 5 分で終了した場合，終了遅延は検知されません。

- **スケジューラーサービス停止中にエージェントホストでジョブが終了した場合**

  スケジューラーサービスの停止中に，エージェントホストですでに実行中だったジョブが終了した場合（マネージャーがエージェントからのジョブの終了通知を受け取る前にスケジューラーサービスが停止した場合），ジョブはスケジューラーサービスが起動されてから終了状態となります。したがって，実際にエージェントホストでジョブが実行されていた時間は，マネージャー上の実行時間が短くなります。この場合，例えばジョブ実行所要時間を 10 分としていて，エージェントホストで 5 分で終了したとしても，マネージャー上はスケジューラーサービス停止中の間も実行中のままのため，停止時間が 10 分を超える場合は終了遅延となります。

- **スケジューラーサービス停止中にエージェントホストでジョブが開始・終了した場合**

  スケジューラーサービスの停止中に，エージェントホストでジョブが開始し，終了した場合（マネージャーがエージェントからのジョブの開始通知および終了通知を受け取る前にスケジューラーサービスが停止した場合），ジョブはスケジューラーサービスが起動されてから実行中となり，すぐに終了状態となります。したがって，マネージャー上のジョブ実行時間は 0 分となります。この場合，例えばジョブ実行所要時間を 10 分としていて，エージェントホストで 20 分間実行中だったとしても，マネージャー上のジョブ実行時間は 0 分であるため，終了遅延は検知されません。

補足事項

- 終了遅延を検知したジョブについて，エージェントホストでのジョブ実行時間を確認したい場合は，ajsshow コマンドを，フォーマット指示子「%V」，「%Q」を指定して実行し，ジョブの実行開始および終了日時を取得することで確認できます。ajsshow コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajsshow」を参照してください。
- スケジューラーサービスの環境設定パラメーターJOBDELAYWARNMSG に「yes」を設定しておくと，終了遅延監視対象のジョブがスケジューラーサービス停止中に実行開始した場合，統合トレースログにメッセージ「KAVS0249-W ジョブ（**ジョブ名：実行 ID**）が実行開始する前にスケジューラーサービスが停止しました」が出力されます。このメッセージによって該当するジョブ名および実行 ID を特定できます。

  エージェントホスト上でのジョブ実行所要時間によって終了遅延を確認したい場合は，メッセージ KAVS0249-W で特定したジョブに対し，ajsshow コマンドでジョブの実行開始および終了日時を取得して確認してください。

  環境設定パラメーターJOBDELAYWARNMSG については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4 スケジューラーサービス環境設定」を参照してください。

5.2 Web GUI での監視方法の検討

Web GUI でジョブネットを監視するに当たって，検討しておく項目を説明します。

なお，Web GUI の環境設定の検討については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 4.6 JP1/AJS3 - Web Console の環境設定について検討する」を参照してください。

5.2.1 ［ダッシュボード］画面での監視対象を検討する

［ダッシュボード］画面では，業務の実行進捗度，配下のジョブの状態別の件数，遅延の有無などを監視できます。前日の業務の実行結果を確認したり，当日のジョブネットの進捗を監視したりするのに適しています。

［ダッシュボード］画面で監視するジョブネットは，事前に Web GUI で登録しておく必要があります。Web GUI で監視したいジョブネットをあらかじめ検討しておいてください。

5.3　API を利用したユーザーアプリケーションでの監視方法の検討

ユーザーアプリケーションでジョブネットを監視する場合，監視対象のユニットは，ユーザーアプリケーションを使用するユーザーの業務に合わせて任意に選択できます。しかし，大量のユニットを監視対象として表示できるようにしたり，複数のユニットを一度に操作できるようにしたりすると，ユーザーアプリケーションの処理速度が遅くなったり，JP1/AJS3 - Manager への負荷が増えたりするおそれがあります。

ユーザーアプリケーションを開発する際は，次のような項目について，事前に検討しておいてください。

- 監視対象とするユニットの数や階層
- ユーザーアプリケーションを使用するユーザーの業務内容

API を利用したユーザーアプリケーションの実装に関する検討事項や注意事項の詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　6. API の概要」を参照してください。

6 アクセス権の検討

この章では，ジョブネットへのアクセス権の検討や，JP1 ユーザーと OS ユーザーの対応づけの検討について説明します。

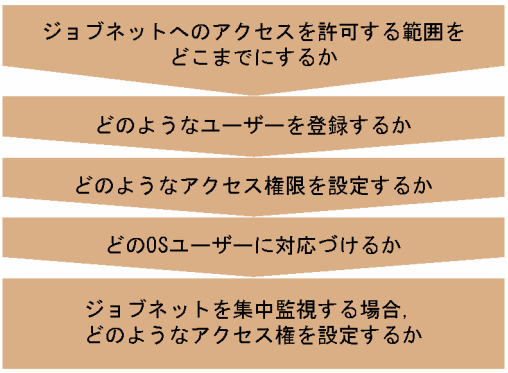
6.1　ジョブネットへのアクセス権を検討する流れ

ユーザーに設定する，ジョブネットへのアクセス権の検討について説明します。JP1/AJS3 を使用するユーザーに，ジョブの実行だけを許可する権限や，処理結果の参照だけを許可する権限を与えることで，誤った操作が行われることを防ぐことができます。

ジョブネットへのアクセス権を検討する流れを次の図に示します。

なお，ジョブネットへのアクセス権の設定・管理には，JP1/Base のユーザー管理機能（ユーザー認証，ユーザーマッピング）を利用します。ユーザー管理機能の詳細については，マニュアル「JP1/Base 運用ガイド」を参照してください。

図 6–1　ジョブネットへのアクセス権を検討する流れ



6.2　アクセス権を設定する範囲についての検討

JP1/AJS3 では，JP1 ユーザーという独自のユーザー名を使用して，ジョブの実行を許可する権限や，処理内容の変更を許可する権限などのアクセス権限を管理しています。

その JP1 ユーザーの登録と管理は，JP1/Base で提供されているユーザー認証機能を使用しています。そのユーザー認証機能で，JP1 ユーザーのアクセス権限を管理している JP1/Base のホストを**認証サーバ**と呼びます。ネットワーク上にある各 JP1/Base のホストで，どのホストにインストールされた JP1/Base を認証サーバにするか指定すると，その認証サーバがアクセス権を管理するホストの範囲（ユーザー認証圏）を決めることができます。そのため，JP1/AJS3 を導入する場合，認証サーバがアクセス権を管理するホストの範囲（ユーザー認証圏）について検討しておく必要があります。

次の表に示す内容を考慮に入れて，ユーザー認証圏をどのようにするか検討してください。

表 6–1　認証圏の数と，長所・短所

認証圏の数	長所・短所
ユーザー認証圏をシステム内に一つだけ構築した場合	システム管理者は，JP1 ユーザーを統一管理でき，登録や変更作業にあまり手間が掛かりません。
ユーザー認証圏をシステム内に複数構築した場合	システム管理者は，構築したユーザー認証圏の数だけ JP1 ユーザーを管理する必要が生じます。JP1 ユーザーの登録や変更などの管理の手間が発生してしまいます。また，異なる認証サーバごとにログイン認証が発生します。ただし，個々の認証サーバが独立しているため，システムとしての耐性は強くなります。

なお，一つのユーザー認証圏内には，認証サーバを 2 台設定できます。通常使用する認証サーバを**プライマリー認証サーバ**，予備として稼働し，トラブル発生時に使用する認証サーバを**セカンダリー認証サーバ**と呼びます。一つのユーザー認証圏内に，認証サーバを 1 台だけ設定している場合は，認証サーバが起動していなかったり，通信障害が発生したりするなど，何らかの理由で認証サーバに接続できなかった場合に，ジョブやリモートコマンドなどが実行できなくなり，業務が停止するおそれがあります。

2 台の認証サーバを設定しておくと，通常使用する認証サーバ（プライマリー認証サーバ）にトラブルが発生しても，予備の認証サーバ（セカンダリー認証サーバ）によってジョブやリモートコマンドを実行できます。ユーザー認証圏がシステム内に一つの場合でも，2 台の認証サーバを設定することで，システムとしての耐性は強くなります。必要に応じて，2 台の認証サーバを利用することを検討してください。

なお，プライマリー認証サーバとセカンダリー認証サーバを設定する場合は，プライマリー認証サーバに設定されている JP1 ユーザーや JP1 資源グループなどの情報をセカンダリー認証サーバにコピーして，設定内容を同じにしておいてください。設定内容が同じでない場合，切り替えが発生したときに認証エラーが発生します。

プライマリー認証サーバとセカンダリー認証サーバの設定の詳細については，マニュアル「JP1/Base 運用ガイド」を参照してください。

プライマリー認証サーバとセカンダリー認証サーバを設定した場合の，認証サーバの指定方法については，次に示す個所を参照してください。

- JP1/AJS3 - Manager がインストールされている Windows ホストの場合

  マニュアル「JP1/Automatic Job Management System 3 構築ガイド　3.1.1 JP1/Base のセットアップ」

- JP1/AJS3 - Agent がインストールされている Windows ホストの場合

  マニュアル「JP1/Automatic Job Management System 3 構築ガイド　3.2.1 JP1/Base のセットアップ」

- JP1/AJS3 - Manager がインストールされている UNIX ホストの場合

  マニュアル「JP1/Automatic Job Management System 3 構築ガイド　13.1.1 JP1/Base のセットアップ」

- JP1/AJS3 - Agent がインストールされている UNIX ホストの場合

  マニュアル「JP1/Automatic Job Management System 3 構築ガイド　13.2.1 JP1/Base のセットアップ」

6.3　登録するユーザーの検討

ユーザーのアクセス権を管理する範囲や，アクセス権を管理するホストの検討が終了したら，JP1/AJS3 で使用する JP1 ユーザーの登録について検討します。運用に応じて JP1 ユーザーを登録し，JP1 ユーザーごとに適切な権限を設定してください。

JP1 ユーザーの登録例を次の表に示します。

表 6-2　JP1 ユーザーの登録例

JP1 ユーザー名	使用目的
jp1admin	システム全体を管理するために使用します。 JP1/AJS3，JP1/IM，および JP1/Base のすべての管理者権限が，デフォルトで設定されています。設定されている権限を次に示します。 jp1admin：＊=JP1_AJS_Admin，JP1_JPQ_Admin，JP1_Console_Admin JP1_AJS_Admin：JP1/AJS3 の管理者権限 JP1_JPQ_Admin：JP1/AJS3 のジョブ実行環境やキューなどの管理者権限，エージェント管理情報の管理者権限 JP1_Console_Admin：JP1/IM の管理者権限 ジョブネットの定義や実行といった通常の業務に使用する JP1 ユーザーとしては，「jp1admin」ではなく「jp1mngr」を使用することを推奨します。
jp1mngr	ジョブネットを管理したり（業務管理者用ユーザー），ジョブを実行したり（ジョブ実行用ユーザー）するために使用します。 このユーザーには，すべてのジョブネットを更新・実行・監視できる権限を設定します。また，OS やプログラムなどの仕様上，Administrators 権限またはスーパーユーザー権限を持つユーザー以外は実行できないジョブを実行できる権限を設定します。権限の設定例を次に示します。 jp1mngr：＊=JP1_AJS_Manager，JP1_JPQ_Operator JP1_AJS_Manager：ジョブネットの編集・実行権限 JP1_JPQ_Operator：ジョブ実行環境やキューなどの操作権限，ジョブの操作権限，エージェント管理情報の操作権限
jp1user	ジョブを実行（ジョブ実行用ユーザー）するために使用します。 運用管理を効率的に実施するため，マネージャーホスト上でのジョブ実行ユーザーは統一しておいてください。権限の設定例を次に示します。 jp1user：＊=JP1_AJS_Manager，JP1_JPQ_Operator JP1_AJS_Manager：ジョブネットの編集・実行権限 JP1_JPQ_Operator：ジョブ実行環境やキューなどの操作権限，ジョブの操作権限，エージェント管理情報の操作権限
jp1ope	ジョブネットを監視したり，手動で再実行したりするために使用します。オペレーター向けの業務監視，および手動操作用のユーザーです。権限の設定例を次に示します。 jp1ope：＊=JP1_AJS_Operator JP1_AJS_Operator：ジョブネットの実行権限，ジョブネットの参照権限
jp1guest	ジョブネットの監視だけをするために使用します。オペレーター向けの業務監視ユーザーです。権限の設定例を次に示します。 jp1guest：＊=JP1_AJS_Guest

JP1 ユーザー名	使用目的
jp1guest	JP1_AJS_Guest：ジョブネットの参照権限
root	JP1/AJS3 のジョブ実行環境やキューなどを監視するために使用します。 jpqXXXX コマンドをコマンドプロンプトから実行する場合，コマンドを実行する OS ユーザーを，JP1 ユーザーとして登録する必要があります。この例で説明している，「root」という JP1 ユーザーは，UNIX の場合，OS ユーザー「root」のことです。Windows の場合は，OS ユーザー「Administrator」を，JP1 ユーザー「Administrator」として登録する必要があります。 権限の設定例を次に示します。 root：＊=JP1_JPQ_Admin JP1_JPQ_Admin：ジョブ実行環境やキューなどの管理者権限

JP1 ユーザーに設定できる権限の詳細については，「6.4.1(2) 設定する JP1 権限レベルの検討」を参照してください。

JP1 ユーザーの登録は省略できますが，ジョブを実行する場合は，JP1 ユーザーを登録してください。

JP1 ユーザーの登録を省略して，「6.4 設定するアクセス権限の検討」で説明するアクセス権限の設定を行う場合，OS ユーザーと同一名称の JP1 ユーザーが登録されていると仮定します。

6.4　設定するアクセス権限の検討

ここでは，JP1 ユーザーに割り当てるアクセス権限について検討します。

6.4.1　アクセス権限についての検討項目

登録する JP1 ユーザーの検討が終了したら，その JP1 ユーザーに，「どのようなグループ（JP1 資源グループ）に対して，どのようなアクセス権限（JP1 権限レベル）を許可するのか」ということを検討してください。

(1)　定義する JP1 資源グループの検討

JP1/AJS3 のユニットのアクセス権限は，幾つかのグループに分けて管理します。このグループを **JP1 資源グループ**と呼びます。JP1 資源グループには，任意の名称を定義できます。例えば，資材部や人事部などの部署ごとに，「purchasingdep」や「personneldep」というような，部署を意味する名称を定義できます。

各 JP1 資源グループには，それぞれアクセス権限を設定できます。JP1 資源グループに対するアクセス権限は，JP1 ユーザーに対して関連づけます。JP1 ユーザーは，関連づけられた JP1 資源グループのユニットに対して，設定されたアクセス権限の範囲内だけで操作できるようになります。

どの JP1 資源グループにも所属していないユニットは，アクセス制御の対象になりません。そのため，すべての JP1 ユーザーが参照したり，変更したりできてしまいます。また，どの JP1 ユーザーにも関連づけられていない JP1 資源グループは，どの JP1 ユーザーからもアクセスできなくなってしまいます。JP1/AJS3 を運用していく上で，すべての JP1 ユーザーがすべてのユニットに対して適切なアクセス権限を保持できるように，次のこと検討しておくことを推奨します。

- すべてのユニットに，JP1 資源グループを設定してあるか
- すべての JP1 資源グループは，適切なアクセス権限を設定した上で JP1 ユーザーと関連づけてあるか
- すべての JP1 ユーザーに，適切な JP1 資源グループが関連づけられているか

また，一人の JP1 ユーザーに対して複数の JP1 資源グループを関連づけることもできます。例えば，一人の JP1 ユーザーに，総務部の JP1 資源グループと営業部の JP1 資源グループを関連づけられます。これによって，一人の JP1 ユーザーに対する総務部と営業部の JP1 資源グループのアクセス権限を，別々に制御できます。

(2)　設定する JP1 権限レベルの検討

JP1 資源グループに対して，どのような操作ができるかを表したものを **JP1 権限レベル**と呼びます。

JP1 権限レベルは，JP1 ユーザーに関連づけられている JP1 資源グループごとに設定します。例えば，一人の JP1 ユーザーに総務部の JP1 資源グループと営業部の JP1 資源グループが関連づけられている場合，総務部の JP1 資源グループにはユニットを実行・編集できる JP1 権限レベルを，営業部の JP1 資源グルー

プにはユニットを参照できるJP1権限レベルを設定できます。この場合，このJP1ユーザーは，総務部のJP1資源グループのユニットを実行・編集できる権限と，営業部のJP1資源グループのユニットを参照できる権限を持ちます。

JP1権限レベルには次の3種類があります。

- ジョブネット定義・実行時のアクセス権限
- QUEUEジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時のアクセス権限
- エージェント管理情報操作時のアクセス権限

それぞれのJP1権限レベル名と操作できる内容を以降に示します。これらの内容を参照して，設定するJP1権限レベルを検討してください。

(a) ジョブネット定義・実行時のアクセス権限

ジョブネット定義・実行時のアクセス権限には，次の5種類があります。

- JP1_AJS_Admin
  管理者権限です。ユニットの所有者やJP1資源グループの操作権限，ジョブネットの定義・実行・編集権限などを持っています。
- JP1_AJS_Manager
  ジョブネットの定義・実行・編集権限などを持っています。
- JP1_AJS_Editor
  ジョブネットの定義・編集権限などを持っています。
- JP1_AJS_Operator
  ジョブネットの実行・参照権限などを持っています。
- JP1_AJS_Guest
  ジョブネットの参照権限などを持っています。

ジョブネット定義・実行時のJP1権限レベル名と詳細な操作内容を次の表に示します。

表6-3 ジョブネット定義・実行時のJP1権限レベル名と操作できる内容

操作内容	JP1_AJS_Admin	JP1_AJS_Manager	JP1_AJS_Editor	JP1_AJS_Operator	JP1_AJS_Guest
所有者権限※1を持たないユニットの所有者，JP1資源グループ名，またはジョブの実行ユーザー種別を変更する	○	－	－	－	－
ユニットを定義する	○	○	○	－	－
ジョブネット配下のユニットの定義内容を変更する	○	○※2	○※2	－	－

操作内容	JP1_AJS_Admin	JP1_AJS_Manager	JP1_AJS_Editor	JP1_AJS_Operator	JP1_AJS_Guest
ジョブネットの定義内容を変更する	○	○	○	－	－
ユニットをコピー・移動したり，名称を変更したりする※3	○	○	○	－	－
ユニットを削除する※4	○	○	○	－	－
ユニット名を標準出力ファイルに出力する	○	○	○	○	○
ユニットの定義内容を標準出力ファイルに出力する	○	○	○	○	○
ユニットを退避する	○	○	○	○	○
ユニットを回復する	○	○	○	－	－
ジョブグループにカレンダー情報を定義する	○	○	○	－	－
特定期間について，ジョブネットの実行スケジュールを定義する	○	○	○	○	○
定義済みのジョブネットを実行登録する	○	○	－	○	－
ジョブネットの実行登録を解除する	○	○	－	○	－
ジョブネットをリリース登録する	○	○	○※5	○※5	－
ジョブネットのリリースを中止する	○	○	○※5	○※5	－
ジョブネットのリリース情報を参照する	○	○	○	○	○
ジョブネットやジョブの実行履歴，現在の状態，次回実行予定などを標準出力ファイルに出力する	○	○	○	○	○
ジョブネットに定義されたスケジュールを一時的に変更する	○	○	－	○	－
ジョブの状態を一時的に変更する	○	○	－	○	－
ジョブの状態を変更する	○	○	－	○	－
ジョブネットの実行を中断する	○	○	－	○	－

操作内容	JP1_AJS_Admin	JP1_AJS_Manager	JP1_AJS_Editor	JP1_AJS_Operator	JP1_AJS_Guest
ジョブネットを再実行する	○	○	－	○	－
ジョブやジョブネットの実行を強制終了させる	○	○	－	○	－
ユニットをエクスポートする	○	○	○	○	○
ユニットをインポートする	○	○	○	－	－
ジョブネットの実行登録状態をエクスポートする	○	○	○	○	○
ジョブネットの実行登録状態をインポートする	○	○	－	○	－

（凡例）

○：操作できる。

－：操作できない。

注※1

ユニットに所有者が設定されていない場合は，すべてのユーザーが JP1 資源グループ名，所有者，および実行ユーザー種別を変更できます。

ユニットの所有者権限については，「(3) ユニットの所有者権限」を参照してください。

注※2

ジョブの詳細定義を変更する場合，ジョブの実行ユーザー種別によって変更できる項目が異なります。詳細については，「(6) ジョブの定義変更とアクセス権限」を参照してください。

注※3

コピー・移動したり，名称を変更したりするユニットの上位ユニットが操作対象ユニットです。例えば，/AAA/BBB/CCC というユニットを移動する場合，ユニット/AAA/BBB に設定されている JP1 資源グループに対し，該当する権限が必要です。コピー・移動したり，名称を変更したりするユニットに対しては，次のような権限が必要です。

コピーの場合は，コピー元ユニット（配下のユニットも含む）に設定されている JP1 資源グループに対し，JP1_AJS_Admin，JP1_AJS_Manager，JP1_AJS_Editor，JP1_AJS_Operator，または JP1_AJS_Guest の権限が必要です。

移動の場合は，移動するユニットに設定されている JP1 資源グループに対し，JP1_AJS_Admin，JP1_AJS_Manager，または JP1_AJS_Editor の権限が必要です。

名称の変更の場合は，名称を変更するユニットに設定されている JP1 資源グループに対し，JP1_AJS_Admin，JP1_AJS_Manager，または JP1_AJS_Editor の権限が必要です。

注※4

削除するユニットの上位ユニットも操作対象ユニットです。例えば，/AAA/BBB/CCC というユニットを削除する場合，ユニット/AAA/BBB/CCC（配下のユニットも含む）に設定されている JP1 資源グループおよびユニット/AAA/BBB に設定されている JP1 資源グループに対し，該当する権限が必要です。

注※5

JP1_AJS_Editor および JP1_AJS_Operator の両方の権限が必要です。

これは，リリース登録およびリリース中止の操作によって，ジョブネットの定義変更と，定義変更されたジョブネットの実行登録が行われるためです。

ユニットを操作する JP1 ユーザーには，操作対象となるユニットに設定されている JP1 資源グループに対して該当の操作権限が必要です。また，上位ユニットに設定されている JP1 資源グループに対しては，JP1_AJS_Admin，JP1_AJS_Manager，JP1_AJS_Editor，JP1_AJS_Operator，または JP1_AJS_Guest の権限が必要です。

ユニットを操作する JP1 ユーザーにマッピングされている OS ユーザーが Administrators 権限またはスーパーユーザー権限を持つユーザーの場合，JP1 権限レベルに関係なく，すべての操作を実行できます。ユニットを操作するコマンドを実行する場合も，環境変数JP1_USERNAME に設定した JP1 ユーザーの権限に関係なく，すべての操作を実行できます。ただし，環境設定パラメーターADMACLIMIT をデフォルトから変更して「yes」に設定しているときは，実行できる操作は JP1 権限レベルの範囲の操作だけです。

環境設定パラメーターADMACLIMIT の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.11.2(4) ADMACLIMIT」を参照してください。

ユニットに JP1 資源グループが設定されていない場合は，そのユニットに対して，すべてのユーザーが JP1/AJS3 のすべての操作を実行できます。

注意事項

1. マネージャージョブグループ，マネージャージョブネットでは，参照先の JP1/AJS3 - Manager のアクセス権限が適用されます。
2. JP1/AJS3 - View が接続している間，JP1/AJS3 - Manager 上で，アクセス権限の情報をキャッシュに保持しています。このため，アクセス権限の変更が，接続中の JP1/AJS3 - View に反映されないことがあります。アクセス権限を変更する場合には，接続中の JP1/AJS3 - View をログアウトし，アクセス権限を変更後，再度ログインしてください。

(b) QUEUE ジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時のアクセス権限

QUEUE ジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時のアクセス権限には，次の 3 種類があります。

- JP1_JPQ_Admin

  管理者権限です。ジョブ実行環境の設定権限，キューやジョブ実行先エージェントの操作権限，ほかのユーザーがキューイングしたジョブの操作権限を持っています。

- JP1_JPQ_Operator

  キューやジョブ実行先エージェントの操作権限，ほかのユーザーがキューイングしたジョブの操作権限を持っています。

- JP1_JPQ_User

サブミットジョブの登録や，自分がキューイングしたジョブの操作権限を持っています。

QUEUE ジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時のアクセス権限の設定では，JP1 資源グループ「JP1_Queue」に対して，これらの JP1 権限レベルを持たせるように設定します。「JP1_Queue」は，大文字・小文字を間違えないように入力してください。

QUEUE ジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時の JP1 権限レベル名と詳細な操作内容を次の表に示します。

表 6-4　QUEUE ジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時の JP1 権限レベル名と詳細な操作内容

操作内容	JP1_JPQ_Admin	JP1_JPQ_Operator	JP1_JPQ_User
サブミットジョブを登録する	○	○	○
ジョブの実行をキャンセル，または強制終了する	○	○	△
ジョブの実行を保留，または保留解除する	○	○	△
ジョブを移動する	○	○	△
ジョブ情報を出力する	○	○	△
終了ジョブ情報を出力する	○	○	△
データベースから終了ジョブ情報を削除する	○	○	－
キューを開く	○	○	－
キューを閉じる	○	○	－
キューを追加する	○	－	－
キューを削除する	○	－	－
キュー情報を出力する	○	○	○
キューの定義を変更する	○	－	－
キューをエージェントに接続する	○	－	－
キューとエージェントの接続を解除する	○	－	－
ジョブ実行多重度を変更する	○	－	－
エージェントを追加する	○	－	－
エージェントを削除する	○	－	－
エージェントホスト情報を出力する	○	－	－
排他実行リソースを追加する	○	－	－
排他実行リソースを削除する	○	－	－
排他実行リソース情報を出力する	○	○	○

（凡例）

○：操作できる。

△：操作できるが，ほかのユーザーが実行したジョブは操作できない。

－：操作できない。

注意事項

QUEUE ジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時は，処理要求先マネージャーのアクセス権限の定義が適用されます。

(c)　エージェント管理情報操作時のアクセス権限

エージェント管理情報操作時のアクセス権限には，次の 3 種類があります。

- JP1_JPQ_Admin
  管理者権限です。実行エージェントや実行エージェントグループの定義を追加・変更・削除する権限を持っています。
- JP1_JPQ_Operator
  実行エージェントや実行エージェントグループの受付配信制限の状態を変更する権限を持っています。
- JP1_JPQ_User
  実行エージェントや実行エージェントグループの状態および定義内容を参照する権限を持っています。

エージェント管理情報操作時のアクセス権限の設定では，JP1 資源グループ「JP1_Queue」に対して，これらの JP1 権限レベルを持たせるように設定します。「JP1_Queue」は，大文字・小文字を間違えないように入力してください。

エージェント管理情報操作時の JP1 権限レベル名と詳細な操作内容を次に示します。

表 6-5　エージェント管理情報操作時の JP1 権限レベル名と詳細な操作内容

操作内容	JP1_JPQ_Admin	JP1_JPQ_Operator	JP1_JPQ_User
実行エージェントを追加する	○	－	－
実行エージェントグループを追加する	○	－	－
実行エージェントを削除する	○	－	－
実行エージェントグループを削除する	○	－	－
実行エージェントの実行ホストを変更する	○	－	－
実行エージェントのジョブ実行多重度を変更する	○	－	－
実行エージェントの説明文を変更する	○	－	－
実行エージェントグループの説明文を変更する	○	－	－
実行エージェントグループに接続する実行エージェントを追加する	○	－	－

操作内容	JP1_JPQ_Admin	JP1_JPQ_Operator	JP1_JPQ_User
実行エージェントグループに接続されている実行エージェントの優先順位を変更する	○	－	－
実行エージェントグループに接続されている実行エージェントを接続先から解除する	○	－	－
実行エージェントの受付配信制限の状態を変更する	○	○	－
実行エージェントグループの受付配信制限の状態を変更する	○	○	－
実行エージェントの状態を表示する※	○	○	○
実行エージェントグループの状態を表示する※	○	○	○
すべての実行エージェントと実行エージェントグループの状態を表示する※	○	○	○
すべての実行エージェントと実行エージェントグループの名称を表示する※	○	○	○
実行エージェントの定義を出力する※	○	○	○
実行エージェントグループの定義を出力する※	○	○	○
すべての実行エージェントと実行エージェントグループの定義を出力する※	○	○	○

（凡例）

○：操作できる。

－：操作できない。

注※

Administrators 権限またはスーパーユーザー権限を持つユーザーは，JP1 権限レベルに関係なく操作できます。

(3)　ユニットの所有者権限

ジョブやジョブネットを定義したユーザーは，そのジョブやジョブネットの所有者として設定され，所有者権限を持ちます。所有者権限を持つと，JP1 権限レベルに関係なく，次の設定を変更できます。

- JP1 資源グループ名
- 所有者
- 実行ユーザー種別（ジョブの場合）

ただし，所有者であっても，ユニットの JP1 資源グループに対して参照権限が与えられていない場合は，JP1/AJS3 - View でユニットの詳細定義ダイアログボックスを表示できません。そのため，JP1/AJS3 - View から JP1 資源グループ名，所有者，および実行ユーザー種別を変更できません。所有者のままこれ

らの設定を変更したい場合は，ajschange コマンドを実行して，JP1 資源グループを，所有者に参照権限が与えられている JP1 資源グループに変更してください。ajschange コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajschange」を参照してください。

なお，ユニットに所有者が設定されていない場合は，すべてのユーザーが JP1 資源グループ名，所有者，および実行ユーザー種別を変更できます。

(4)　ユニットの新規作成，コピー，またはリリース登録時のアクセス権限

ユニットを新規作成，コピー，またはリリース登録すると，ユニットの所有者および JP1 資源グループは，デフォルトでは次のように設定されます。

ユニットを JP1/AJS3 - View で新規作成した場合
　所有者：ユニットを作成した JP1 ユーザー
　JP1 資源グループ：上位ユニットの JP1 資源グループ

ユニットをajsdefine コマンドで新規作成した場合
　所有者：ユニット定義ファイルで設定した所有者
　JP1 資源グループ：ユニット定義ファイルで設定した JP1 資源グループ

ユニットを JP1/AJS3 - View またはajscopy コマンドでコピーした場合
　所有者：ユニットをコピーした JP1 ユーザー
　JP1 資源グループ：コピー元ユニットの JP1 資源グループ

ユニットを JP1/AJS3 - View またはajsrelease コマンドでリリース登録した場合
　所有者：リリース元ユニットの所有者
　JP1 資源グループ：リリース元ユニットの JP1 資源グループ

JP1/AJS3 - Definition Assistant で一括または単独でエクスポートしてユニットを新規作成した場合
　所有者：定義情報管理テンプレートで設定した所有者
　JP1 資源グループ：定義情報管理テンプレートで設定した JP1 資源グループ

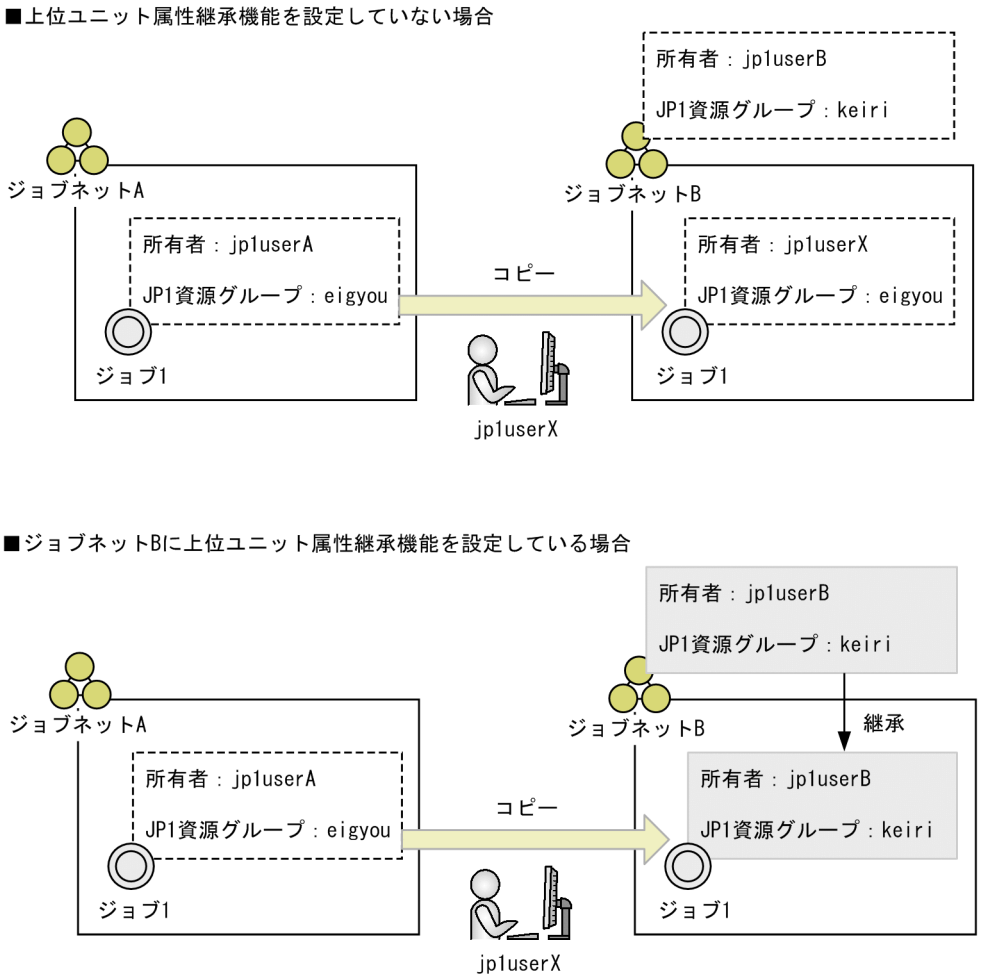
新規作成，コピー，またはリリース登録したユニットの所有者や JP1 資源グループを，上位のユニットから継承するように設定することもできます。この機能を，**上位ユニット属性継承機能**といいます。この機能を使用することで，システム内のユニットの所有者を統一して運用したい場合に，新規作成，コピー，またはリリース登録したユニットの所有者，および JP1 資源グループを変更する必要がなくなります。

(a)　上位ユニット属性継承機能の概要

上位ユニット属性継承機能は，ジョブグループまたはジョブネットに対して設定します。上位ユニット属性継承機能を設定したジョブグループまたはジョブネット配下に，ユニットを新規作成したり，コピーして作成したり，またはリリース先としてリリース登録したりすると，作成したユニットは上位のユニットの所有者や JP1 資源グループを継承します。なお，定義済みのユニットや，切り取って貼り付けたユニットの場合，所有者や JP1 資源グループは変更されません。

上位ユニット属性継承機能の例を，次の図に示します。

図 6-2　上位ユニット属性継承機能の例



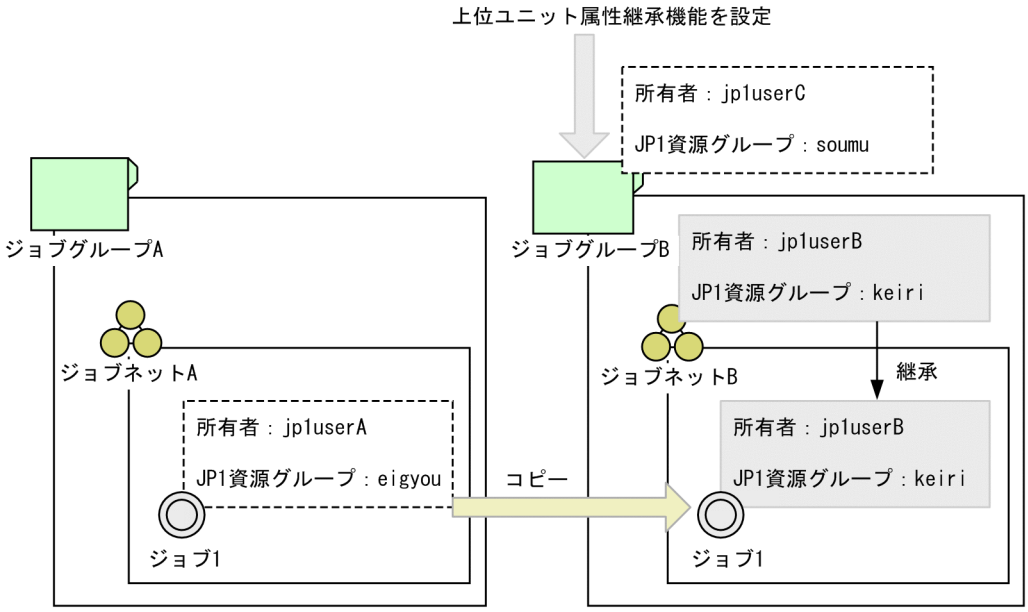
上位ユニット属性継承機能を設定していない場合，ジョブ 1 をジョブネット B の配下にコピーすると，コピーして作成されたジョブ 1 の所有者はコピー操作をした JP1 ユーザーの「jp1userX」が，JP1 資源グループはコピー元ユニットの「eigyou」が設定されます。

上位ユニット属性継承機能を設定している場合，ジョブ 1 をジョブネット B の配下にコピーすると，コピーして作成されたジョブ 1 の所有者および JP1 資源グループは，ジョブネット B の設定を継承します。

2 階層以上のユニット構成で上位ユニット属性継承機能を設定している場合，新規作成したユニットおよびコピーして作成したユニットの所有者および JP1 資源グループは，作成したユニットの一つ上の階層のユニットの所有者および JP1 資源グループを継承します。

2 階層以上のユニット構成で上位ユニット属性継承機能を設定している場合の例を，次の図に示します。

図 6-3　2 階層以上のユニット構成で上位ユニット属性継承機能を設定している場合の例



この例では，ジョブグループ B に上位ユニット属性継承機能を設定しています。ジョブネット A の配下にあるジョブ 1 をジョブネット B の配下にコピーすると，コピーして作成したジョブ 1 の所有者および JP1 資源グループの設定は，作成したユニットの一つ上の階層のユニットであるジョブネット B の設定を継承します。ジョブグループ B の設定は継承しません。

注意事項

- 上位ユニット属性継承機能は，バージョン 09-50 以降の JP1/AJS3 - Manager で使用できます。ただし，バージョン 09-50 以降の JP1/AJS3 - Manager にバージョン 09-10 以前の JP1/AJS3 - View で接続している場合，コピーしたユニットは上位ユニットの設定を継承できますが，新規作成したユニットは上位ユニットの設定を継承できません。
- `ajsrestore` コマンドまたは JP1/AJS3 - View によるユニットの回復，および`ajsimport` コマンドによるユニットのインポートは，元の定義の復元を目的としています。そのため，復元先の上位ユニットに上位ユニット属性継承機能を設定していても，上位ユニット属性継承機能は有効になりません。
- ジョブネットをリリース登録する場合，リリース先の上位ユニットに上位ユニット属性継承機能を設定すると，上位ユニット属性継承機能の設定が優先されます。リリース元のユニットの所有者，および JP1 資源グループを優先する場合は，リリース先の上位ユニットに上位ユニット属性継承機能を設定しないか，リリース元とリリース先は同じユニット属性になるようユニット属性プロファイルを定義してください。
- コピーして作成するユニットのユニット完全名が，上位ユニット属性継承機能を設定するユニットの完全名と同じ場合，コピーして作成するユニットは，コピー元ユニットの所有者および JP1 資源グループの設定が維持されます。

(b) 上位ユニット属性継承機能の設定

上位ユニット属性継承機能は，ユニット属性プロファイルで設定します。ユニット属性プロファイルには，上位ユニット属性継承機能を設定するユニットの完全名，および上位ユニット属性継承機能の設定方法を設定します。

- 上位ユニット属性継承機能を設定するユニットの完全名

  上位ユニット属性継承機能を設定するジョブグループまたはジョブネットの完全名を指定します。スケジューラーサービス配下のすべてのユニットに設定する場合は，「/」を指定します。

  なお，ここで指定するユニットは，実行ユーザー固定機能を設定するユニットと共通です。実行ユーザー固定機能については，「(5) ジョブの実行ユーザー」を参照してください。

- 上位ユニット属性継承機能の設定方法

  上位ユニット属性継承機能の設定方法を，次のどちらかの値で指定します。

  entryuser

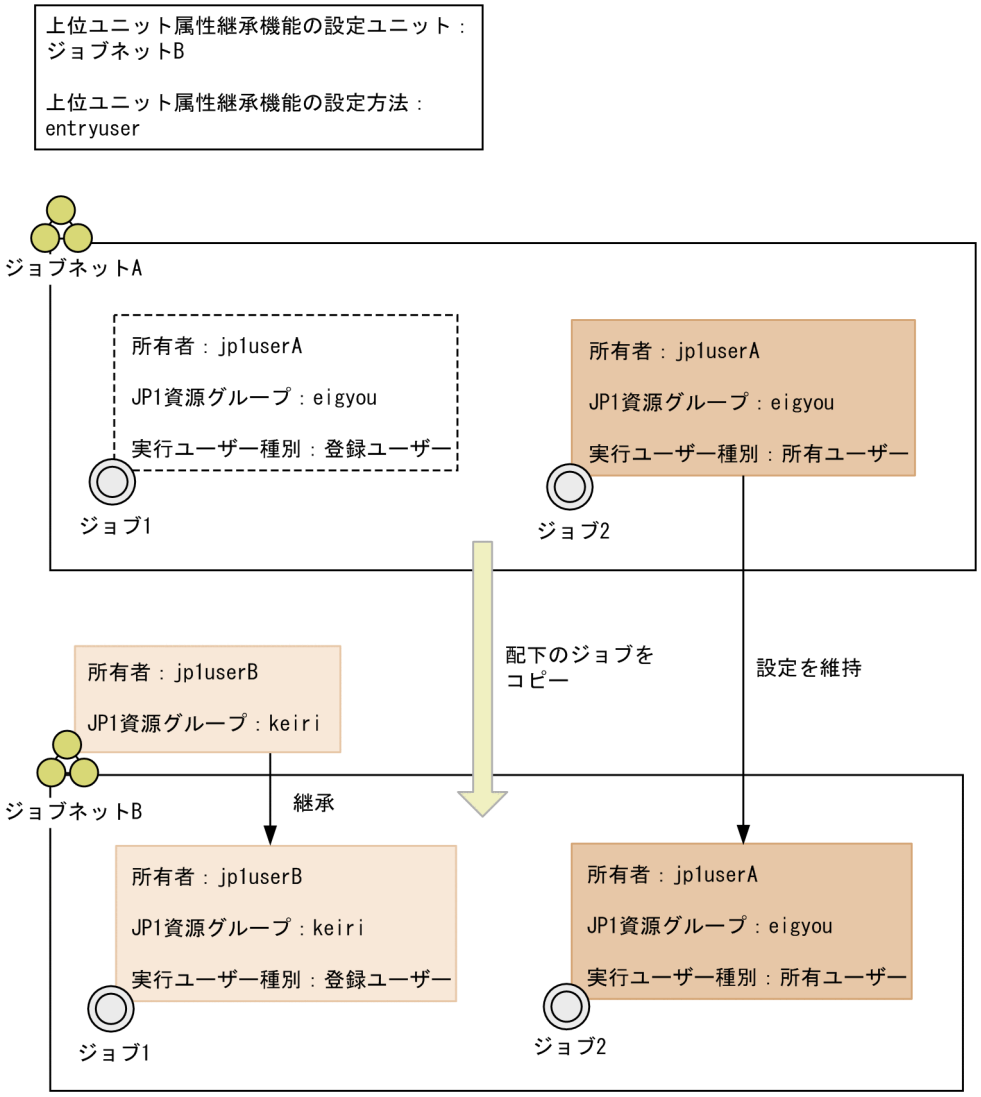
  上位ユニット属性継承機能を設定したジョブグループまたはジョブネットの配下のユニットが，所有者および JP1 資源グループを上位ユニットから継承します。ただし，実行ユーザー種別が所有ユーザーに設定されているジョブをコピーして作成した場合は，上位ユニットの設定を継承しないで，コピー元ユニットの設定を維持します。

  all

  上位ユニット属性継承機能を設定したジョブグループまたはジョブネットの配下のユニットが，所有者および JP1 資源グループを上位ユニットから継承します。実行ユーザー種別が所有ユーザーに設定されているジョブをコピーして作成した場合でも，上位ユニットの設定を継承します。

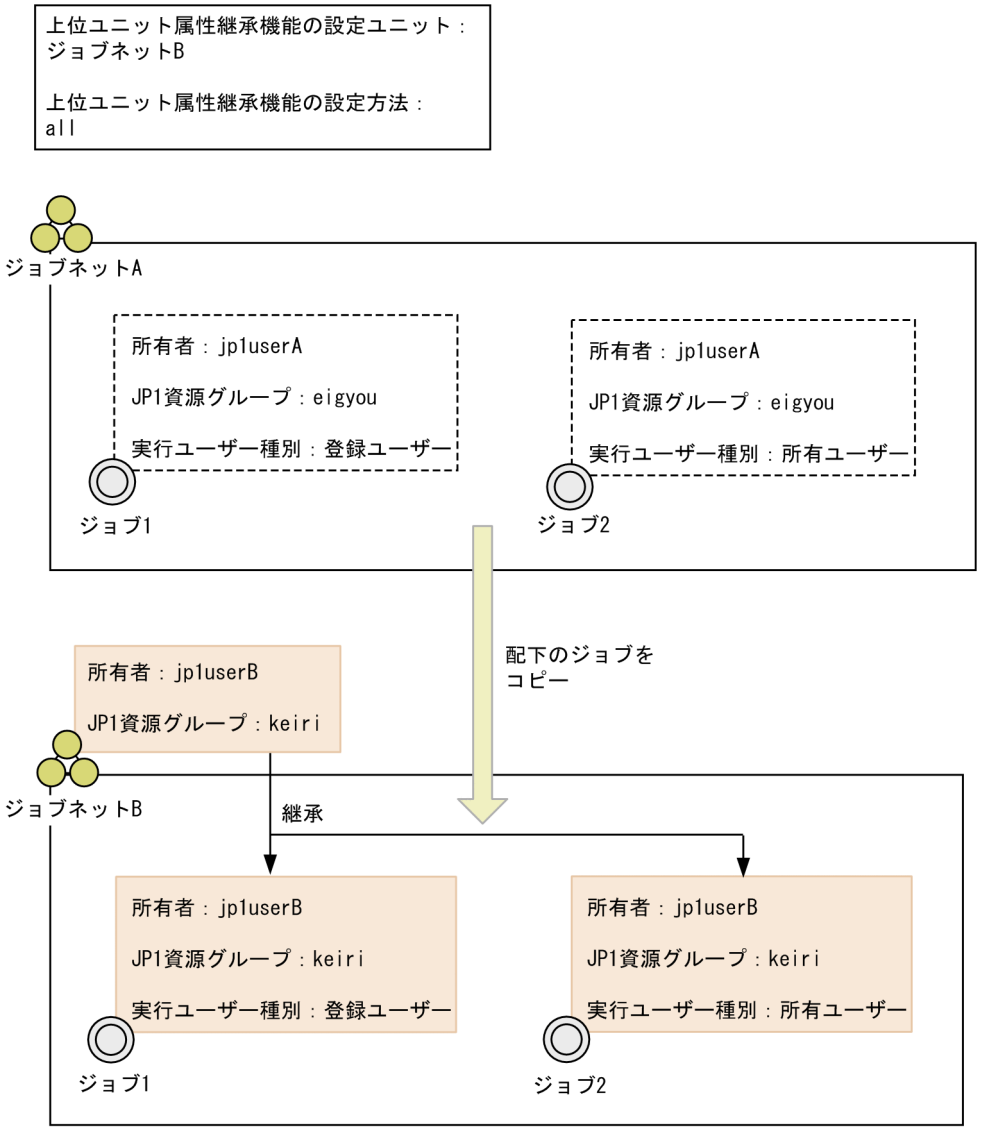
  上位ユニット属性継承機能の設定方法の違いを，次の図に示します。

図 6-4　上位ユニット属性継承機能の設定方法の違い（上位ユニット属性継承機能の設定方法が entryuser の場合）



この例では，上位ユニット属性継承機能の設定方法を entryuser にしています。ジョブ 1 およびジョブ 2 をジョブネット B 配下にコピーすると，ジョブ 1 は実行ユーザー種別が登録ユーザーに設定されているため，ジョブネット B の所有者および JP1 資源グループの設定を継承します。一方，ジョブ 2 は実行ユーザー種別が所有ユーザーに設定されているため，コピー元ユニットの所有者および JP1 資源グループの設定を維持します。

図 6-5　上位ユニット属性継承機能の設定方法の違い（上位ユニット属性継承機能の設定方法が all の場合）



この例では，上位ユニット属性継承機能の設定方法を all にしています。ジョブ 1 およびジョブ 2 をジョブネット B 配下にコピーすると，ジョブ 1 およびジョブ 2 は両方ともジョブネット B の所有者および JP1 資源グループの設定を継承します。

補足事項

同一ユニット配下で，異なる階層にある複数のユニットに対して上位ユニット属性継承機能を設定している場合，その配下のユニットは，最も近い上位のユニットの設定方法を優先します。例えば，ジョブグループとそのジョブグループ配下にあるジョブネットの両方に上位ユニット属性継承機能の設定している場合，ジョブネット配下のジョブは，ジョブネットの設定方法を優先します。

ユニット属性プロファイルの設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　21.1.3 上位ユニット属性継承機能と実行ユーザー固定機能の設定手順」を参照してください。ユニット属性プロファイルの詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　21.1.4 ユニット属性プロファイルの詳細」を参照してください。

(c) 上位ユニット属性継承機能の使用例

上位ユニット属性継承機能の使用例について説明します。

ここでは，次のような方針でジョブネットを定義することを想定します。

アクセス権限の定義方針

すべてのユニットのアクセス権（所有者および JP1 資源グループ）は，所有者「user1」，JP1 資源グループ「keiri」に統一する。

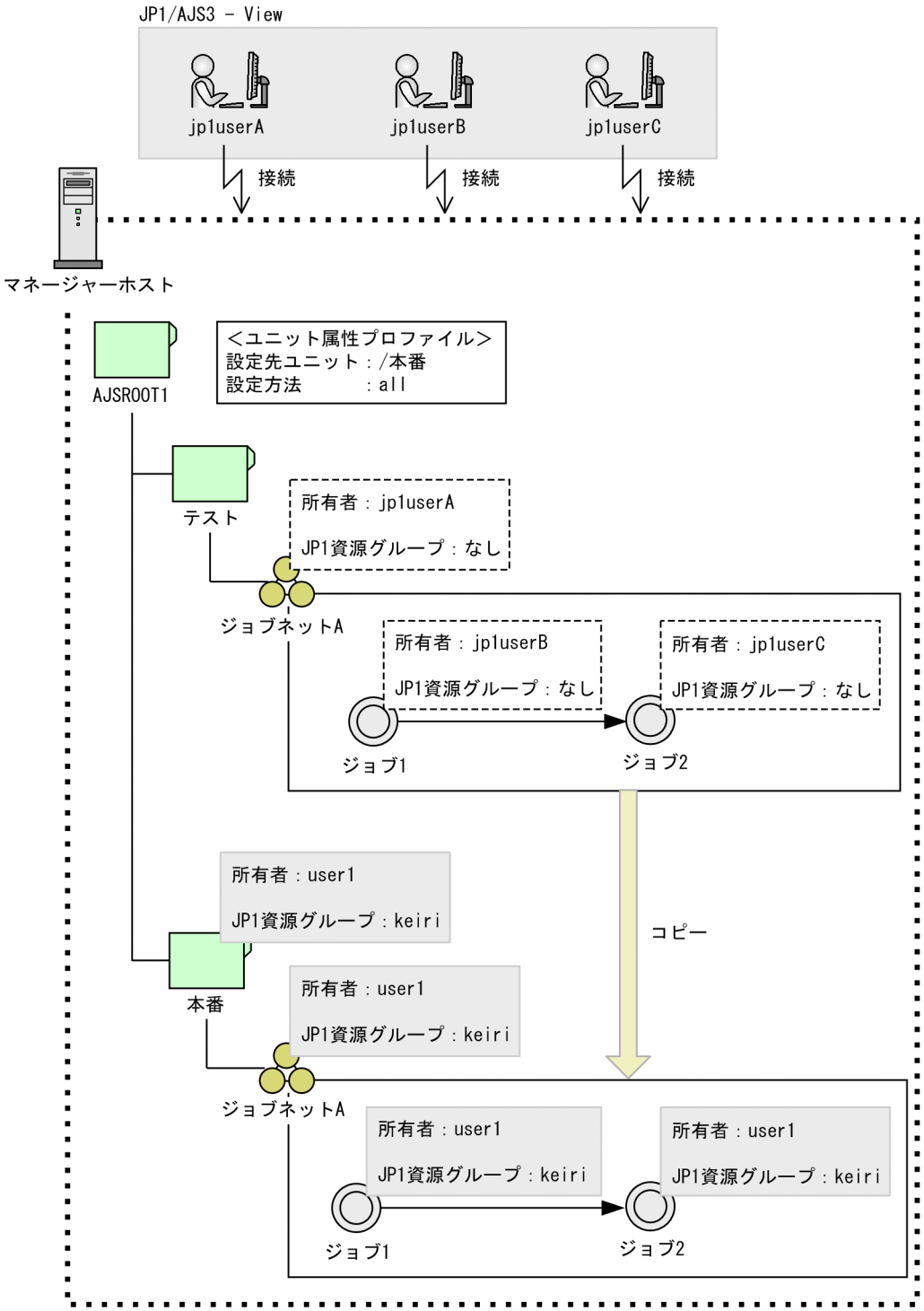
ユニットの定義方針

- マネージャーホストに複数の JP1 ユーザーが JP1/AJS3 - View で接続し，それぞれの JP1 ユーザーが担当業務のユニットを定義する。
- テスト用のジョブグループ「テスト」でジョブネット A を作成し，配下にジョブ 1 とジョブ 2 を定義する。
- ジョブネット A をジョブグループ「テスト」でテストしたあと，ジョブネット A を本番用のジョブグループ「本番」の配下にコピーし，運用を開始する。

なお，マネージャーホスト環境は構築済みであり，ジョブグループ「テスト」および「本番」は作成済みとします。また，ジョブグループ「本番」の所有者には「user1」を，JP1 資源グループには「keiri」を設定します。

このようなジョブネットを，上位ユニット属性継承機能を使用して定義する例を，次の図に示します。

図 6-6　上位ユニット属性継承機能の使用例



この例では，JP1 ユーザーの jp1userA，jp1userB，および jp1userC が，それぞれ JP1/AJS3 - View でマネージャーホストに接続しています。AJSROOT1 のユニット属性プロファイルには，上位ユニット属性継承機能を設定するユニットとして「/本番」を，上位ユニット属性継承機能の設定方法には「all」を指定します。

マネージャーホストのテスト用のジョブグループ「テスト」に，それぞれの JP1 ユーザーが，ジョブネット A，ジョブ 1，およびジョブ 2 を定義します。このとき，定義したユニットの所有者は，定義した JP1

ユーザー名が設定されます。ジョブグループ「テスト」でジョブネット A をテストしたあと，ジョブネット A をジョブグループ「本番」にコピーします。コピーして作成したジョブネット A は，上位ユニットであるジョブグループ「本番」のアクセス権を継承します。同様に，コピーして作成したジョブ 1 およびジョブ 2 も，上位ユニットであるジョブネット A のアクセス権，つまりジョブグループ「本番」のアクセス権を継承します。そのため，ジョブグループ「本番」の配下にコピーしたすべてのユニットは，所有者および JP1 資源グループを統一して運用できます。

(5) ジョブの実行ユーザー

ジョブネットを実行したとき，配下に定義されているジョブを実行する JP1 ユーザーのことを，ジョブの**実行ユーザー**といいます。

ジョブの実行ユーザーは，次の二つの種別から選択します。

登録ユーザー

ジョブネットを実行登録した JP1 ユーザーがジョブを実行します。実行登録した JP1 ユーザーにジョブの実行権限が与えられていれば，ジョブを実行できます。

所有ユーザー

ジョブの所有者に設定されている JP1 ユーザーがジョブを実行します。実行登録した JP1 ユーザーにジョブの実行権限が与えられていなくても，所有者に設定されている JP1 ユーザーにジョブの実行権限が与えられていれば，ジョブを実行できます。

OR ジョブ，判定ジョブ，およびイベントジョブは，JP1/AJS3 を起動しているユーザーの権限で実行されます。そのため，OR ジョブおよび判定ジョブでは実行ユーザー種別を指定できません。イベントジョブでは指定しても無効となります。

ジョブを実行する際，ジョブは実行先ホスト（エージェントホスト）に転送されて実行されます。そのため，ジョブの実行ユーザーは実行先ホストで OS ユーザーとユーザーマッピングしておく必要があります。実行ユーザー種別で登録ユーザーを指定した場合は，ジョブネットを実行登録する JP1 ユーザーを OS ユーザーとマッピングします。所有ユーザーを指定した場合は，ジョブの所有者を OS ユーザーとマッピングします。

ジョブの実行ユーザーを，ジョブに定義してある実行ユーザー種別とは関係なく，上位のユニットの所有者に設定してある JP1 ユーザーで固定してジョブを実行することもできます。実行ユーザーを上位ユニットの所有者に固定する機能を，**実行ユーザー固定機能**といいます。この機能を使用することで，ジョブネットを実行登録するユーザーに関係なく，すべてのジョブを一つの JP1 ユーザーで実行できるようになります。

注意事項

実行ユーザー固定機能は，バージョン 09-50 以降の JP1/AJS3 - Manager で使用できます。

(a) 実行ユーザー固定機能の概要

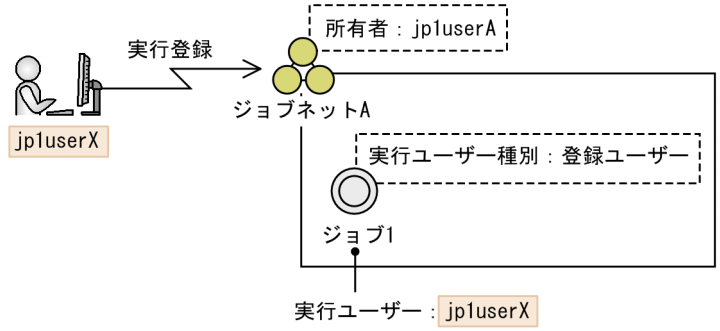
実行ユーザー固定機能は，ジョブグループまたはジョブネットに対して設定します。実行ユーザー固定機能が設定されたジョブグループまたはジョブネット配下で実行するジョブは，詳細定義に定義されている

実行ユーザー種別とは関係なく，実行ユーザー固定機能を設定したユニットの所有者に設定されている JP1 ユーザーの権限で実行されます。

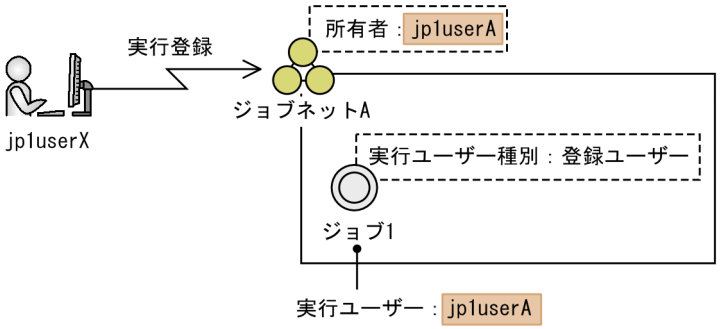
実行ユーザー固定機能の例を，次の図に示します。

図 6-7　実行ユーザー固定機能の例

■実行ユーザー固定機能を設定していない場合



■ジョブネットAに実行ユーザー固定機能を設定している場合



実行ユーザー固定機能を設定していない場合，JP1 ユーザー「jp1userX」がジョブネット A を実行登録すると，ジョブ 1 の実行ユーザー種別は登録ユーザーであるため，実行ユーザーはジョブネット A の登録ユーザー「jp1userX」になります。

ジョブネット A に実行ユーザー固定機能を設定している場合，JP1 ユーザー「jp1userX」がジョブネット A を実行登録すると，実行ユーザー種別とは関係なく，ジョブ 1 の実行ユーザーはジョブネット A の所有者である「jp1userA」になります。

注意事項

- 実行ユーザー固定機能を設定している場合，ジョブの詳細定義で設定している実行ユーザーと，実際の実行ユーザーが異なります。そのため，詳細定義でジョブを実行できるようにアクセス権限を定義しても，実行ユーザー固定機能で設定される実行ユーザーにジョブの実行権限がないと，ジョブ実行時に権限エラーになるので注意してください。
- 実行ユーザー固定機能を設定している場合，ジョブの定義情報は変更されません。ジョブ実行時には，ジョブの定義情報とは関係なく，実行ユーザー固定機能で設定された JP1 ユーザーでジョブが実行されます。

実行ユーザー固定機能は，次に示すジョブの上位ユニットに設定することで，実行ユーザーを固定できます。イベントジョブは実行ユーザーに依存しないため，対象外です。

- 標準ジョブ※
- アクションジョブ
- カスタムジョブ
- 引き継ぎ情報設定ジョブ
- HTTP 接続ジョブ

注※

UNIX ジョブまたは PC ジョブがキューレスジョブの場合でも，実行ユーザーを固定できます。

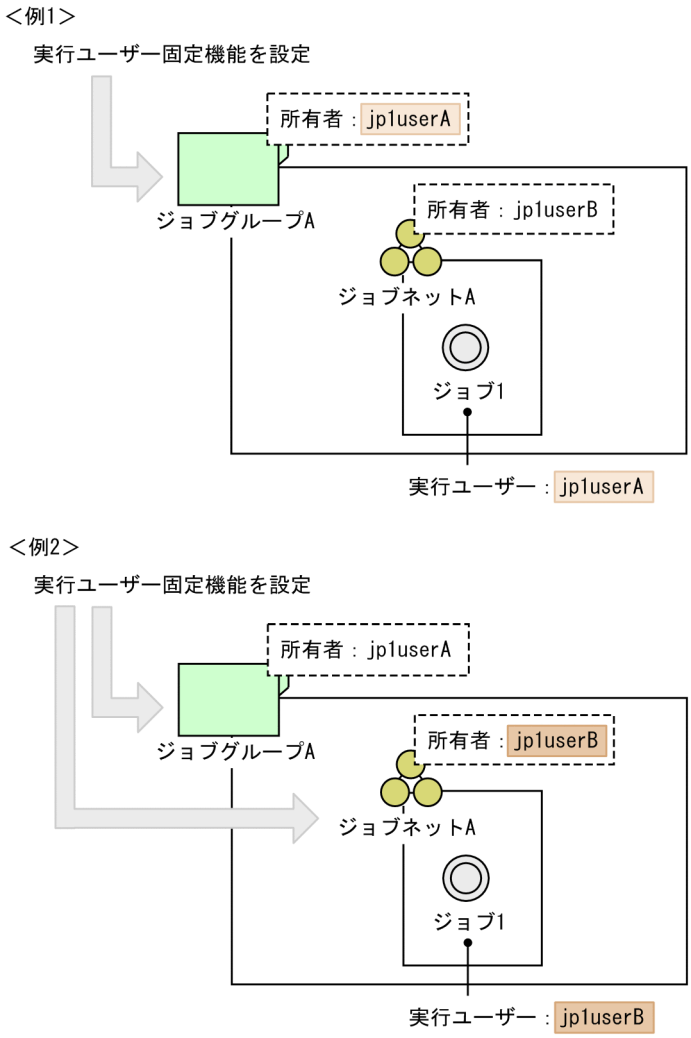
注意事項

リモートジョブネット配下のジョブの場合は，接続先ホストの実行ユーザー固定機能の設定が有効になります。

2 階層以上のユニット構成で実行ユーザー固定機能を設定している場合，ジョブの実行ユーザーとして固定される JP1 ユーザーは，実行ユーザー固定機能を設定しているユニットのうち，ジョブに最も近い階層のユニットの所有者になります。

2 階層以上のユニット構成で実行ユーザー固定機能を設定している場合の例を，次の図に示します。

図 6-8　2 階層以上のユニット構成で実行ユーザー固定機能を設定している場合の例



例 1 では，ジョブグループ A に実行ユーザー固定機能を設定しています。ジョブグループ A 配下のジョブネット A を実行登録すると，ジョブ 1 の実行ユーザーは，実行ユーザー固定機能を設定したジョブグループ A の所有者 jp1userA になります。

例 2 では，ジョブグループ A およびジョブネット A に実行ユーザー固定機能を設定しています。ジョブネット A を実行登録すると，ジョブ 1 の実行ユーザーは，実行ユーザー固定機能を設定したユニットのうち最も近い階層にあるジョブネット A の所有者 jp1userB になります。

(b)　実行ユーザー固定機能の設定

実行ユーザー固定機能は，ユニット属性プロファイルで設定します。ユニット属性プロファイルには，実行ユーザー固定機能を設定するユニットの完全名，および実行ユーザー固定機能の設定方法を設定します。

- 実行ユーザー固定機能を設定するユニットの完全名

  実行ユーザー固定機能を設定するジョブグループまたはジョブネットの完全名を指定します。スケジューラーサービス配下のすべてのユニットに設定する場合は，「/」を指定します。

なお，ここで指定するユニットは，上位ユニット属性継承機能を設定するユニットと共通です。上位ユニット属性継承機能については，「(4) ユニットの新規作成，コピー，またはリリース登録時のアクセス権限」を参照してください。

- 実行ユーザー固定機能の設定方法

  実行ユーザー固定機能の設定方法を，次のどちらかの値で指定します。

  entryuser

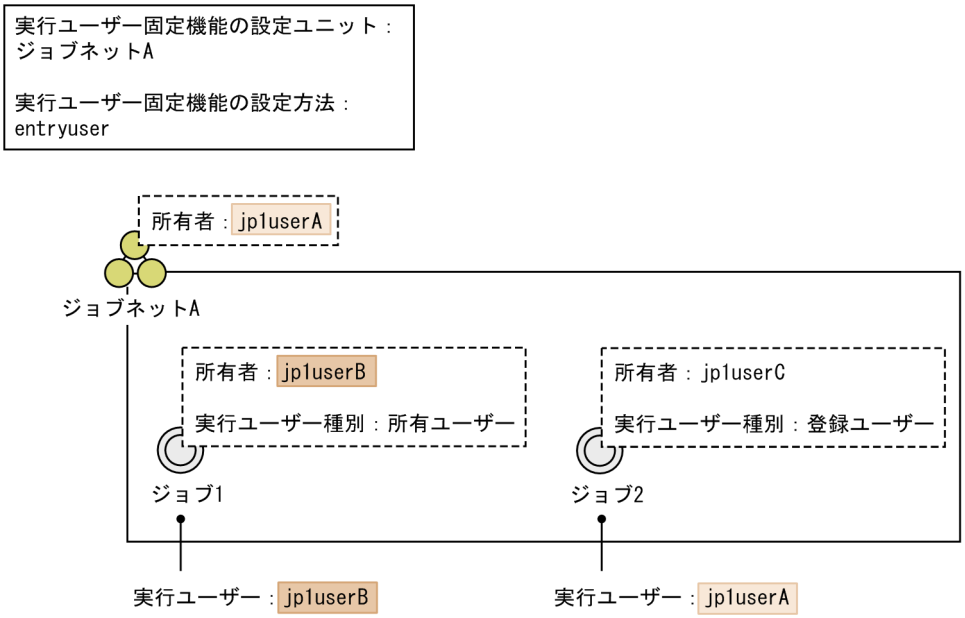
  実行ユーザー固定機能を設定したユニット配下にあるジョブの実行ユーザーを，実行ユーザー固定機能を設定したユニットの所有者で固定します。ただし，配下のジョブのうち実行ユーザー種別が所有ユーザーに設定されているジョブについては，それぞれのジョブの所有者が実行ユーザーになります。

  all

  実行ユーザー固定機能を設定したユニット配下にあるジョブの実行ユーザーを，実行ユーザー固定機能を設定したユニットの所有者で固定します。実行ユーザー種別が所有ユーザーのジョブでも，実行ユーザー固定機能を設定したユニットの所有者で固定します。

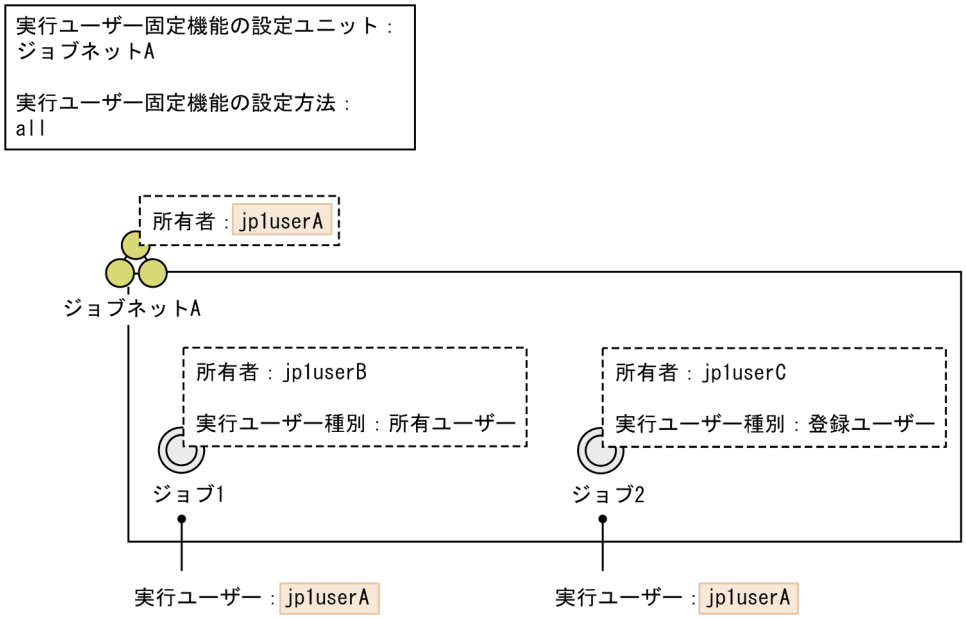
  実行ユーザー固定機能の設定方法の違いを，次の図に示します。

図 6-9　実行ユーザー固定機能の設定方法の違い（実行ユーザー固定機能の設定方法が entryuser の場合）



この例では，実行ユーザー固定機能の設定方法を entryuser に設定しています。ジョブネット A を実行登録すると，ジョブ 1 は実行ユーザー種別が所有ユーザーに設定されているため，実行ユーザーは所有者の jp1userB になります。一方，ジョブ 2 は実行ユーザー種別が登録ユーザーに設定されているため，実行ユーザーは，ジョブネット A の所有者である jp1userA に固定されます。

図 6-10　実行ユーザー固定機能の設定方法の違い（実行ユーザー固定機能の設定方法が all の場合）



この例では，実行ユーザー固定機能の設定方法を all に設定しています。ジョブネット A を実行登録すると，ジョブ 1 およびジョブ 2 の実行ユーザーは，ジョブネット A の所有者である jp1userA に固定されます。

補足事項

同一ユニット配下で，異なる階層にある複数のユニットに対して実行ユーザー固定機能を設定している場合，その配下のジョブは，最も近い上位のユニットの設定方法を優先します。例えば，ジョブグループとそのジョブグループ配下にあるジョブネットの両方に実行ユーザー固定機能の設定している場合，ジョブネット配下のジョブは，ジョブネットの設定方法を優先します。

ユニット属性プロファイルの設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　21.1.3 上位ユニット属性継承機能と実行ユーザー固定機能の設定手順」を参照してください。ユニット属性プロファイルの詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　21.1.4 ユニット属性プロファイルの詳細」を参照してください。

(c)　実行ユーザー固定機能の使用例

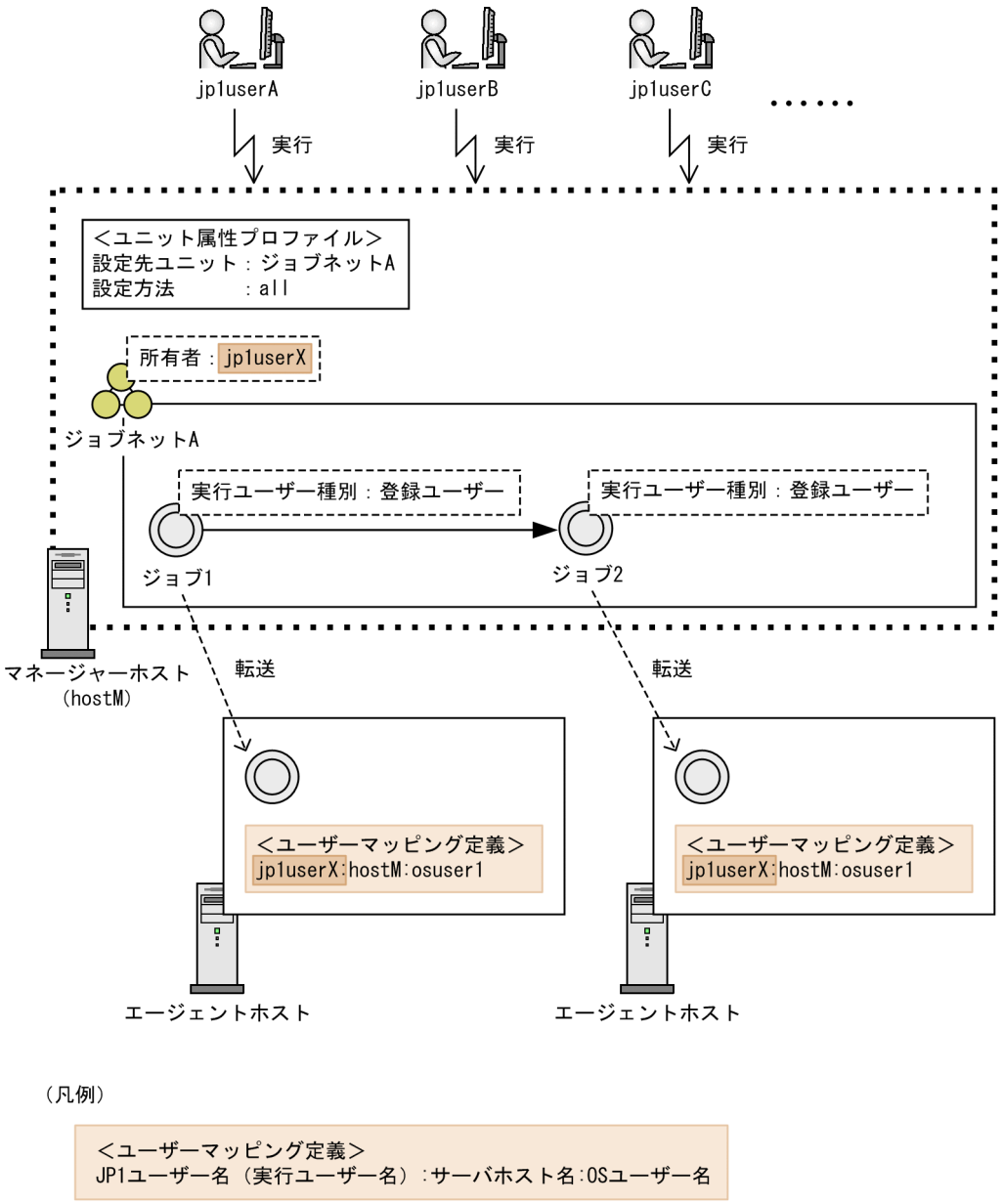
実行ユーザー固定機能の使用例について説明します。

ここでは，次のようなジョブネットを実行することを想定します。

- ジョブネット A はマネージャーホスト hostM に定義されていて，所有者は jp1userX である。
- ジョブネット A 配下にはジョブ 1 とジョブ 2 が定義されていて，実行ユーザー種別はどちらも登録ユーザーである。

このようなジョブネット A を，実行ユーザー固定機能を使用して実行する例を次に示します。

図 6-11　実行ユーザー固定機能の使用例



この例では，ジョブネット A に対して実行ユーザー固定機能を設定しています。実行ユーザー固定機能の設定方法は all です。ジョブネット A を jp1userA，jp1userB，または jp1userC のだれかが実行登録すると，ジョブ 1 およびジョブ 2 は実行先のエージェントホストに転送されます。このとき，登録ユーザーがだれであっても，ジョブ 1 およびジョブ 2 の実行ユーザーは，ジョブネット A の所有者 jp1userX に固定されます。そのため，ジョブの転送先のエージェントホストでジョブ 1 およびジョブ 2 を実行するには，jp1userX と OS ユーザー osuser1 のユーザーマッピングが必要になります。登録ユーザーである jp1userA，jp1userB，または jp1userC のユーザーマッピングは必要ありません。また，ジョブネット A を実行する JP1 ユーザーが追加されても，ユーザーマッピング定義を追加する必要はありません。

(d)　実行ユーザー固定機能と上位ユニット属性継承機能の組み合わせ

実行ユーザー固定機能と上位ユニット属性継承機能は，ともにユニット属性プロファイルを使って設定します。両方の機能を同時に使用する場合，それぞれの機能で設定方法を entryuser または all から選択し

ます。このとき，設定方法が異なる組み合わせで運用すると，ジョブ実行の動作がわかりにくくなります。そのため，両機能で設定方法は統一することを推奨します。

実行ユーザー固定機能と上位ユニット属性継承機能の設定方法の組み合わせを，次の表に示します。

表 6-6　実行ユーザー固定機能と上位ユニット属性継承機能の設定方法の組み合わせ

項番	実行ユーザー固定機能の設定方法	上位ユニット属性継承機能の設定方法	説明
1	entryuser	entryuser	特定のジョブだけ，実行ユーザーをそのジョブの所有者に固定できます。ジョブをコピーしても，設定は変更されません。 特殊なジョブを部品として別に作成し，コピーしてシステムに組み込むような運用の場合に指定します。
2	all	all	実行ユーザー，所有者，および JP1 資源グループを，指定したユニット配下で統一します。
3	entryuser	all	推奨しない組み合わせです。 特定のジョブだけ，実行ユーザーをそのジョブの所有者に固定できますが，そのジョブをコピーすると設定が変更されてしまいます。
4	all	entryuser	推奨しない組み合わせです。 特定のジョブをコピーした場合にコピー元のジョブの実行ユーザーを維持しようとしても，ジョブを実行する際に実行ユーザーが別のユニットの所有者に固定されてしまいます。

(6)　ジョブの定義変更とアクセス権限

JP1 ユーザーがジョブの詳細定義を変更する場合，アクセス権限の有無やジョブの定義内容によって，変更できる詳細定義の項目が異なります。次のどれかに該当する場合は，すべての詳細定義を変更できます。

- 環境設定パラメーター`ADMACLIMIT` を省略している，または環境設定パラメーター`ADMACLIMIT` に「`no`」を設定していて，操作する JP1 ユーザーが次の管理者権限を持つ OS ユーザーにマッピングされている。
  - Windows の場合
    Administrators 権限を持つ OS ユーザー
  - UNIX の場合
    スーパーユーザー権限を持つ OS ユーザー
- 操作対象のジョブに資源グループおよび所有者が設定されていて，操作する JP1 ユーザーがその資源グループに対して JP1_AJS_Admin 権限を持っている。
- 操作対象のジョブに資源グループが指定されていない。

管理者権限が与えられていない場合，JP1 権限レベルが JP1_AJS_Manager または JP1_AJS_Editor の JP1 ユーザーがジョブの詳細定義を変更できます。ただし，JP1 ユーザーがジョブの所有者権限を持っているかどうか，およびジョブの実行ユーザー種別の設定によって，変更できる詳細定義の項目が異なります。

JP1 ユーザーの所有者権限の有無，およびジョブの実行ユーザー種別の設定と，変更できる詳細定義の項目を，次の表に示します。

表 6-7　変更できる詳細定義の項目

所有者権限	実行ユーザー種別	
	登録ユーザー	所有ユーザー
あり	○	○※1
なし	△	×※2

（凡例）

○：すべての項目を変更できる。

△：所有者，JP1 資源グループ，および実行ユーザー種別以外の項目を変更できる。

×：すべての項目を変更できない。

注※1

所有者権限のあるユーザーが所有者を変更すると，実行ユーザー種別が登録ユーザーに変更されます。所有者を変更しても実行ユーザー種別が所有ユーザーのままだと，ユーザーが所有者を任意の所有者に変更することで，実行を許可されていないジョブでも任意の所有者の権限で実行できるようになってしまいます。そのため，所有者を変更すると実行ユーザー種別が登録ユーザーに変更されるようにすることで，許可されていないジョブの実行を防止しています。

注※2

所有者権限がないユーザーは，実行ユーザー種別が所有ユーザーのジョブの詳細定義を変更できません。これは，ユーザーがジョブの定義を自由に変更して，設定されている所有者の権限でジョブを実行してしまうことを防ぐためです。

なお，所有者が設定されていないジョブは，アクセス権限や実行ユーザー種別に関係なく，すべての項目を変更できます。

6.5　対応づけるユーザーの検討

マネージャーホストでのジョブの実行ユーザーと，ジョブの実行先ホスト（エージェントホスト）での OS ユーザーの対応づけを検討します。

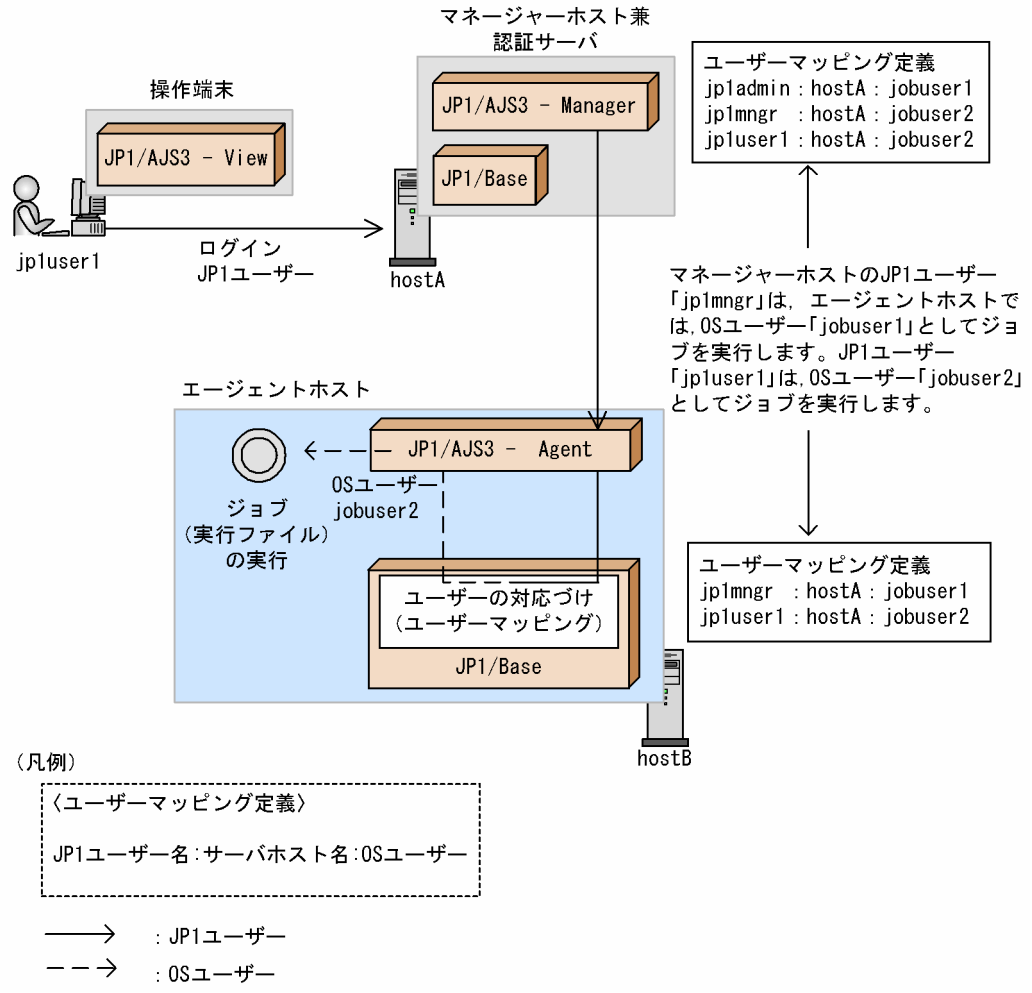
6.5.1　対応づけるユーザーの検討項目

JP1/AJS3 でジョブを実行する場合，ジョブはマネージャーホストからエージェントホストに転送されます。転送されたジョブは，ジョブの実行ユーザーに対応づけられている OS ユーザーの権限で実行されます。この対応づけのことをユーザーマッピングといい，JP1/Base のユーザーマッピング機能を使用します。

JP1/AJS3 - View からログインするときにも，ユーザーマッピングは必要です。JP1/AJS3 - View を使用する場合には，あらかじめユーザーマッピングを設定しておいてください。

ユーザーマッピングを使用したジョブ実行の概要を次の図に示します。

図 6-12　ユーザーマッピングを使用したジョブ実行の概要



上記の図では，エージェントホストで次のようなマッピングがされています。

- jp1mngr：「jobuser1」
- jp1user1：「jobuser2」

OS ユーザー「jobuser1」には，Administrators 権限またはスーパーユーザー権限を持つユーザーを設定してください。再起動など，プログラムの仕様上これらの権限が必要な場合に使用してください。

OS ユーザー「jobuser2」には，実行するジョブの権限（OS ユーザーアカウントやファイルのアクセス権限など）を設定して，ジョブが異常終了しないようにしてください。また，すべてのエージェントホストで OS ユーザー名（ジョブ実行ユーザー）を統一しておくと，管理しやすくなります。

また，ジョブやジョブネットを操作する際に使用される JP1 ユーザー名とユーザーマッピングの設定は，使用するコマンドによって決定方法が異なります。ユニット（ジョブやジョブネット）をajsxxxx コマンドおよび JP1/AJS3 - View から操作する場合，ジョブ実行環境上のジョブをjpqxxxx コマンドで操作，実行する場合，エージェント管理情報を操作する場合のそれぞれについて次に示します。次のルールを参照して，対応づけを検討してください。

なお，イベントジョブを操作するコマンドは JP1 権限レベルに依存しないため，JP1 ユーザー名は使用しません。

(1) ユニットを JP1/AJS3 - View およびコマンドで操作する場合の JP1 ユーザー名

ユニットを JP1/AJS3 - View から操作する場合，権限の確認に使用される JP1 ユーザー名は，JP1/AJS3 - View でログインした JP1 ユーザー名です。

ユニットをajsxxxx コマンドから操作する場合は，次の規則に従って JP1 ユーザー名が決定されます。

- 環境変数JP1_USERNAME を設定している場合

  環境変数JP1_USERNAME に設定しているユーザー名が JP1 ユーザー名として扱われます。このとき，JP1/Base のユーザーマッピング定義で，環境変数JP1_USERNAME に設定しているユーザー名とコマンド実行時の OS ユーザー名とがマッピングされている必要があります。コマンド実行時の OS ユーザーに Administrators 権限またはスーパーユーザー権限がある場合，ユーザーマッピングは不要です。

  また，環境変数JP1_HOSTNAME を設定しているかどうかで使用されるユーザーマッピング定義が異なります。

  環境変数 JP1_HOSTNAME を設定している場合

  環境変数JP1_HOSTNAME に設定している論理ホストに定義したユーザーマッピング定義の設定が使用されます。

  環境変数 JP1_HOSTNAME を設定していない場合

  物理ホストに定義したユーザーマッピング定義の設定が使用されます。

- 環境変数JP1_USERNAME を設定していない場合

  OS ユーザー名が JP1 ユーザー名として扱われます。

なお，ジョブ実行時にユーザーマッピング定義の内容がチェックされるため，OS ユーザー名と同じ名称の JP1 ユーザーの登録が必要です。

操作するジョブやジョブネットの属性に JP1 資源グループ名の指定がある場合，JP1/Base 認証サーバにアクセス権限を問い合わせます。環境変数JP1_HOSTNAME を設定しているときは，設定値の論理ホストに定義した JP1/Base 認証サーバが使用され，環境変数JP1_HOSTNAME を設定していないときは，物理ホストに定義した JP1/Base 認証サーバが使用されます。ただし，コマンド実行時の OS ユーザーに Administrators 権限またはスーパーユーザー権限がある場合には，JP1/Base 認証サーバにアクセス権限を問い合わせません。

次に，ユニットを操作するコマンドをリモート実行する場合について説明します。リモート実行ができるコマンドについては，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス 1.1 コマンドの記述形式」を参照してください。コマンド実行先ホストで必要な設定を次に示します。

- 環境変数JP1_USERNAME を設定している場合

  リモート実行時に，環境変数JP1_USERNAME に設定しているユーザー名が JP1 ユーザー名として扱われ，コマンド実行先ホスト上で実行されます。コマンド実行先ホスト上のユーザーマッピング定義の設定で，環境変数JP1_USERNAME に設定しているユーザー名とコマンド実行先ホスト上の任意の OS ユーザー名とがマッピングされている必要があります。

  また，コマンド実行先ホストに，論理ホスト名または物理ホスト名のどちらを指定しているかで，使用されるユーザーマッピング定義が異なります。

  コマンド実行先ホストに論理ホスト名を指定した場合

  指定した論理ホストに定義したユーザーマッピング定義の設定が使用されます。

  コマンド実行先ホストに物理ホスト名を指定した場合

  指定した物理ホストに定義したユーザーマッピング定義の設定が使用されます。

- 環境変数JP1_USERNAME を設定していない場合

  コマンド実行元ホストの OS ユーザー名が JP1 ユーザー名として扱われます。

操作するジョブやジョブネットの属性に JP1 資源グループ名の指定がある場合は，JP1/Base 認証サーバにアクセス権限を問い合わせます。コマンド実行先ホストに論理ホスト名を指定した場合，論理ホストに定義した JP1/Base 認証サーバが使用されます。コマンド実行先ホストに物理ホスト名を指定した場合，物理ホストに定義した JP1/Base 認証サーバが使用されます。コマンドを使用する際に必要な JP1 権限レベルを設定してください。ただし，マッピングされているプライマリーユーザーに Administrators 権限またはスーパーユーザー権限がある場合は，JP1/Base 認証サーバにアクセス権限を問い合わせません。

(2) ジョブ実行環境上のジョブをコマンドで実行・操作する場合の JP1 ユーザー名

jpqXXXX コマンドを使用してジョブ実行環境上のジョブを単体で操作したり，ジョブ実行環境を操作したりする場合は，コマンドを実行している OS ユーザーと同名の JP1 ユーザー名で権限を確認します。そのため，環境変数JP1_USERNAME の設定に関係なく，コマンドを実行する OS ユーザーを JP1 ユーザーとして登録してください。

JP1 ユーザーの登録および JP1 権限レベルの設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　3.1.1 JP1/Base のセットアップ」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　13.1.1 JP1/Base のセットアップ」（UNIX の場合）を参照してください。

また，それぞれのコマンドを使用する際に必要な JP1 権限レベルについては，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　1.5 コマンド一覧」を参照してください。

(3)　エージェント管理情報をコマンドで操作する場合の JP1 ユーザー名

エージェント管理情報をコマンドで操作する場合，権限の確認に使用される JP1 ユーザー名は次のルールによって決定されます。

- 環境変数JP1_USERNAME を設定している場合

  環境変数JP1_USERNAME に設定しているユーザー名が JP1 ユーザー名として扱われます。このとき，JP1/Base のユーザーマッピング定義で，環境変数JP1_USERNAME に設定しているユーザー名とコマンド実行時の OS ユーザー名とがマッピングされている必要があります。コマンド実行時の OS ユーザーに Administrators 権限またはスーパーユーザー権限がある場合，ユーザーマッピングは不要です。

  また，環境変数JP1_HOSTNAME を設定しているかどうかで使用されるユーザーマッピング定義が異なります。

  環境変数 JP1_HOSTNAME を設定している場合

  環境変数JP1_HOSTNAME に設定している論理ホストに定義したユーザーマッピング定義の設定が使用されます。

  環境変数 JP1_HOSTNAME を設定していない場合

  物理ホストに定義したユーザーマッピング定義の設定が使用されます。

- 環境変数JP1_USERNAME を設定していない場合

  OS ユーザー名が JP1 ユーザー名として扱われるため，OS ユーザー名と同じ名称の JP1 ユーザーを登録する必要があります。

エージェント管理情報を操作する場合，JP1/Base 認証サーバにアクセス権限を問い合わせます。このとき，操作するエージェント管理情報の対象が論理ホストであれば論理ホストに定義した JP1/Base 認証サーバが使用され，対象が物理ホストであれば物理ホストに定義した JP1/Base 認証サーバが使用されます。ただし，ajsagtshow コマンドおよびajsagtprint コマンドについては，Administrators 権限またはスーパーユーザー権限があればユーザーマッピングの必要はなく，アクセス権の問い合わせもされません。

(4)　フレキシブルジョブ実行時の JP1 ユーザー名

フレキシブルジョブでは，中継エージェントおよび宛先エージェントでユーザーマッピングを実施します。中継エージェントでは，通常のジョブ実行時と同様にユーザーマッピングを実施し，マッピングされた OS ユーザー中継処理を実施します。宛先エージェントでも，ユーザーマッピングを実施して，マッピングされた OS ユーザーでジョブを実行します。なお，中継エージェントを使用する場合，宛先エージェントの

ユーザーマッピング定義の［サーバホスト名］に中継エージェントのホスト名または「*（アスタリスク）」を設定する必要があります。

7 業務設計時の注意事項

この章では，業務設計での注意事項をまとめて説明します。

7.1　ルートジョブネットの実行登録数に関する注意事項

ルートジョブネットの実行登録数の最大数は，2,147,483,647 個ですが，この値は，あくまでディスクやメモリーなどの資源や実行性能に依存しない場合の論理値です。JP1/AJS3 で 4,000 個のルートジョブネットを実行登録しても，機能上の問題はありません。ただし，JP1/AJS3 でジョブネットおよびジョブを実行する場合，性能面から次に示す点について考慮する必要があります。

- 1 階層に何千個ものルートジョブネットを定義する場合

  1 階層に 4,000 個のルートジョブネットを定義すると，性能面で問題が発生するおそれがあります。必要に応じて，次の対応を取ってください。この対応は，実行性能だけでなく，GUI での監視やコマンドの性能にも効果があります。

  - 配下にあるユニットの数が少ないルートジョブネットが多数ある場合は，できるだけ一つのルートジョブネットにまとめるようにする。
  - ジョブグループを階層化する。

    一つのジョブグループの下にあるルートジョブネットの合計は，500 個以下を目安にしてください。
  - 登録するジョブネットについて次のことに配慮する。

    ・ルートジョブネットから数えた最大階層数は，2～3 階層程度にする。

    ・一つのルートジョブネット，またはネストジョブネットの直下に定義するネストジョブネットおよびジョブの数は，最大で 50～80 程度にする。

    ・一つのルートジョブネット配下のネストジョブネットおよびジョブの数は，最大で 500 程度にする。
  - スケジューラーサービスのルートジョブネットの実行登録数が，最大で 4,000 個程度になるように，ルートジョブネットを分割してスケジューラーサービスを多重起動する。

- その他

  - 保存世代数を使用した場合，「**保存世代数＊登録ルートジョブネット数**」で求められる値が，ディスク容量およびリソース管理上で登録されているルートジョブネット数として扱われます。そのため，ディスク容量などのリソース管理に留意する必要があります。
  - ルートジョブネットが多量に登録できることと，ジョブの同時実行数とは切り離して運用を設計する必要があります。システムリソースや処理性能に留意し，多量のジョブを同時に実行するなど，ジョブの実行量がある時期に集中する運用はしないように業務設計してください。
  - 業務の時間帯による分散，または複数 CPU での処理分散などを考慮して運用設計してください。
  - ルートジョブネットを多数定義した場合，ルートジョブネットの登録方法（一括，分割）と登録時間，起動開始に掛かる時間，コマンドによる一括定義・操作・表示についての性能を考慮して運用設計してください。
  - 大量のルートジョブネットを短時間に実行登録，リリース登録，またはリリース中止したり，確定実行登録の確定期間に長い期間を指定して実行登録したりした場合，システムへの負荷が一時的に集中して，起動条件成立によって作成される世代の実行開始が遅延することがあります。

大量のルートジョブネットを実行登録，リリース登録，またはリリース中止する場合は，回数を分けて実施してください。確定実行登録する場合は，確定期間を短くして，システムに負荷が掛からないようにしてください。

- 一つのルートジョブネットの中に，大量のジョブネットやジョブの定義をしたり，関連設定を多用したりすると，JP1/AJS3 - View および JP1/AJS3 - Web Console の操作性や性能が低下します。操作性や性能を考慮する場合，ジョブネットは 2～3 段の階層に分け，一つのルートジョブネット，またはネストジョブネットの直下に定義するユニット数は 50～80 程度にすることを推奨します。

7.2　保存世代数と性能との関係

保存世代数とは，ジョブネットの実行結果として保存される世代数のことです。保存世代数は，ルートジョブネットに設定できます。保存世代数を設定すると，設定した世代分（回数分）の実行結果を［デイリースケジュール］ウィンドウおよび［マンスリースケジュール］ウィンドウ，またはajsshowコマンドで確認できます。保存世代数は，1から99まで設定できますが，JP1/AJS3 - Managerのスケジューラーサービスの環境設定で，999世代まで保存世代数を持つようにすることができます。

しかし，保存世代数の増加は，そのジョブネットの規模と関連して「**保存世代数＊登録ユニット数**」で求められるレコード数が増加するため，データベースにアクセスする操作に多大な影響を及ぼします。したがって，保存世代数に設定する値は，システム性能への影響を十分配慮した上で決定してください。影響を受ける主な機能（処理）を次に示します。

- スケジューラーサービスの起動
- 実行登録解除
- JP1/AJS3 - Viewの［マンスリースケジュール］および［デイリースケジュール］ウィンドウの表示
- JP1/AJS3 - Web Consoleの［ダッシュボード］画面および［マンスリースケジュール］画面の表示
- 起動条件成立時の世代作成処理
- ajsshowなどのコマンド実行
- ジョブ実行

これらの処理が実行される上で，レコード数の増加とその処理時間は比例傾向にあります。システム性能によって異なりますが，負荷のない状態で，例えば100万レコードを実行登録解除する場合，その処理時間は1～2分程度，［マンスリースケジュール］ウィンドウを表示する場合には，8～10分程度掛かってしまいます。システム負荷が高い（CPU使用率が高い，ディスクアクセス頻度が高いなど）場合には，さらに時間が掛かります。これに対し，10万レコードを実行登録解除する場合，その処理時間は数秒程度，［マンスリースケジュール］ウィンドウを表示する場合には，1～2分程度になります。

この場合，ジョブネットの規模を2～3階層にし，50～80程度のユニット構成（推奨構成）にすると，最大保存世代数を999に設定したとしても運用に耐えられる計算になります。

このことから，保存世代数を大きくして運用したい場合には，ジョブネットの規模が小さくなるように設計し，規模の大きなジョブネットで運用したい場合には，その保存世代数を小さくするというように，保存世代数とユニット数のバランスを考えて設定することを検討してください。

さらに，登録解除処理中に別のジョブが実行されているのと同時に，［マンスリースケジュール］ウィンドウで状態監視している場合など，処理が並行して行われる運用を考慮し，保存世代数，ジョブネット規模は少なめに見積もるように設計してください。

上記の実行登録解除の処理時間については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.1.7 登録解除や保存世代数管理による世代削除処理方式の変更」（Windowsの場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.1.7 登録解除や保存世代数

管理による世代削除処理方式の変更」（UNIX の場合）で説明している，登録解除処理の非同期型の設定をした場合の時間です。同期型の設定のままで登録解除した場合には，何十分という処理時間が掛かってしまいます。

新規インストールした場合は，世代削除の方式がデフォルトで「非同期型（yes）」に設定されますが，バージョンアップインストールの場合，それまでの運用によっては「同期型（no）」が設定されていることがあります。環境設定パラメーターBACKGROUNDLEAVE の設定値を確認し，同期型が設定されている，または BACKGROUNDLEAVE が設定されていない場合は，非同期型に変更することを推奨します。

また，バージョン 8 以前からバージョンアップしている場合，デフォルトでは，起動条件を有効にしたジョブネットは，監視世代（「監視中」状態になる世代）と実行世代（起動条件の成立によって生成される世代）で別々に保存世代が管理されます。そのため，保存世代数に大きい値を設定した場合の影響が顕著に現れます。そのレコード数は，「**登録ユニット数＊保存世代数＊1 スケジュール当たりの起動条件成立数**」となります。起動条件付きジョブネットの場合は，10 より小さい値を設定し，ajsshow コマンドで随時実行結果を保存することを推奨します。ajsshow コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajsshow」を参照してください。バージョン 8 以前からのバージョンアップの場合に起動条件が設定されているジョブネットの保存世代数を増やす場合の注意事項については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4.2.3(4) バージョン 8 以前の JP1/AJS2 からバージョンアップインストールした場合の注意事項」を参照してください。

7.3 PC ジョブ使用時の注意事項

PC ジョブ使用時の注意事項を次に示します。

なお，PC ジョブが起動失敗したり，異常終了したりする場合の要因や注意事項については，マニュアル「JP1/Automatic Job Management System 3 トラブルシューティング　2.6.2 標準ジョブ，HTTP 接続ジョブ，アクションジョブ，およびカスタムジョブのトラブルへの対処」もあわせて参照してください。

- **システムのリソース不足の発生を少なくするための対応（Windows の場合）**

  エージェントホストが Windows の場合，ある数以上のジョブを同時に実行すると，システムの環境によってはシステムのリソース（デスクトップヒープ領域）不足が発生し，エラーになることがあります。こうした場合の対応として次のことを検討してください。

  - **ほかのアプリケーションとデスクトップヒープ領域を共用しない方法**

    JP1/AJS3 サービスのアカウントをユーザーアカウントにしてください。JP1/AJS3 サービスのアカウントを，ほかのサービスのアカウントやログオンユーザーと異なるユーザーアカウントにすれば，デスクトップヒープ領域を共用しないで運用できます。

  - **デスクトップヒープ領域の消費を少なくする方法**

    ジョブを実行するときによく使用する OS ユーザーを，JP1/AJS3 のサービスのアカウントと同じにすることで，デスクトップヒープ領域の消費を少なくできます。

- **アプリケーションファイル名に空白文字が含まれている場合の対応（Windows の場合）**

  エージェントホストが Windows の場合，ファイルタイプ（拡張子）に関連づけられているアプリケーションファイル名に空白文字が含まれるときは，エクスプローラのファイルタイプの編集で確認し，アプリケーションファイル名を「”」（ダブルクォーテーションマーク）で囲んでください。

- **ジョブが「起動失敗」状態にならないようにするための注意（Windows の場合）**

  エージェントホストが Windows の場合，JP1/AJS3 サービスの起動ユーザーに次に示す権限がないと，ジョブが「起動失敗」状態になることがあります。そのため，JP1/AJS3 サービスの起動ユーザーに，次に示す権限を設定してください。キューレスジョブを使用する場合は，キューレスエージェントサービスのアカウントに次の権限を設定してください。

  - ジョブの実行ファイル名の読み取り権限および実行権限
  - ジョブの環境変数ファイル名の読み取り権限
  - ジョブの標準入力ファイル名の読み取り権限
  - ジョブの標準出力ファイル名の読み取り権限および書き込み権限
  - ジョブの標準エラー出力ファイル名の読み取り権限および書き込み権限
  - ジョブの転送元ファイル名の読み取り権限

- **PC ジョブに OS のシャットダウンコマンドを登録しないこと**

  PC ジョブに自ホストやエージェントホストに対する OS のシャットダウンコマンドを登録して実行しないでください。Windows のシャットダウンは JP1/AJS3 の停止完了を待たないため，JP1/AJS3 が稼働中にシャットダウンを実行すると，JP1/AJS3 のデータファイルが壊れるなど問題が発生するおそ

れがあります。ジョブの自動運用でシャットダウンしたい場合は，JP1/Power Monitor をインストールし，アクションジョブの「ローカル電源制御ジョブ」を実行する運用を検討してください。

また，手動でシステムを終了する場合，JP1/Power Monitor の電源制御コマンド（aompwcon コマンド）を実行してください。JP1/Power Monitor をインストールしないで手動でシャットダウンを実行する場合は，JP1/AJS3 のサービスを手動で停止したあとにシャットダウンを実行してください。

- **バックアップ強化機能を有効にして PC ジョブから jajs_dbbackup コマンドを実行する場合**

  PC ジョブが定義されている組み込み DB を指定して jajs_dbbackup コマンドを実行した場合，実際はジョブが実行されている状態でも PC ジョブの状態が次の状態のまま遷移しないことがあります。ただし，jajs_dbbackup コマンドが終了したあと，PC ジョブの状態は正常に遷移します。

  - 実行待ち
  - キューイング

  この状態を回避するには，jajs_dbbackup コマンドを実行する PC ジョブは，バックアップ対象外の組み込み DB で動作するスケジューラーサービス上に定義して実行してください。

  バックアップ強化機能の概要については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 5.2.5 バックアップ強化機能による組み込み DB のバックアップとリカバリー」を参照してください。

- **ジョブで指定するファイル名を 254 バイト以内にする（Windows の場合）**

  次に示すファイル名に文字数が 255 バイト以上のファイル名を指定した場合，ジョブが「起動失敗」状態または「異常検出終了」状態になることがあります。

  - ジョブの実行ファイル名
  - ジョブの環境変数ファイル名
  - ジョブの標準入力ファイル名
  - ジョブの標準出力ファイル名
  - ジョブの標準エラー出力ファイル名
  - ジョブの終了判定のファイル名
  - ジョブの転送元ファイル名
  - ジョブの転送先ファイル名

  この現象は，キューレスジョブの場合にも起こります。上記のファイル名には，254 バイト以内のファイル名を指定してください。

- **実行ファイル名に標準入力ファイルからのデータの入力に対応していないコマンドまたはプログラムを指定する場合**

  詳細定義ダイアログボックスの［実行ファイル名］に，標準入力ファイルからのデータの入力に対応していないコマンドまたはプログラムを指定してジョブを実行する場合，［標準入力ファイル名］，［標準出力ファイル名］，および［標準エラー出力ファイル名］のすべてに「CON」を指定してください。この場合，JP1/AJS3 では，標準入力からのデータの読み込みおよび標準出力，標準エラー出力に出力されたデータの取得を実施しません。このため，JP1/AJS3 - View の［実行結果詳細］ダイアログボック

スにジョブの実行結果が表示されません。標準入力ファイルからのデータの入力に対応していないコマンドの例としては，timeout コマンドがあります。

- **任意のプログラムのショートカットファイルを Windows のスタートアップに格納する**

  任意のプログラムのショートカットファイルを Windows のスタートアップに格納するには，次のフォルダに格納してください。

  %ALLUSERSPROFILE%¥Microsoft¥Windows¥Start Menu¥Programs¥StartUp

  「%ALLUSERSPROFILE%」のデフォルトは「**システムドライブ**¥ProgramData」です。

7.4 UNIX ジョブ使用時の注意事項

UNIX ジョブ使用時の注意事項を次に示します。

なお，UNIX ジョブが起動失敗したり，異常終了したりする場合の要因や注意事項については，マニュアル「JP1/Automatic Job Management System 3 トラブルシューティング　2.6.2 標準ジョブ，HTTP 接続ジョブ，アクションジョブ，およびカスタムジョブのトラブルへの対処」もあわせて参照してください。

- **UNIX で実行したジョブで設定できる終了コードについて**

  UNIX で実行したジョブで設定できる終了コードの値は 0～255 です。ジョブの起動失敗や，標準出力データまたは標準エラー出力データの取得に失敗した場合は，終了コードは-1 となります。

- **ジョブが「起動失敗」状態にならないようにするための注意（UNIX の場合）**

  エージェントホストが UNIX の場合，JP1/AJS3 サービスの起動ユーザーおよびジョブの実行 OS ユーザーに，次に示すファイルおよび格納先ディレクトリに対する書き込み権限および読み込み権限がないと，ジョブが「起動失敗」状態になることがあります。そのため，JP1/AJS3 サービスの起動ユーザーに，次に示すファイルおよび格納先ディレクトリに対する書き込み権限および読み込み権限を設定してください。

  - ジョブの標準出力ファイル
  - ジョブの標準エラー出力ファイル
  - エージェントプロセス実行時のワークディレクトリ
  - ジョブの詳細定義で指定するワークパス※1
  - 実行 OS ユーザーのホームディレクトリ※1
  - ジョブ実行制御のログファイル※2

  注※1

  ジョブの詳細定義で指定するワークパスは，ジョブを実行するときのカレントディレクトリです。省略した場合，実行 OS ユーザーのホームディレクトリが仮定されます。ホームディレクトリが未定義の場合は「/（ルート）」が仮定されます。

  注※2

  ジョブ実行制御で使用するログファイルの詳細については，マニュアル「JP1/Automatic Job Management System 3 トラブルシューティング　1.2.5 ログファイルおよびディレクトリ一覧」を参照してください。

- **UNIX ジョブ実行時のリソース制限値についての注意事項**

  UNIX 版 JP1/AJS3 では，JP1/AJS3 起動時のリソース制限値がジョブ実行時にも有効になります。そのため，リソースの制限値を設定したい場合は，JP1/AJS3 の起動ユーザーである root ユーザーに対して設定してください。ただし，実行するジョブに制限値を指定した場合は，指定した値が有効になります。詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5 ジョブ実行環境設定」を参照してください。

  ファイルサイズの制限値を変更する例を次に示します。

1. root ユーザーのログインプロファイル（通常は[/.profile]（$HOME/.profile））に次に示す記述を設定する。

   fsize には必要なファイルサイズを設定します。無制限にする場合は，unlimited を設定してください。

   ulimit -f fsize

2. root ユーザーでログインする。

3. JP1/AJS3 サービスを root ユーザーで起動する。

   fsize の値が有効になります。

注意事項

AIX および Linux の場合，OS のリソース設定ファイル（AIX の場合は/etc/security/limits，Linux の場合は/etc/security/limits.conf）の値は，telnet 接続などで起動されるlogin コマンド経由のプロセスでだけ有効になります。JP1/AJS3 から起動されるジョブは，サービスから起動されるプロセスであるため OS のリソース設定ファイルの設定は有効になりません。

- JP1/AJS3 サービスを自動起動している場合の注意

JP1/AJS3 サービスを自動起動している場合，root ユーザーのログインプロファイルは読み込みません。そのため，ログインプロファイルで root ユーザーのリソース制限値を変更する設定にしても，手動でログインして JP1/AJS3 サービスを起動した際に設定されるリソース制限値と値が異なることがあります。この場合，ジョブ実行環境の環境設定パラメーターでリソース制限値を設定してください。ジョブ実行環境の環境設定パラメーターについては，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5 ジョブ実行環境設定」を参照してください。

また，リソース制限値は JP1/AJS3 の自動起動スクリプト（/etc/opt/jp1ajs2/jajs_start）にも記載できます。その場合は，十分な検証を実施した上で運用してください。

なお，ジョブ実行時のグループ ID は，root ユーザーでログインしたときに設定されるグループ ID と異なります。詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　5.4.12 ジョブ実行時のグループ ID（UNIX 限定）」を参照してください。

- ジョブプロセスの umask 値をジョブ実行ユーザーのプロファイルや実行シェル内で明示的に設定する

JP1/AJS3 サービスから起動するジョブプロセスの umask 値は，明示的な指定がなければ，JP1/AJS3 サービスを起動するシェルの umask 値になります。

/etc/profile や$HOME/.profile など，ジョブ実行ユーザーのプロファイルで指定している場合は，JP1/AJS3 サービス起動シェルの umask 値にその値を上書きした値が設定されます。

ジョブプロセスの umask 値は，ジョブ実行ユーザーのプロファイルや，実行シェル内で明示的に設定してください。

また，ジョブの定義で指定する「標準出力ファイル」および「標準エラー出力ファイル」の umask 値を変更する場合は，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　5.4.7 標準出力ファイル名・標準エラー出力ファイルの umask 値（UNIX 限定）」を参照してください。

- 環境変数 PATH に「/bin：/usr/bin」以外の値を設定したい場合

JP1/AJS3からジョブを起動した場合，JP1/AJS3が明示的に環境変数PATHに「/bin：/usr/bin」を設定しています。それ以外の値を設定したい場合はジョブ定義時に指定したコマンドやスクリプトファイルでの定義，またはローカルログインスクリプトでの定義で設定するようにしてください。

- **端末を必要とするユーザープログラムをジョブとして実行する場合の注意**

  UNIX版JP1/AJS3で端末を必要とするユーザープログラムをジョブとして実行すると，正しく動作しない（ジョブが異常終了する）場合があります。

- **キューレスジョブ使用時にログインシェルなどのOSユーザー情報を変更する場合の注意**

  キューレスジョブ使用時にログインシェルなどのOSユーザー情報を変更する場合は，変更後に次のどちらかの手順でキャッシングを消去してください。

  - ajsqlalterコマンドを，-rオプションを指定して実行する。
  - キューレスエージェントサービスを再起動する。

- **OSユーザーを登録または更新する場合の注意**

  ジョブ実行中に，システム管理者権限でpasswdコマンドなどによってOSユーザーを登録または更新しないでください。OSユーザーの登録または更新が終了したあとにジョブを実行してください。

- **バックアップ強化機能を有効にしてUNIXジョブからjajs_dbbackupコマンドを実行する場合**

  UNIXジョブが定義されている組み込みDBを指定してjajs_dbbackupコマンドを実行した場合，実際はジョブが実行されている状態でもUNIXジョブの状態が次の状態のまま遷移しないことがあります。ただし，jajs_dbbackupコマンドが終了したあと，UNIXジョブの状態は正常に遷移します。

  - 実行待ち
  - キューイング

  この状態を回避するには，jajs_dbbackupコマンドを実行するUNIXジョブは，バックアップ対象外の組み込みDBで動作するスケジューラーサービス上に定義して実行してください。

  バックアップ強化機能の概要については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 5.2.5 バックアップ強化機能による組み込みDBのバックアップとリカバリー」を参照してください。

7.5　リカバリーユニット使用時の注意事項

リカバリーユニット定義時の注意事項を次に示します。

- リカバリーユニットが正常終了しても，そのリカバリーユニットが定義されているジョブネットは異常終了として扱われます。
- リカバリーユニットの後続ユニットには，リカバリーユニットしか定義できません。リカバリーユニットの後続に通常のユニットを定義した場合，そのユニットは実行されません。
- リカバリーユニットの後続ユニットとして定義したユニットは，ジョブネットが正常に実行されているときには実行されません。異常終了によって先行のリカバリーユニットが実行され，そのリカバリーユニットが正常終了した場合だけ実行されます。
- リカバリーユニットの警告または異常状態は，上位ジョブネットの状態に影響を与えません。ただし，リカバリージョブネットが開始遅延または終了遅延した場合は，上位ジョブネットも遅延状態となります。
- 先行ユニットの状態とリカバリーユニットの実行の可否について，次の表に示します。

表 7-1　先行ユニットの状態によるリカバリーユニットの実行可否

項番	先行ユニットの状態	リカバリーユニットの実行可否
1	未計画	×
2	開始時刻待ち	×
3	先行終了待ち	×
4	保留中	×
5	実行待ち	×
6	キューイング	×
7	未実行終了	×
8	未実行終了-W	×
9	計画未実行	×
10	実行中	×
11	異常検出実行中	×
12	警告検出実行中	×
13	正常終了	×
14	正常終了-偽	×
15	警告検出終了	×
16	異常検出終了	○※
17	異常検出終了-WR	○※

項番	先行ユニットの状態	リカバリーユニットの実行可否
18	繰り越し未実行	×
19	順序不正	○※
20	中断	○※
21	強制終了	○※
22	強制終了-WR	○※
23	起動失敗	○※
24	起動失敗-WR	○※
25	終了状態不明	×
26	終了状態不明-WR	×
27	閉塞	×
28	起動条件待ち	－
29	監視中	－
30	監視未起動終了	－
31	監視打ち切り終了	－
32	監視中断	－
33	監視正常終了	－

（凡例）

○：リカバリーユニットを実行できる

×：リカバリーユニットを実行できない

－：先行ユニットの状態として存在しない

注※

ウォームスタートまたはディザスターリカバリースタートで起動した場合やルートジョブネットを中断または強制終了した場合には，異常状態になってもリカバリーユニットは実行されません。

7.6 イベントジョブ使用時の注意事項

イベントジョブ使用時の注意事項を次に示します。

- イベントジョブは，実行エージェントグループでの運用に対応していません。ルートジョブネットやネストジョブネットに実行エージェントグループを指定した場合，ジョブネット配下の実行エージェントの指定がないイベントジョブは，ジョブネットに指定した実行エージェントグループを実行エージェントとしてジョブを実行しようとします。このとき，実行エージェントグループと同名の実行エージェントがあれば，該当する実行エージェントに対してイベントジョブを実行します。実行エージェントグループと同名の実行エージェントがなければ，ジョブの実行時に統合トレースログにメッセージ「KAVT0403-E 指定されたエージェントはジョブ実行環境に定義されていません（host=**実行エージェント名**，**保守情報**）」を出力してエラーになります。したがって，ルートジョブネットやネストジョブネットに実行エージェントグループを指定する場合には，ジョブネット配下のイベントジョブに明示的に実行エージェントを指定しておいてください。
- イベントジョブの詳細定義の［実行エージェント］には，実行エージェントグループ名を指定できません。
- JP1/AJS3 が発行した JP1 イベント，Windows イベントログ，syslog，HNTRLib2，およびスケジューラーログなどのログファイルの内容については，事象を発生させた JP1/AJS3 自身では正しく検知できないことがあります。JP1/AJS3 が発行した情報を監視する場合の注意事項や，イベントジョブの監視対象の詳細については，「7.6.9 JP1/AJS3 が発行するイベントやメッセージを監視する」を参照してください。
- 起動条件中のイベントジョブについては，起動条件監視中に JP1/AJS3 - Manager が停止した場合，再起動後に再びイベント監視を始められます。また，起動条件で複数のイベントを監視していた場合，条件が成立していたイベントの受信情報を，再起動後も保持しておけます。
- 起動条件を定義してジョブネットを起動するのと同じように，イベントジョブをジョブネットの先頭に定義すると，条件が成立するのを待ってジョブネットを実行できます。起動条件を定義した場合，イベント受信の監視中は，ジョブネットが起動条件待ち状態になります。イベントジョブを定義した場合，イベント受信の監視中は，ジョブネットが実行中状態になります。イベントジョブを先頭に定義する場合は，計画的に発生することがわかっている事象を待つ場合に使うことを推奨します。
- 複数のイベントを監視する場合，正規表現などで一つのイベントジョブとしてまとめて監視すると，性能が向上します。

  例えば，JP1 イベント受信監視ジョブで，イベント ID が 00004131 でメッセージが KAVS0272-E と KAVS0273-E の JP1 イベントを監視する場合，イベント ID だけを指定する，またはイベント ID に「00004131」を指定し，メッセージに「KAVS」を指定するなどの方法で，イベントジョブを一つにまとめてください。なお，引き継ぎ情報を利用すると，イベントの内容を識別できます。
- JP1 イベント受信監視ジョブ，ログファイル監視ジョブ，および Windows イベントログ監視ジョブの次の項目は，正規表現で定義できます。
  - イベント発行元ユーザー名（JP1 イベント受信監視ジョブ）
  - イベント発行元グループ名（JP1 イベント受信監視ジョブ）

- イベント発行元ホスト名（JP1 イベント受信監視ジョブ）
- メッセージ（JP1 イベント受信監視ジョブ）
- イベント詳細情報（JP1 イベント受信監視ジョブ）
- トラップするデータ（ログファイル監視ジョブ）
- ログ情報以外のデータ行の指定（ログファイル監視ジョブ）
- 説明（Windows イベントログ監視ジョブ）

正規表現で定義できる項目では，すべての文字に一致する正規表現「.*」を多数使用すると，検知したイベントと監視条件の突き合わせに時間が掛かるため，イベントの検知が遅延する場合があります。そのため，「.*」を使用する場合は必要な個所にだけ使用するようにしてください。

また，正規表現で定義できる項目では，大文字と小文字が区別されます。監視するイベントに合わせて指定してください。

- イベントジョブに打ち切り時間を指定した場合，打ち切り時間は実行先のエージェントホストでカウントします。このため，実行先のエージェントホストが電源ダウンなどで再起動するなどして，イベントジョブによる事象の監視が継続された場合，打ち切り時間は実行先のエージェントホストが再起動した時刻からカウントされます。

  なお，打ち切り時間のカウントの再開始，および再開始された時刻は，イベントジョブの実行結果詳細に出力されるメッセージ「KAVT0603-W 監視が一時中断された為，打ち切り時間は再開始時間からの経過となります」で確認できます。

  また，エージェントホストの状態に関係なく，絶対時刻で監視を打ち切る場合は，ジョブネットの起動条件にイベントジョブを定義し，起動条件の有効範囲を絶対時刻で指定してください。起動条件の詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　3.4 起動条件の定義」を参照してください。

- Windows の場合，JP1/AJS3 のイベントジョブは，実行時の JP1 ユーザーには依存しません。JP1/AJS3 サービスのアカウント権限に依存します。そのため，イベントジョブの一つであるファイル監視ジョブを使用する場合は，ファイル監視ジョブの監視対象ファイルおよびフォルダに対する書き込み権限を JP1/AJS3 サービスのアカウントに与える必要があります。

  権限が与えられていない場合，次に示す現象が起こります。

  - ファイル監視ジョブが異常終了する。
  - イベントが発生しない

- JP1/AJS3 のイベントジョブは，JP1/Base の環境設定に定義されている JP1 ユーザー，およびそれぞれのジョブに定義されている JP1 ユーザーの権限には依存しません。

- イベントジョブを使用する場合，システム負荷や一時的なネットワーク障害などによって，イベントジョブを実行した時間と実際に実行エージェントでイベントの監視が実行中になるまでのタイムラグが発生することがあります。この場合，実際にイベントの監視が実行中になってから発生したイベントが検知の対象となります。このため，イベントジョブを実行する時間は監視対象とするイベントが発生する時間を考慮し，余裕を持った時間に実行する必要があります。

- イベントジョブ（起動条件に設定しているものを含む）を実行した場合，実行したイベントジョブの定義データや，監視条件が成立した際のイベントの情報などがイベント・アクション制御マネージャーやイベント・アクション制御エージェントなどのプロセス間で通信されます。その際，一時的なネットワーク障害や通信相手のプロセスがビジー状態などで通信できないと，いったん通信する情報をファイルに保存し，時間をおいて再試行します。JP1/AJS3 ではこのファイルを「未通知情報」と表現しています。再試行に成功すると，未通知情報を削除します。
- JP1/AJS3 を使用する場合，JP1/Base のイベントサービス環境の設定で，API 設定ファイルの`server`パラメーターの通信タイプに「keep-alive」を設定してください。「close」を設定すると，JP1/AJS3 が起動時に発行する JP1 イベントが発行できなくなったり，統合トレースログにメッセージ KAVT1040-E が出力されて，JP1 イベント受信監視ジョブ，ログファイル監視ジョブ，Windows イベントログ監視ジョブがイベントを検知できなくなったり，JP1 イベント送信ジョブが異常検出終了したりするなど，正常に動作しない問題が発生することがあります。設定方法および API 設定ファイルについては，マニュアル「JP1/Base 運用ガイド」を参照してください。
- マネージャー・エージェント構成でイベント・アクション制御マネージャーとイベント・アクション制御エージェントがネットワーク障害などで通信できない場合，

  次のような操作をすると，イベントジョブ（起動条件として定義されているイベントジョブも含む）はマネージャー上で終了状態になってもエージェント上では監視し続けるという矛盾が発生します。

  - イベントジョブを強制終了する。
  - 起動条件付きジョブネットを強制終了する。
  - イベントジョブの状態を状態変更して終了状態に遷移させる。

  矛盾が発生しているイベントジョブは再実行できなくなったり，その他の正常なイベントジョブの処理を遅延させたりします。

  そのため，ネットワーク障害などが発生している間にこのような操作をした場合は，マネージャーホストで`jpomanjobshow` コマンドを，エージェントホストで`jpoagtjobshow` コマンドを実行してください。その結果を比較して，マネージャー上で終了しているイベントジョブがエージェント上では監視中となっていないか確認してください。

  確認の結果，エージェント上のイベントジョブだけが監視中となっていた場合は，エージェントホストの JP1/AJS3 サービスを再起動し，エージェント上で監視し続けているイベントジョブを終了させてください。

7.6.1 JP1 イベント受信監視ジョブの注意事項

JP1 イベント受信監視ジョブの注意事項（使用する前に知っておいた方がよいこと）を次に示します。

JP1 イベントは JP1/Base で管理されるイベントで，JP1 シリーズプログラムで発生した事象を契機として発行されます。JP1 イベントは，エラー，警告，通知などの重大度やメッセージなどの情報を持っているので，重大度ごとに異なる後続ジョブを実行したり，特定のメッセージを受け取ったときだけ後続ジョブを実行したりできます。JP1 イベント中のメッセージや詳細情報の一部分を正規表現で切り出して，後続ジョブに引き継ぐこともできます。

JP1 イベント受信監視ジョブを使った例を次に示します。

- JP1 シリーズプログラムでエラーが発生するか，または警告が通知された場合に後続ジョブを実行する。
- 複数の JP1 シリーズプログラムで処理を実行している場合，すべての処理が終了してから後続ジョブを実行する。
- 他ホストの JP1/AJS3 - Manager で実行されたジョブネットの終了時に JP1 イベント送信ジョブを実行し，送信された JP1 イベントを別のジョブネットで受信し，後続ジョブを実行する。
- JP1/Base のイベント変換機能を使って，JP1 シリーズ以外のアプリケーションの終了を契機として後続ジョブを実行する。

  イベント変換機能については，マニュアル「JP1/Base 運用ガイド」を参照してください。

(1) JP1 イベントが発生するタイミングに関する注意事項

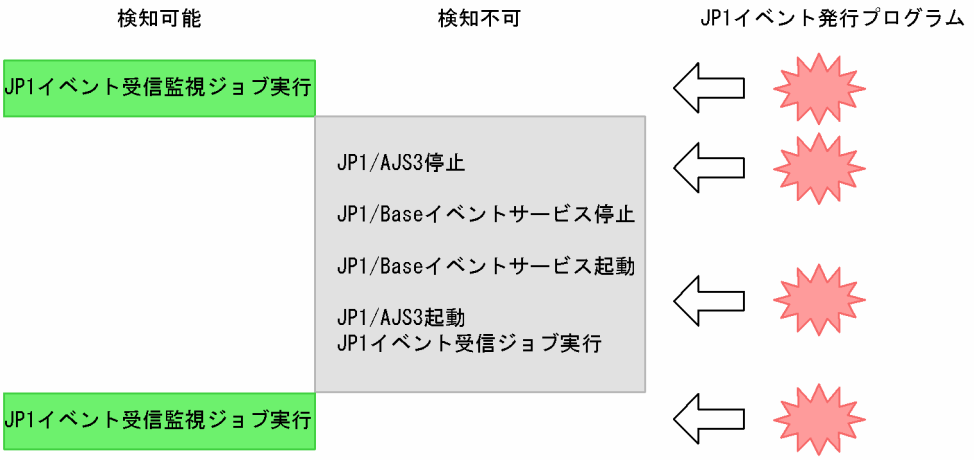
JP1 イベント受信監視ジョブで監視できる JP1 イベントは，JP1 イベント受信監視ジョブが実行状態になったあとに発生した JP1 イベントが対象です。そのため，次のタイミングで発生した JP1 イベントは，JP1/AJS3 では検知されません。

- JP1/AJS3 が停止中に発行された JP1 イベント
- JP1/AJS3 が起動したあとにイベントジョブが実行状態になる間に発生した JP1 イベント

JP1 イベント受信監視ジョブで監視する対象となる JP1 イベントは，JP1/AJS3 のイベントジョブが監視状態になるまで発行されない JP1 イベントを監視対象にしてください。

JP1 イベント受信監視ジョブが検知できないタイミングを次の図に示します。

図 7-1　JP1 イベント受信監視ジョブが検知できないタイミング



(2) JP1 イベント受信監視ジョブのオプション

JP1 イベント受信監視ジョブには，JP1 イベント受信監視ジョブが JP1 イベントの監視を開始する前に発生した JP1 イベントを受信監視の対象とするかどうかを指定する **JP1 イベント受信監視実行前のイベント検索オプション**を設定できます。このオプションは，イベント ID を指定し，かつ JP1 イベントの監視を

開始した時刻を基準に何分前まで検索するかを指定した場合に有効になります。指定できる値は 1～720（単位：分）の間です。なお，このオプションを指定した場合に基準となる時刻は，JP1 イベント受信監視ジョブが実行されるホストで設定されている時刻です。

このオプションを指定しない場合は，通常の JP1 イベント受信監視だけになり，JP1 イベントの監視を開始した時刻以前の JP1 イベントは検索しません。

このオプションを指定するには，次の注意事項があります。

- 「JP1 イベント受信監視実行前のイベント検索オプション」を指定する際に指定したイベントの検索範囲（JP1 イベントの監視を開始した時刻を基準に何分前まで検索するか）が大きくなればなるほど，JP1 イベント受信監視実行前のイベント検索に掛かる時間が長くなります。また，イベントの検索対象範囲内にある JP1 イベントの件数が多ければ多いほど，JP1 イベント受信監視実行前のイベント検索に掛かる時間が長くなります。そのため，イベントの検索範囲は必要最小限の短い時間となるようにしてください。長くても 10 分程度とすることを推奨します。
- JP1 イベント受信監視ジョブは，JP1/Base のイベント DB から JP1 イベントを取得しています。そのため，イベントの検索範囲内に JP1 イベントが発生していても，イベント DB が切り替わり，イベント DB から検索対象の JP1 イベントが削除された場合，イベントを検知しません。検索対象となる JP1 イベントが削除されるおそれがある場合は，「JP1 イベント受信監視実行前のイベント検索オプション」でイベントの検索範囲に必要最小限の短い時間を指定した上で，イベントの検索範囲内に監視対象イベントが発生する運用を行ってください。必要最小限の短い時間を指定しても，検索対象となる JP1 イベントが削除されることがあるときは，イベント DB を拡張してください。イベント DB については，マニュアル「JP1/Base 運用ガイド」を参照してください。
- 「JP1 イベント受信監視実行前のイベント検索オプション」を使用した JP1 イベント受信監視ジョブでは，検索範囲内に JP1 イベントが大量にある場合，次の現象が発生することがあります。
  - CPU 使用率が高くなる
  - JP1 イベント受信監視ジョブが終了するまでに時間が掛かる
  - JP1 イベント受信監視ジョブが異常検出終了する

  そのため，「JP1 イベント受信監視実行前のイベント検索オプション」は，数件程度の特定の JP1 イベントを監視条件としたい場合に使用してください。大量に発生したり，継続的に発生したりするような JP1 イベントを監視条件としたい場合には，「JP1 イベント受信監視実行前のイベント検索オプション」を使用しないでください。検索対象となる JP1 イベントが大量にある場合は，JP1 イベント受信監視ジョブの監視条件を細かく設定し，検索対象となる JP1 イベントを絞り込むか，「JP1 イベント受信監視実行前のイベント検索オプション」を指定する際に指定したイベントの検索範囲を短い時間にしてください。

  また，「JP1 イベント受信監視実行前のイベント検索オプション」を指定した JP1 イベント受信監視ジョブが同時に複数実行されると，検索範囲の条件が結合されて検索範囲が広がることがあります※。この場合，検索範囲が広くなるため，別々に JP1 イベント受信監視ジョブを実行した場合と比較して，検索範囲内にある JP1 イベントの量が増加する可能性があります。

そのため，「JP1 イベント受信監視実行前のイベント検索オプション」を指定している複数の JP1 イベント受信監視ジョブを同時に実行する場合は，監視条件に指定するイベント ID を 0～1FFF にするなど，検索範囲となるイベント ID の幅が広くならない運用を推奨します。

注※

例えば，次のように指定した二つの JP1 イベント受信監視ジョブが同時に実行されて検索範囲が結合された場合，「1～7FFFFFFF」のイベント ID が検索範囲となります。

・JP1 イベント受信監視ジョブ A の監視条件の指定

イベント ID：1

・JP1 イベント受信監視ジョブ B の監視条件の指定

イベント ID：7FFFFFFF

- 「JP1 イベント受信監視実行前のイベント検索オプション」は起動条件ジョブの中での使用を推奨します。起動条件ではないイベントジョブでも使用できますが，イベントジョブで「JP1 イベント受信監視実行前のイベント検索オプション」を使用する場合には，同じイベントを何度も受信監視の条件としてしまう場合があります。イベントジョブとして JP1 イベント受信監視ジョブを使用する場合には，その使用方法に注意が必要です。

  同一の JP1 イベントを受信監視の条件にしてしまう例を次に示します。

  （例）

  ジョブネット「recv」に，イベントジョブとして次の条件で JP1 イベント受信監視ジョブが登録されているとします。

  - 起動条件：

    ジョブネット「recv」は，実行間隔制御ジョブによって 9:00 と 9:10 に起動するように設定

  - JP1 イベント受信監視実行前のイベント検索オプションの指定：

    「30」（単位：分）を指定

  - イベント ID：

    「111」を指定

  - JP1 イベント（イベント ID：111）の発行状況：

    8:20 と 8:50

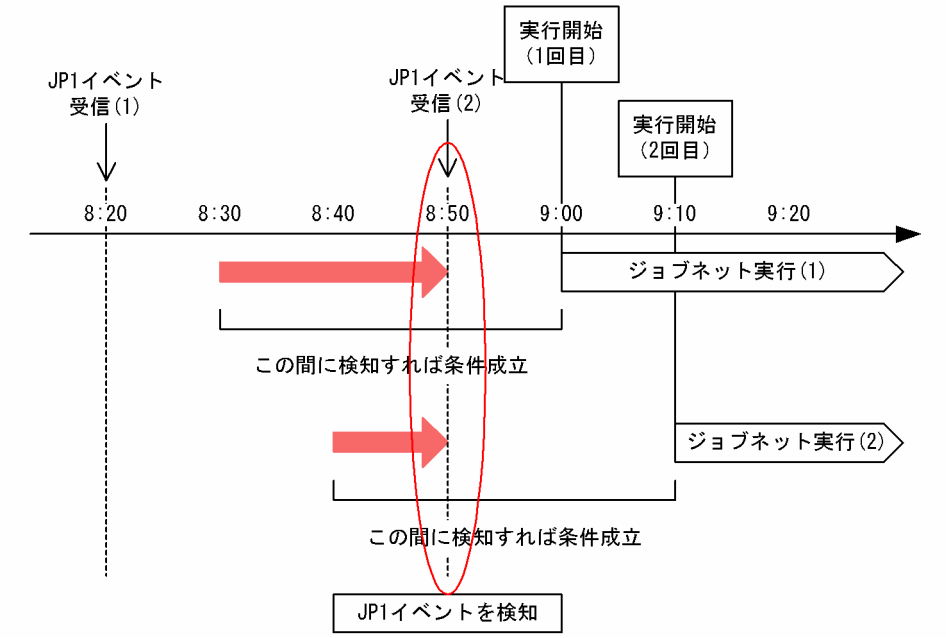
  動作

  1. ジョブネット「recv」が，実行間隔制御ジョブによって 9:00 に起動する。
  2. JP1 イベント受信監視実行前のイベント検索オプションで指定した条件に従って 8:30 の以降の JP1 イベントを検索します。
  3. ジョブネット「recv」が，実行間隔制御ジョブによって 9:10 に起動する。
  4. JP1 イベント受信監視実行前のイベント検索オプションで指定した条件に従って 8:40 の以降の JP1 イベントを検索します。

  結果

このとき，8:30 以降と 8:40 以降に検索条件を満たす JP1 イベントはともに 8:50 の 1 回であったとすると，9:00 に実行された JP1 イベント受信監視ジョブも 9:10 に実行された JP1 イベント受信監視ジョブも，両方とも同じ 8:50 の JP1 イベントを検知し，受信監視の条件としてしまいます。

図 7-2　同一の JP1 イベントを受信監視の条件にしてしまう例



(3)　JP1 イベント受信監視ジョブ定義時の注意事項

JP1 イベント受信監視ジョブを定義するときの注意事項を次に示します。

- JP1 イベント受信監視ジョブを実行する前に，JP1/Base イベントサービスを起動しておいてください。また，JP1/Base イベントサービスの API 設定は「keep-alive」に設定しておいてください。JP1/Base イベントサービスが起動されていない場合，JP1 イベント受信監視ジョブは，イベントサービスが起動されるまで実行待ちの状態になります。
- JP1 イベント受信監視ジョブが実行中になる前に発生した JP1 イベントは，受信できません。受信したい JP1 イベントは，JP1 イベント受信ジョブが実行中になってから発行するか，「JP1 イベント受信監視実行前のイベント検索オプション」を使用するようにしてください。
- JP1 イベントにはメッセージや詳細情報が含まれています。JP1 イベント受信監視ジョブで JP1 イベント中の文字列を監視したい場合は，正規表現を使って文字列を指定できます。正規表現を拡張した拡張正規表現を使用したい場合は，Windows は JP1/Base の設定を行ってください。UNIX では，JP1/Base の設定に関係なく，OS の拡張正規表現の文法で使用できます。

  Windows の拡張正規表現の設定方法については，マニュアル「JP1/Base 運用ガイド」の「使用する正規表現を拡張する」を参照してください。UNIX の正規表現については，UNIX のドキュメントを参照してください。
- 正規表現で，すべての文字に一致する「.*」を多数使用すると，JP1 イベントと突き合わせるために時間が掛かることがあります。長いメッセージなどに対しては，必要な個所に対してだけ「.*」を使用するようにしてください。「.*」を使用した JP1 イベント受信監視ジョブを複数実行すると，JP1 イベントを各ジョブの監視条件と突き合わせる時間が指数関数的に増えるため，イベントの検知が極端に遅延

するおそれがあります。そのため，いったん別のホストでJP1/Baseのイベント転送機能とフィルター条件でJP1イベントの数を絞り込んでから監視したり，監視条件を追加して「.*」部分の比較が行われる頻度を下げたりしてください。なお，UNIXでは，空白以外の文字に一致させたい場合など，「.*」の代わりに「[^]*（空白以外の文字の繰り返し）」を使用すると，JP1イベントと突き合わせる時間を短縮できます。

- JP1イベント受信監視ジョブを起動条件として定義する場合，必ず一つ以上の項目を定義してください。一つも定義がない場合，イベント受信監視を行っているホスト上でJP1イベントが発生するたびに，JP1イベント受信監視ジョブが終了します。JP1/AJS3が発行するJP1イベントでも，監視条件が成立してしまうため，ジョブを実行するたびにジョブネットが起動してしまうので注意してください。
- WindowsのJP1/Baseイベントサービスで，JP1/SES Version 5またはJP1/AJS Version 5以前のイベントサービス，JP1/SESのプロトコルを使用して発行されたJP1イベントを受信した場合，JP1イベントにはイベント発行元IPアドレスが設定されていません。そのため，WindowsではJP1イベント受信監視ジョブの監視条件にイベント発行元IPアドレスを指定しても，上記のイベントは監視条件に合致しません。
- JP1イベントのイベント詳細情報については，JP1イベント中の詳細情報がテキストの場合だけ監視対象になります。バイナリーデータが含まれている場合には，JP1イベント中の詳細情報が無視され，監視条件に合致しません。監視するJP1イベントにバイナリーデータが含まれている場合は，監視条件にイベント詳細情報を指定しないでください。
- JP1イベント中のイベント発行元ユーザー名には，JP1イベントを発行したOSユーザー名が設定されます。JP1イベント受信監視ジョブの監視条件のうち，イベント発行元ユーザー名にJP1ユーザー名を指定しても，JP1イベント受信監視ジョブはJP1イベントの受信を正しく監視できません。また，JP1イベント受信監視ジョブの監視条件のうち，イベント発行元ユーザー名はイベント受信監視時に大文字／小文字が区別されます。JP1イベント中の発行元ユーザー名に合わせて指定してください。
- JP1イベント受信監視ジョブの監視条件のうち，イベント発行元ホスト名はJP1イベント受信監視時に大文字小文字が区別されます。JP1イベント中のホスト名（発行元イベントサーバ名）に合わせて指定してください。
- 正規表現で指定する項目では，指定した文字列は部分一致で条件一致となります。完全一致させたい場合は，完全名を明示的に指定した正規表現を使用してください。
- JP1イベント受信監視ジョブで監視するJP1イベント情報については，JP1/Baseのイベントサービスの仕様に従います。監視したいJP1イベントの詳細については，マニュアル「JP1/Base 運用ガイド」，およびJP1イベントを発行する製品のマニュアルを参照してください。
- Windowsでは，JP1/Baseイベントサービスのイベントサーバ設定（conf）でJP1イベントを取得できるユーザー名（usersパラメーター）に，JP1/AJS3のサービスを起動するアカウントのユーザー名を設定してください。なお，デフォルトでは「users」はすべてのユーザーがJP1イベントを取得できるように設定されています。
- JP1イベントには，イベント発行元ホストのIPアドレスとしてIPv6形式のIPアドレスは出力されません。そのため，イベント発行元IPアドレスとしてIPv6アドレスは指定できません。

7.6.2　ファイル監視ジョブの注意事項

ファイル監視ジョブの注意事項（使用する前に知っておいた方がよいこと）を次に示します。

ファイル監視ジョブを使った例を次に示します。

- ファイルの書き込み時間を監視して，ファイルが変更されたときに後続ジョブを実行する。
- アプリケーションが出力するファイル，または他ホストから転送されてくるファイルを監視して，ファイルの作成が完了したあとに後続ジョブを実行する。

ファイル監視ジョブで監視する事象，ファイル名の指定方法，およびファイル監視ジョブのオプションを次に説明します。

(1)　ファイル監視ジョブで監視する事象

監視する事象を次に示します。

表 7-2　監視する事象

監視する事象	監視内容
作成※1，※2	指定した名称のファイルが作成されたかを監視します。
削除※3	指定した名称のファイルが削除されたかを監視します。
サイズ変更※2，※4	指定した名称のファイルサイズが変更されたかを監視します。
最終書き込み時刻変更※2，※4	指定した名称のファイルが更新されたかを監視します。更新した時刻を監視し，更新時刻が変わったら条件成立とします。

注※1

監視開始の時点で，指定した名称のファイルがすでにある場合には，そのファイルが削除され，再度作成されたときに条件が成立します。

なお，監視開始の時点で，指定した名称のファイルがすでにある場合に条件を成立させるかどうかを，監視開始オプションで指定することもできます。監視開始オプションの詳細については，「(3) ファイル監視ジョブのオプション」を参照してください。

注※2

監視条件に合致した場合，監視対象のファイルにファイル監視ジョブ以外のプロセスがアクセスしていないかチェックします（クローズチェック）。

監視条件に合致しても，ファイル監視ジョブ以外のプロセスが監視対象のファイルにアクセスしていた場合は条件成立となりません。この場合，次回の監視間隔に繰り越して監視し，そこでファイル監視ジョブ以外のプロセスのアクセスがなければ条件成立となります。このクローズチェックをすることで，ファイルコピー中など転送中（ファイルコピー中など）の監視対象のファイルを条件成立とすることを防ぐことができます。

注※3

監視開始の時点で，指定した名称のファイルがない場合には，そのファイルが作成されて，削除されたときに条件が成立します。

注※4

監視開始の時点で，指定した名称のファイルがない場合には，そのファイルが作成されて，サイズ変更または最終書き込み時刻が変更されたときに条件が成立します。ファイルが作成されただけでは条件は成立しません。

これらの条件は，複数指定することもできます。例えば，ファイルが削除または更新されたら後続ジョブを実行するように定義する場合は，「削除」と「最終書き込み時刻変更」を指定します。ただし，「サイズ変更」と「最終書き込み時刻変更」は同時に指定できません。

ファイル監視ジョブの基本動作例を紹介します。

(a) 監視条件で「作成」を指定した場合

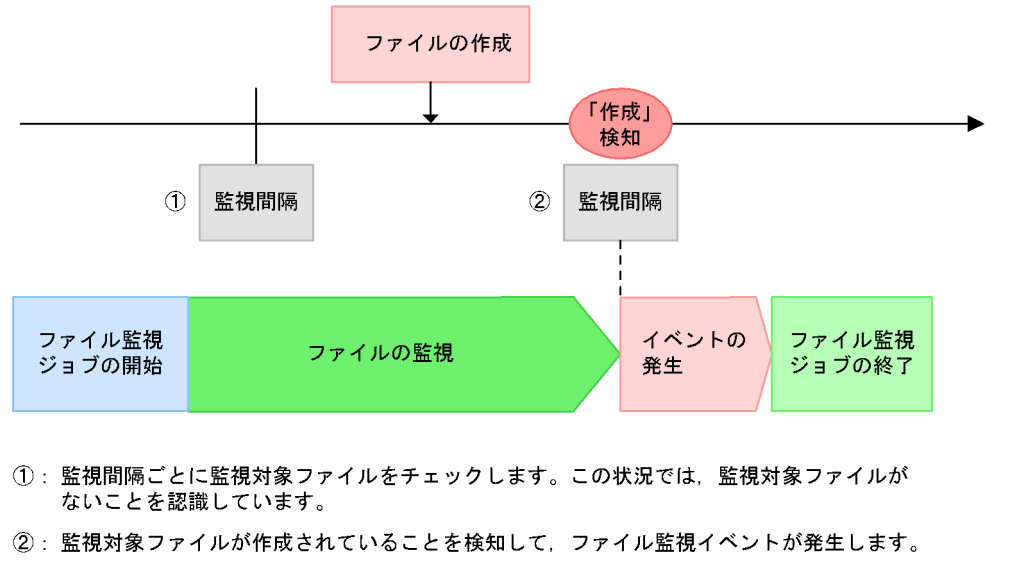
監視条件で「作成」を指定して監視している場合の，ファイル監視ジョブの基本動作を次の図に示します。

なお，図中の「ファイル」とは，監視対象ファイルを指します。

■ ジョブの開始時に監視対象ファイルがないとき

ジョブの開始時に監視対象ファイルがないときは，次のように動作します。

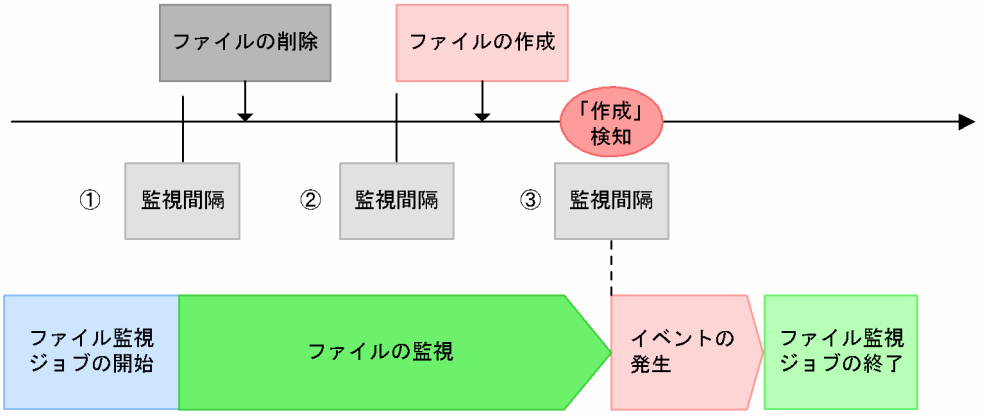
図 7-3　ジョブの開始時に監視対象ファイルがないとき



■ ジョブの開始時に監視対象ファイルがあるとき

ジョブの開始時に監視対象ファイルがあるときは，次のように動作します。

図 7-4　ジョブの開始時に監視対象ファイルがあるとき



①：監視間隔ごとに監視対象ファイルをチェックします。この時点では，監視対象ファイルがあることを認識しています。

②：監視対象ファイルが削除されたことを認識しています。

③：監視対象ファイルが作成されていることを検知して，ファイル監視イベントが発生します。

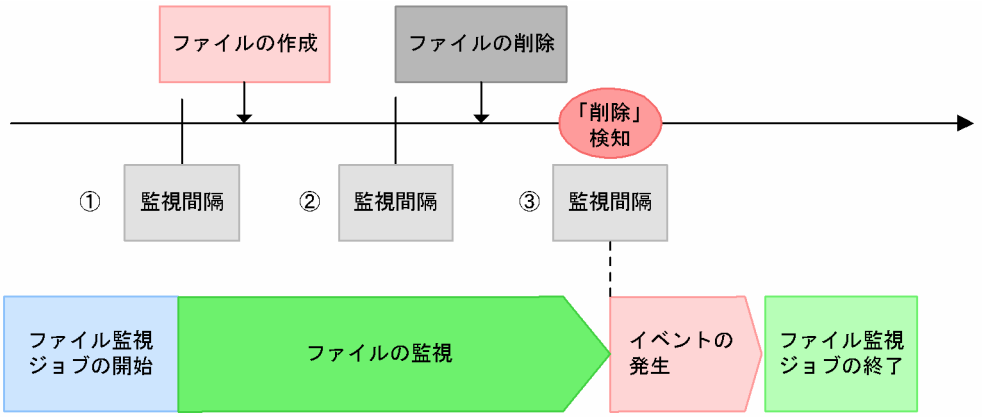
(b)　監視条件で「削除」「サイズ変更」または「最終書き込み時刻変更」を指定した場合

監視条件で「削除」，「サイズ変更」，または「最終書き込み時刻変更」を指定して監視している場合の，ファイル監視ジョブの基本動作を次の図に示します。この例では，監視条件に「削除」を指定したものとします。

■ ジョブの開始時に監視対象ファイルがないとき

ジョブの開始時に監視対象ファイルがないときは，次のように動作します。

図 7-5　ジョブの開始時に監視対象ファイルがないとき



①：監視間隔ごとに監視対象ファイルをチェックします。この時点では，監視対象ファイルがないことを認識しています。

②：監視対象ファイルが作成されたことを認識しています。

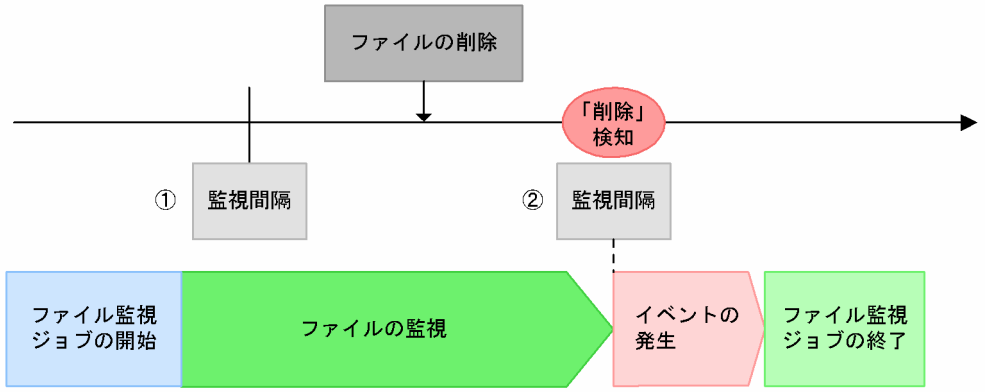
③：監視対象ファイルが削除されていることを検知して，ファイル監視イベントが発生します。

注　監視条件が「サイズ変更」または「最終書き込み時刻変更」の場合には，図中の「ファイルの削除」の部分を，「ファイルサイズの変更」または「ファイルの上書き更新（最終書き込み時刻を変更）」と読み替えてください。

■ ジョブの開始時に監視対象ファイルがあるとき

ジョブの開始時に監視対象ファイルがあるときは，次のように動作します。

図 7-6　ジョブの開始時に監視対象ファイルがあるとき



①：監視間隔ごとに監視対象ファイルをチェックします。この状況では，監視対象ファイルがあることを認識しています。

②：監視対象ファイルが削除されていることを検知して，ファイル監視イベントが発生します。

注　監視条件が「サイズ変更」または「最終書き込み時刻変更」の場合には，図中の「ファイルの削除」の部分を，「ファイルサイズの変更」または「ファイルの上書き更新（最終書き込み時刻を変更）」と読み替えてください。

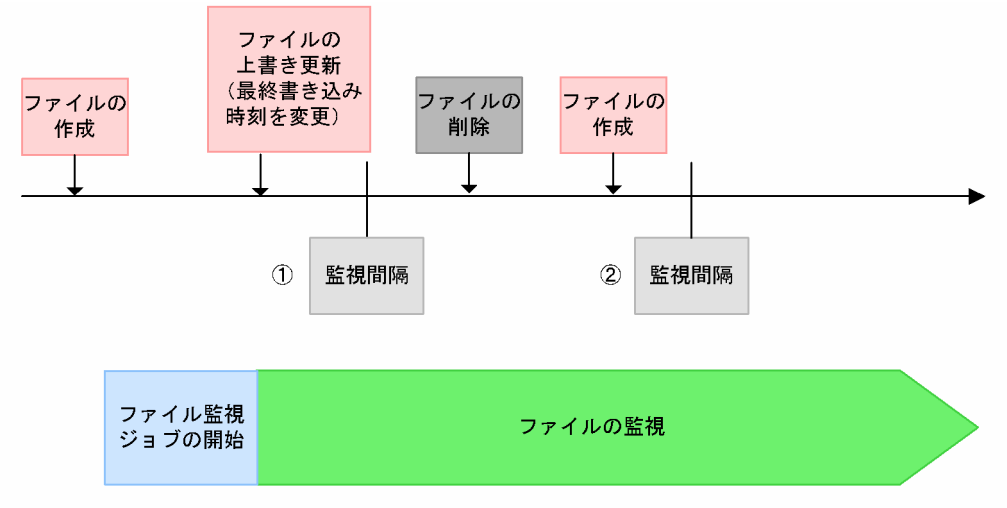
(c)　監視間隔内に複数回イベントが発生した場合（監視条件が「作成」の場合）

監視条件で「作成」を指定して監視していて，監視間隔内に複数回ファイルが更新されたときの，ファイル監視ジョブの基本動作を次の図に示します。

■ ファイルの作成を検知しないとき

ファイルの作成を検知しないときは，次のように動作します。

図 7-7　検知しない例



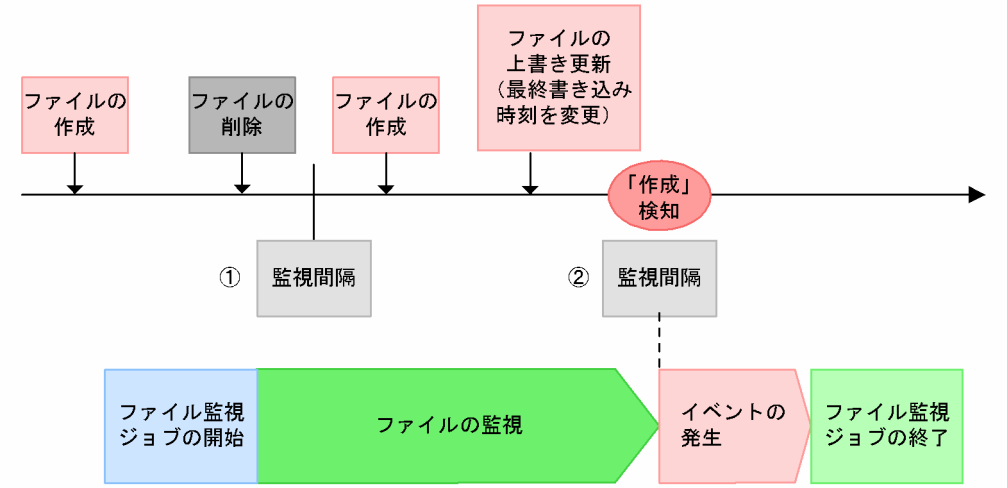
①：監視対象ファイルが作成されたのは，ファイル監視ジョブ開始前なので，この時点では監視対象ファイルの作成を検知しません。ただし，最終書き込み時刻が変更されたことは認識しています。

②：監視対象ファイルは削除されたあとに作成されていますが，ファイル監視ジョブは①の状態と比較するため，監視対象ファイルが作成されたとみなされません。そのため，ファイル監視イベントは発生しません。

■ ファイルの作成を検知するとき

ファイルの作成を検知するときは，次のように動作します。

図 7-8　検知する例



①：ファイル監視ジョブ開始前に作成されていた監視対象ファイルが，削除されたことを認識します。

②：監視対象ファイルは，作成されたあとに最終書き込み時刻が変更されていますが，ファイル監視ジョブは①の状態と比較するため，監視対象ファイルが作成されたことを検知してファイル監視イベントが発生します。

(2) ファイル名の指定方法

ファイル名の指定方法には，絶対パス名でファイル名を指定する方法のほかに，絶対パス名と「*（ワイルドカード）」を使用した総称名で指定する方法があります。ワイルドカードを使った監視ファイル名の指定例を次に示します。

表 7-3 ワイルドカードを使った監視ファイル名の指定例

指定例	指定内容	監視ファイル名の例
/jp1/*	/jp1 にあるファイルすべてを監視します。ただし，先頭が「.」（ピリオド）で始まるファイルは監視しません。	/jp1/aaa /jp1/aaa.sh
/jp1/.*	/jp1 にあるファイルで，「.」（ピリオド）で始まるファイルすべてを監視します。	/jp1/.aaa
/jp1/aaa*	/jp1 にあるファイルで，「aaa」という文字列で始まるファイルすべてを監視します。	/jp1/aaa /jp1/aaabbb /jp1/aaa.sh
/jp1/*aaa	/jp1 にあるファイルで，「aaa」という文字列が最後に付くファイルすべてを監視します。	/jp1/aaa /jp1/bbbaaa /jp1/bbb.aaa
/jp1/aaa*bbb	/jp1 にあるファイルで，「aaa」という文字列で始まり，「bbb」という文字列が最後に付くファイルすべてを監視します。	/jp1/aaabbb /jp1/aaacccbbb /jp1/aaa.bbb
/jp1/*aaa*	/jp1 にあるファイルで，「aaa」という文字列がファイル名に含まれているファイルすべてを監視します。	/jp1/aaa /jp1/bbbaaa /jp1/bbbaaaccc /jp1/bbbaaa.sh

注

このファイル名の指定例は，すべて UNIX の場合の例です。Windows でファイル名を指定する場合は，「c:¥jp1¥*」のようになります。なお，次の場合は，実行時にエラーとなります。

- ディレクトリ名にワイルドカードを指定した場合
- 相対パスで指定した場合
- 指定したファイル名が，ディレクトリとしてすでにある場合（例えば，「/jp1/aaa」というファイル名を指定したときに，すでに「/jp1/aaa」というディレクトリがある場合）
- ワイルドカードを指定したファイル名の上位ディレクトリがない場合（例えば，「/jp1/*」というファイル名を指定したときに，「/jp1/」というディレクトリがない場合）

(3) ファイル監視ジョブのオプション

ファイル監視ジョブには，次の二つのオプションを設定できます。

- 監視開始オプション
- ファイル監視ジョブの状態引き継ぎオプション

上記のオプションについて，次に説明します。

(a)　監視開始オプション

ファイル監視ジョブが実行中の状態になったときに，すでに監視対象ファイルがある場合，ファイル監視ジョブの監視条件を成立させるかどうかをオプションで指定します。このオプションを，**監視開始オプション**といいます。

なお，監視開始オプションは，ファイルの監視条件で「作成」を選択した場合に有効になります。また，オプションを指定しない場合は，ファイルがあっても，監視条件を成立させません。

監視開始オプションを設定した場合のオプションの基本動作例を紹介します。

■ 監視開始オプションを設定していない場合の動作例

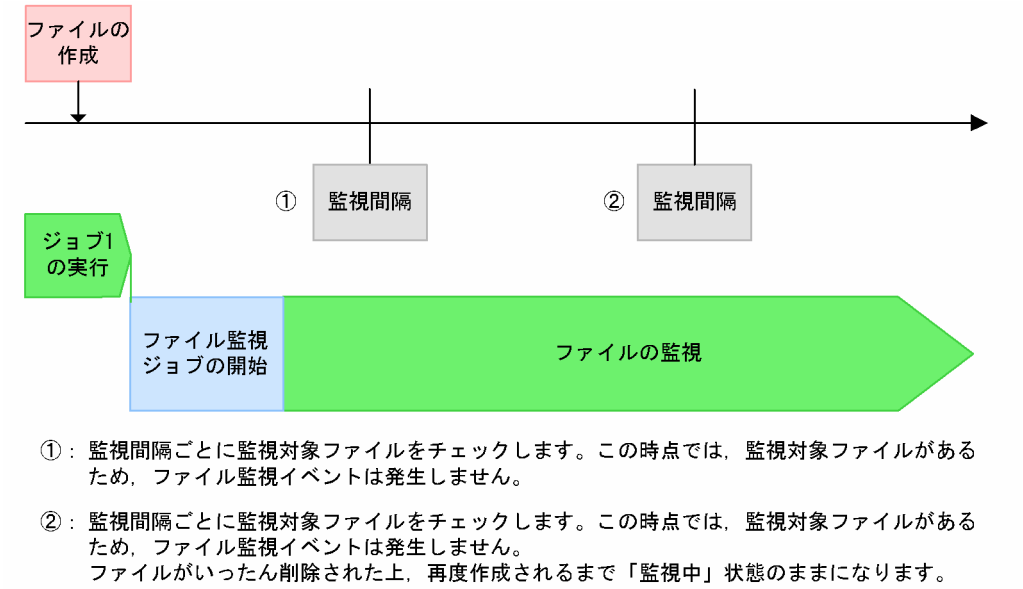
監視開始オプションを設定していない場合の動作例を次の図に示します。

なお，図中の「ファイル」とは，「監視対象ファイル」を指します。

ファイル監視ジョブの実行前に監視対象ファイルが作成されたとき

ファイル監視ジョブの実行前に監視対象ファイルが作成されたときは，次のように動作します。

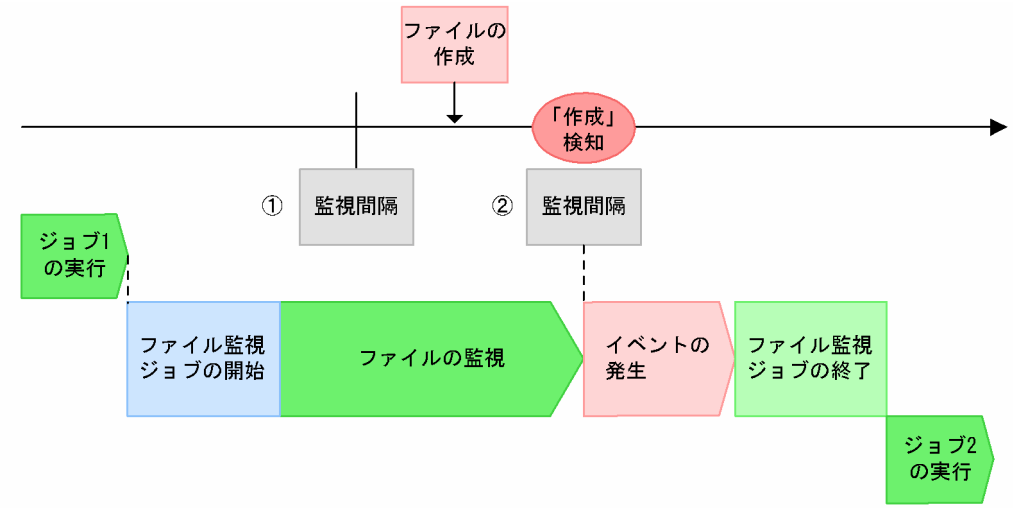
図 7-9　ファイル監視ジョブの実行前に監視対象ファイルが作成されたとき



ファイル監視ジョブの実行後に監視対象ファイルが作成されたとき

ファイル監視ジョブの実行後に監視対象ファイルが作成されたときは，次のように動作します。

図 7-10　ファイル監視ジョブの実行後に監視対象ファイルが作成されたとき



①：監視間隔ごとに監視対象ファイルをチェックします。この状況では，監視対象ファイルがないことを認識しています。

②：監視対象ファイルが作成されていることを検知して，ファイル監視イベントが発生します。

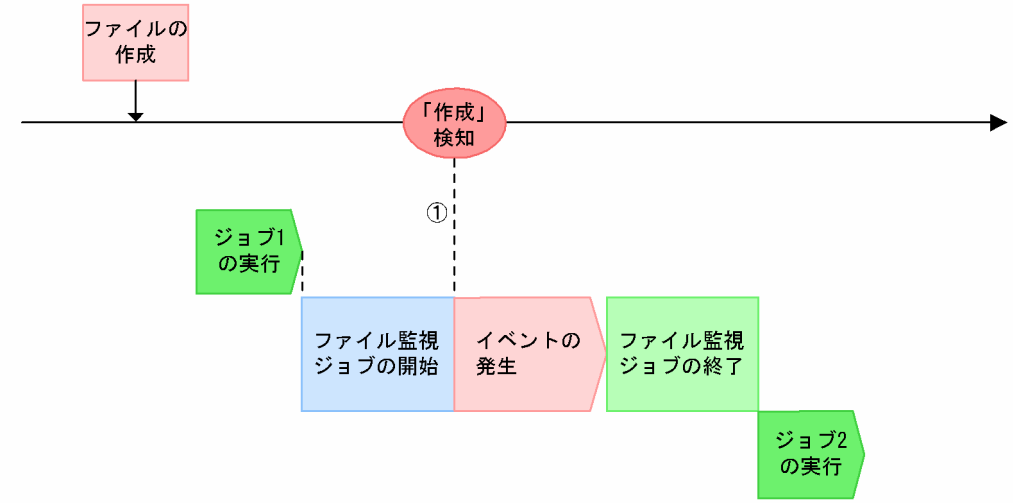
■ 監視開始オプションを設定している場合の動作例

監視開始オプションを設定している場合の動作例を次の図に示します。

ファイル監視ジョブの実行前に監視対象ファイルが作成されたとき

ファイル監視ジョブの実行前に監視対象ファイルが作成されたときは，次のように動作します。

図 7-11　監視対象ファイルをファイル監視ジョブの実行前に作成した場合

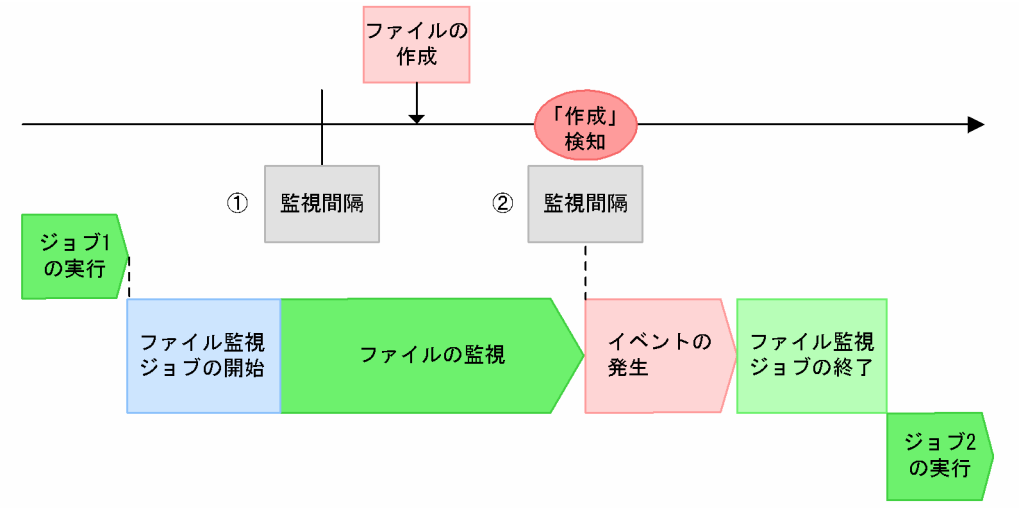


①：ファイル監視ジョブが開始された時点で，監視対象ファイルがあるため，ファイル監視イベントが発生します。

ファイル監視ジョブの実行後に監視対象ファイルが作成されたとき

ファイル監視ジョブの実行後に監視対象ファイルが作成されたときは，次のように動作します。

図 7-12　監視対象ファイルをファイル監視ジョブの実行後に作成した場合



①：　ファイル監視ジョブが開始された時点で監視対象ファイルがなかったため，監視間隔ごとに監視対象ファイルをチェックします。この時点では，監視対象ファイルがないことを認識しています。

②：　監視対象ファイルが作成されていることを検知して，ファイル監視イベントが発生します。

監視開始オプションの指定内容によって，ファイル監視ジョブがどのように動作するかを，次に説明します。

「既存ファイルも条件成立」を指定した場合

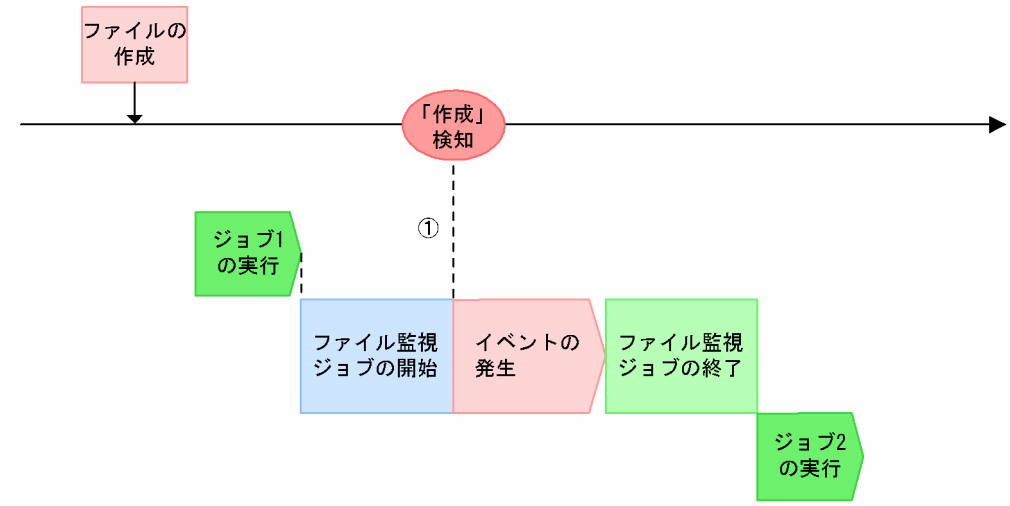
ファイルの新規作成を監視しているファイル監視ジョブが「実行中」状態になったときに監視対象ファイルがすでにあると，監視条件が成立し，ファイル監視ジョブが正常終了します。ただし，監視対象ファイルのクローズチェックが実行されるため，監視対象ファイルが使用中の場合，ファイル監視ジョブは「監視中」状態のままになります。

なお，ジョブの種類によって，監視開始オプションで監視条件を成立させるファイル監視ジョブが異なります。監視開始オプションの動作を，ファイル監視ジョブの種類ごとに示します。

ジョブネット中の監視対象ファイル名を指定したファイル監視ジョブ

ファイル監視ジョブの実行時に監視対象ファイルがある場合，監視開始オプションによって監視条件が成立し，即時にイベントが発生してファイル監視ジョブは正常終了します。

図 7-13　ジョブネット中の監視対象ファイル名を指定したファイル監視ジョブ

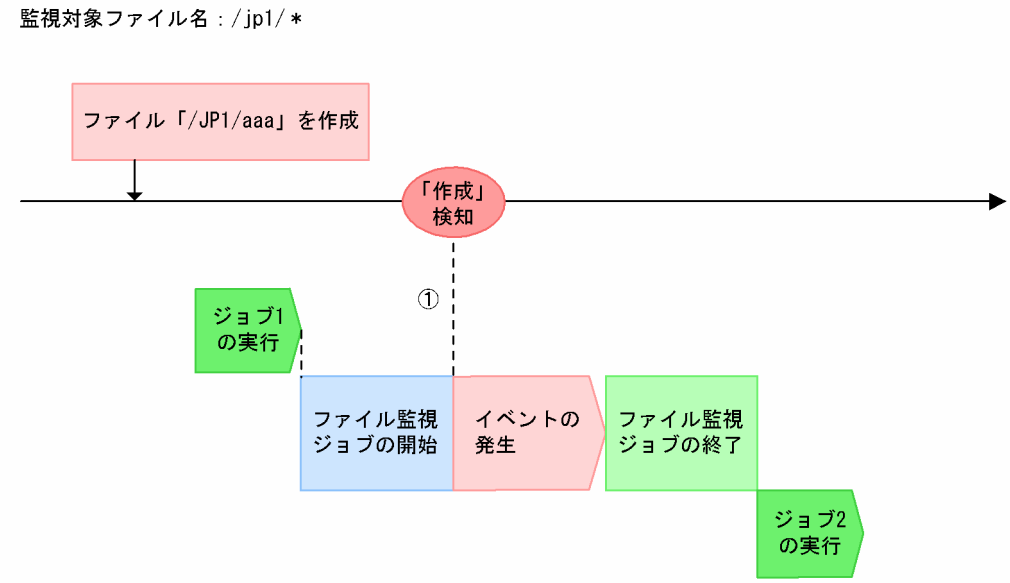


①：ファイル監視ジョブが開始された時点で，監視対象ファイルがあるため，ファイル監視イベントが発生します。

ジョブネット中の監視対象ファイル名を「*（ワイルドカード）」で指定したファイル監視ジョブ

ファイル監視ジョブの実行時に監視対象のディレクトリに一つでもファイルがある場合は，監視開始オプションによって監視条件が成立し，即時にイベントが発生してファイル監視ジョブは正常終了します。

図 7-14　ジョブネット中の監視対象ファイル名を「*（ワイルドカード）」で指定したファイル監視ジョブ



①：ファイル監視ジョブが開始された時点で，「/JP1」ディレクトリの下にファイルがあるため，ファイル「/jp1/aaa」に関するファイル監視イベントが発生します。

補足事項

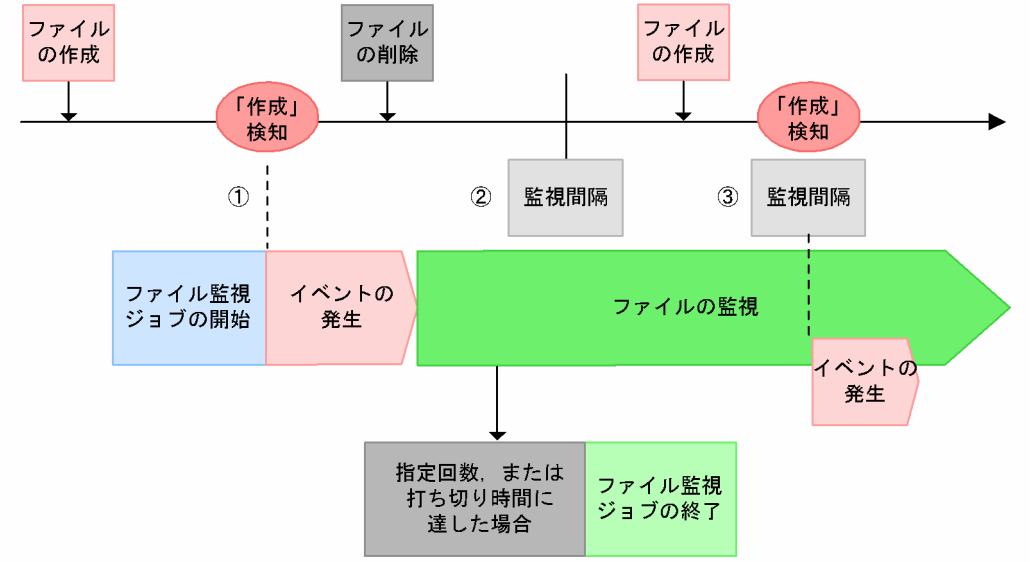
「*（ワイルドカード）」の使用については，「(2) ファイル名の指定方法」を参照してください。

起動条件中の，監視対象ファイル名を指定したファイル監視ジョブ

起動条件中のファイル監視ジョブの実行時に監視対象ファイルがある場合は，監視開始オプションによって監視条件が成立し，即時にイベントが発生します。

監視開始オプションは，イベントが 1 回成立した時点で満了します。イベント発生後もファイル監視ジョブは「監視中」状態となって監視を継続します。ただし，オプション満了後は，新規ファイルの作成時に条件が成立します。

図 7-15　起動条件中の監視対象ファイル名を指定したファイル監視ジョブ



①： ファイル監視ジョブが開始された時点で，監視対象ファイルがあるため，イベントが発生する条件が満たされています。ファイル監視イベントが発生します。

②： 監視対象ファイルが削除されたことを認識しています。

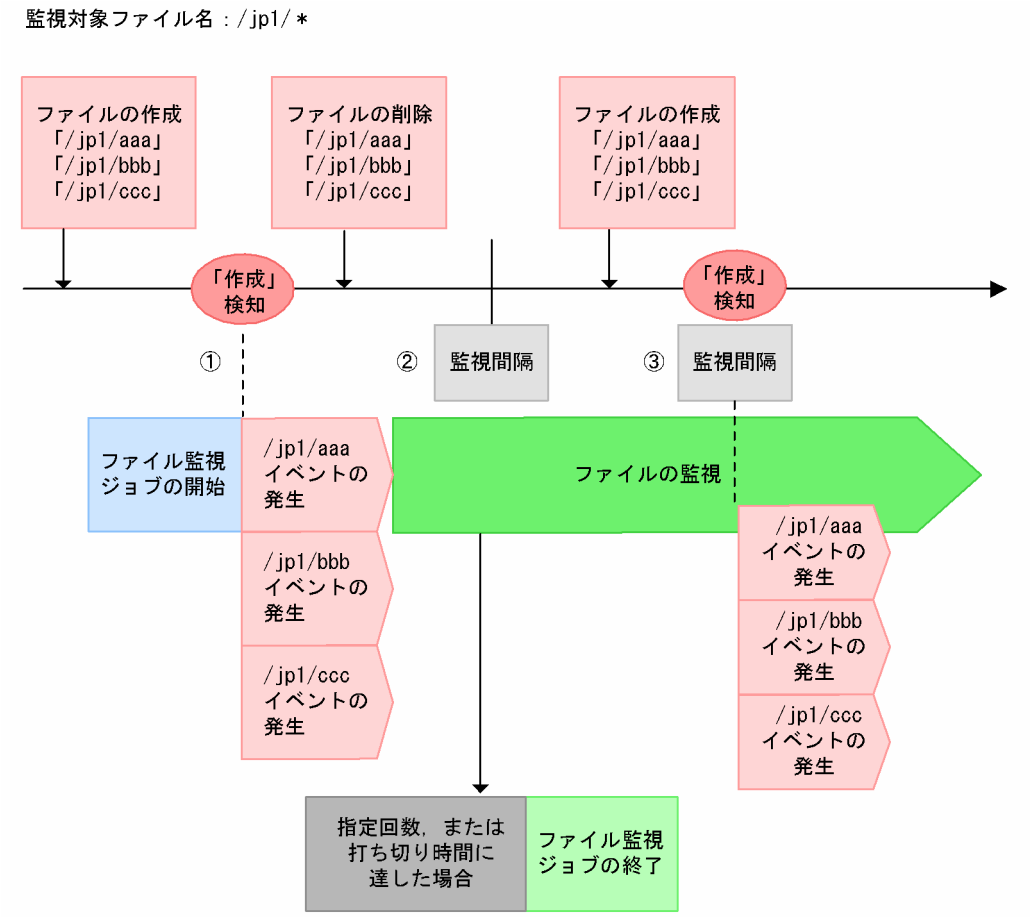
③： 監視対象ファイルが作成されていることを検知して，ファイル監視イベントが発生します。

起動条件中の，監視対象ファイル名を「*（ワイルドカード）」で指定したファイル監視ジョブ

起動条件中のファイル監視ジョブの実行時に，監視対象のディレクトリにあるすべてのファイルに対して，監視開始オプションによって監視条件が成立し，即時イベントが発生します。

監視開始オプションは，各ファイルに対してイベントが 1 回ずつ成立した時点で満了します。イベント発生後もファイル監視ジョブは「監視中」状態となって監視を継続します。ただし，オプションが成立したファイルは，新規ファイルの作成時に条件が成立します。

図 7-16　起動条件中の監視対象ファイル名を「*（ワイルドカード）」で指定したファイル監視ジョブ



①：ファイル監視ジョブが開始された時点で，監視対象ファイルがあるため，イベントが発生する条件が満たされています。ファイル監視イベントが発生します。
「*（ワイルドカード）」の条件に当てはまる，「/jp1/aaa」，「/jp1/bbb」，「/jp1/ccc」すべてのイベントが発生します。

②：監視対象ファイルが削除されたことを認識しています。

③：監視対象ファイルが作成されていることを検知して，ファイル監視イベントが発生します。
「*（ワイルドカード）」の条件に当てはまる，「/jp1/aaa」，「/jp1/bbb」，「/jp1/ccc」すべてのイベントが発生します。

補足事項

「*（ワイルドカード）」の使用については，「(2) ファイル名の指定方法」を参照してください。

「新規ファイル作成時に条件成立」を指定した場合（デフォルトの場合）

ファイルの「作成」を監視しているファイル監視ジョブが「実行中」状態になったときに，すでに監視対象ファイルがあっても，監視条件成立にはなりません。ファイル監視ジョブは監視を継続します。

なお，クラスタで JP1/AJS3 を運用している場合，JP1/AJS3 サービスがフェールオーバーしたあとに，監視開始オプションが設定されているファイル監視ジョブが再度実行されると，「ファイル監視ジョブの状態引き継ぎ」を有効にしている場合は，フェールオーバー前の状態で実行され，「ファイル監視ジョブの状態引き継ぎ」を無効にしている場合は，監視開始オプションが再び有効になります。

ファイル監視ジョブの監視開始オプションの指定方法については，マニュアル「JP1/Automatic Job Management System 3 操作ガイド　12.4.20 ［詳細定義－［ファイル監視］］ダイアログボックス」またはマニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　5.2.10 ファイル監視ジョブ定義情報の記述方法」を参照してください。

(b)　ファイル監視ジョブの状態引き継ぎオプション

ファイル監視ジョブが監視対象ファイルを監視している間の情報を，随時退避しておき，ファイル監視ジョブの監視状態を引き継ぐことができます。状態引き継ぎを行うように設定した場合，ファイル監視ジョブ一つにつき，一つの状態引き継ぎ情報格納ファイルが生成されます。

例えば，クラスタで，ファイル監視ジョブの実行中に JP1/AJS3 サービスが停止した場合，JP1/AJS3 サービスの停止後，再度 JP1/AJS3 サービスが起動して，サービスが停止する前と同じファイル監視ジョブが実行されたときに，サービスが停止する前のファイル監視ジョブの監視状態を引き継いでジョブが実行されます。なお，クラスタでなくても，ファイル監視ジョブの情報を引き継げます。

監視状態を引き継ぐには，ファイル監視ジョブが継続して動作している必要があります。ファイル監視ジョブが継続して動作するか，終了するかによって，監視状態が引き継がれるかどうかが異なります。

監視状態が引き継がれる条件を，次の表に示します。

表 7-4　監視状態の引き継ぎ条件

ファイル監視ジョブの種類	サービスの再起動	停止を伴う フェールオーバー	システムダウンによるフェールオーバー
ジョブネット中の ファイル監視ジョブ	ジョブが終了するため，引き継がれない※	ジョブが終了するため，引き継がれない※	引き継がれる
起動条件中のファイル監視ジョブ	引き継がれる	引き継がれる	引き継がれる

注※

イベントジョブ実行継続オプションを併用していても，実行エージェントの JP1/AJS サービスの再起動または停止を伴うフェールオーバー時にはファイル監視ジョブがいったん終了するため，表 7-4 に示したケースに該当します。そのため，ファイル監視ジョブが実行中である実行エージェントの JP1/AJS サービスが停止し，再起動してイベント監視を再開するまでの間に発生したイベントについては検知されません。

また，その際に実行エージェント側の統合トレースログにメッセージ KAVT2031-E および KAVT2034-W が出力されます。イベントジョブ実行継続オプションを併用している場合は，これらのメッセージが出力されても無視して運用を継続してください。イベントジョブ実行継続オプションの詳細については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　8.2.1 JP1/AJS3 のサービスが停止してもイベントジョブの実行を継続させる」を参照してください。

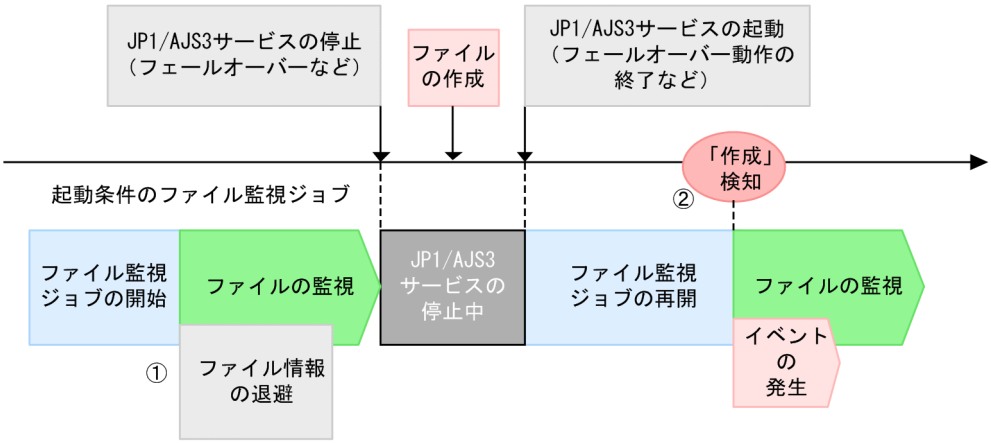
なお，次のような場合には，状態引き継ぎ情報格納ファイルが削除されます。

- 各ファイル監視ジョブに対して作成される状態引き継ぎ情報格納ファイルは，ファイル監視ジョブの終了時に削除されます。
- 状態引き継ぎを行ったあと，状態引き継ぎを行わない設定に変更した場合，状態引き継ぎ情報格納ファイルはすべて削除されます。
- コールドオプションを設定して JP1/AJS3 サービスを起動した場合，状態引き継ぎ情報格納ファイルはすべて削除されます。

ファイル監視ジョブの状態引き継ぎオプションを設定した場合の動作例を次の図に示します。

図 7-17　ファイル監視ジョブの状態引き継ぎオプションを設定した場合の動作例

監視条件：「作成」を指定している場合



①：　状態引き継ぎオプションが有効になっている場合，ファイル監視ジョブが保持している監視対象ファイルの情報が，引き継ぎ情報格納ファイルに退避されます。

②：　JP1/AJS3サービスが起動して，サービス停止前のファイル監視ジョブが実行されると，状態引き継ぎ情報格納ファイルが読み込まれます（状態が引き継がれる条件については，「表7-4 監視状態の引き継ぎ条件」を参照してください）。
そのため，監視対象ファイルが作成されていることを検知して，ファイル監視イベントが発生します。

ファイル監視ジョブの監視状態を引き継ぐ機能は，デフォルトでは無効になっています。有効にするには，ファイル監視ジョブを実行するすべてのホストおよびノードで設定を行ってください。クラスタの場合は，実行系・待機系ともに設定が必要になります。設定手順については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.3.3 ファイル監視ジョブの状態引き継ぎオプションの設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.3.3 ファイル監視ジョブの状態引き継ぎオプションの設定」（UNIX の場合）を参照してください。

なお，2 ギガバイト以上のファイルを監視する設定（ラージファイル対応オプション）を「対応する」から「対応しない」に変更し，2 ギガバイト以上のファイルの情報を引き継いだ場合は，統合トレースログと実行結果詳細にメッセージ KAVT2038-E が出力され，ジョブは異常検出終了します。

(4)　ファイル監視ジョブ定義時の注意事項

ファイル監視ジョブを定義するときの注意事項を次に示します。

- 監視間隔を 1～9 秒に設定したファイル監視ジョブを実行した場合，ファイルの更新発生（イベント発生）に対して，即時（1～9 秒）に後続ジョブが実行されるわけではありません。特に，多数のファイル監視ジョブを実行する場合，後続ジョブの実行までに時間が掛かることがあります。この場合，監視間隔には十分に余裕を持った値を設定してください。

  なお，ファイル監視ジョブの監視間隔時間の見積もり式については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 3.1.6 ファイル監視ジョブ使用時に設定する監視間隔」を参照してください。

- ファイル監視ジョブに指定する監視対象ファイル名を，「*（ワイルドカード）」を使用した総称名で指定し，かつ監視対象ファイル名に合致するファイルが多い場合，ファイル監視ジョブの動作が遅延したり，ファイル監視プロセスの CPU 使用率が高くなったりします。監視対象ファイル名を総称名で指定するときは，監視対象ファイル名に合致するファイルが必要最小限となるように監視対象ファイル名を指定したり，不要になったファイルを削除してファイル数を減らしたりすることを推奨します。または，監視対象ファイル名は完全名で指定してください。

- ジョブ実行時にディスクがマウントされたり，アンマウントされたりする環境にあるファイルを，ファイル監視ジョブの監視対象にしないでください。このようなファイルを監視対象にした場合，監視処理が正常に動作しなくなったり，ファイルの作成や削除を意図しないタイミングで検知してしまったりすることがあります。

- Windows の場合，ファイル監視ジョブが監視できるファイルは，デフォルトではファイルサイズが 2 ギガバイト（2,147,483,647 バイト）までのファイルです。これより大きいファイル（ラージファイル）を監視したい場合は，環境設定が必要です。設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド 6.3.16 ラージファイルを監視するための設定」を参照してください。なお，バージョン 11-10 以降の UNIX 版の JP1/AJS3 の場合は，この設定とは関係なくラージファイルを監視できます。

  Windows 版の JP1/AJS3 の場合，ラージファイルを監視するための設定をしたとき，または，バージョン 11-10 より前のバージョンからバージョン 11-10 以降へ UNIX 版の JP1/AJS3 をバージョンアップしてラージファイルを監視できるようになったときは，次に示す注意事項があります。

  - ファイル監視ジョブの引き継ぎ情報FLSIZE を UNIX ジョブなどのユーザープログラムで参照・使用している場合，2 ギガバイト以上の値が引き継がれるおそれがあります。この場合，ユーザープログラムのファイルサイズを扱う処理でけたあふれが発生するおそれがあるため，ユーザープログラムの修正などの対処が必要です。

- ファイル監視ジョブの監視対象ファイル名は，JP1/AJS3 が動作している環境変数LANG に準じた名称にしてください。それ以外の文字コードを使用したファイルを監視する場合の動作は保証できません。

- ファイル監視ジョブで監視対象となっているファイルが，別のイベントジョブまたは JP1/Base のイベントサービス機能を使用したログファイル監視の対象にもなっている場合，監視対象ファイルはオープンされていると認識されるため，監視条件は成立しません。そのため，別のイベントジョブや JP1/Base のイベントサービス機能を使用したログファイル監視の対象ファイルは，ファイル監視ジョブの監視対象としないでください。

- syslog やデバイススペシャルファイルなどの OS のシステムファイルは，不特定多数のプロセスのアクセス，ファイルへの頻繁なアクセス，およびファイルを使用しているプロセスを判断できないケース

があるため，監視対象としないでください。また，RAW ファイルについてもファイルの更新，およびファイルを使用しているプロセスを判断することができないため，監視対象としないでください。

- ファイル監視ジョブの監視対象ファイル名に，ネットワークを介したファイル※の名称を指定する場合は，環境設定パラメーターNetworkFilewatch に「Y」を設定してください。環境設定パラメーターNetworkFilewatch の詳細については，「JP1/Automatic Job Management System 3 構築ガイド 20.6.2(30) NetworkFilewatch」を参照してください。

  環境設定パラメーターNetworkFilewatch に「Y」を設定しないでネットワークを介したファイルの名称を指定して運用した場合，ネットワーク状況によって，ファイル監視ジョブが意図しない動作をすることがあります。

  指定したいファイルの参照先が自ホスト上のファイルの場合は，ネットワークを介さないファイルの名称を指定してください。

  指定したいファイルの参照先が他ホスト上のファイルの場合は，次のように監視してください。

  - 監視するファイルが存在するホストに JP1/AJS3 をインストールし，自ホストのファイルとして監視する
  - 監視するファイルが存在するホストから，JP1/AJS3 がインストールされているホストに FTP などで転送して，監視する

  注※

  ネットワークを介したファイルの例を次に示します。

  Windows：ネットワークドライブおよび UNC で参照できるファイル

  UNIX：NFS マウントで参照できるファイル

- 環境設定パラメーターNetworkFilewatch に「Y」を設定しない場合で，ファイル監視ジョブの実行先ホストが UNIX のとき，ネットワークを介して他ホストから操作（作成，更新，または参照）されるファイルは監視しないでください。ファイルがネットワークを介して他ホストから操作された場合，ファイルの操作中に条件成立したり，1 回のファイル操作で複数回条件成立したりするおそれがあります。
- 頻繁にファイルのオープン・クローズを行い，データを追加書きするようなファイルについては，ファイル監視ジョブで正しく監視できません。ファイル監視ジョブの監視するタイミングと，ファイルのクローズが重なると，監視対象ファイルの更新処理が終了したと見なしてイベントが発生することがあります。このようなファイルを監視する場合には，直接ファイル監視ジョブで監視するのではなく，ファイル更新が終わったときに処理の完了を示す別ファイルを作成し，そのファイルを監視するような運用を推奨します。
- 「*（ワイルドカード）」を含む監視対象ファイル名で監視している場合の注意事項を次に示します。
  - 同一の監視間隔内で発生したファイル監視イベントについて，ファイルの作成順，更新順，および削除順にファイル監視を行うことはできません。なお，イベント順序保証オプションは，発生したファイル監視イベントの順序が通信の状態などによって変更されないことを保証するものであり，ファイル監視ジョブの動作を変更するものではありません。
- ファイル監視ジョブの監視条件成立に関する注意事項を次に示します。
  - 監視条件に「ファイルの削除」を指定し，監視開始時に監視対象ファイルがなかった場合，そのファイルが一度作成され，その後削除されてから監視条件成立となります。

- 監視条件に「ファイルのサイズ変更」または「ファイルの最終書き込み時刻変更」を指定し，監視開始時に監視対象ファイルがなかった場合，そのファイルが一度作成され，その後サイズの変更，または最終書き込み時刻の変更が行われてから監視条件成立となります。

  なお，ファイルのサイズ変更，および最終書き込み時刻変更は，ディスクに実際に書き込まれてから監視条件成立となります。エディターなどで編集しているだけでは，監視条件成立にはなりません。

- ファイル監視ジョブの状態引き継ぎオプションを設定していないと，ファイル監視ジョブがファイルを監視中に JP1/AJS3 サービスが停止したときに，ファイル監視ジョブが監視中に持っていた情報が失われます。再度 JP1/AJS3 サービスを開始してファイル監視を実行した場合，前回のファイル監視情報は引き継がれません。

- ファイル監視ジョブは，監視間隔に指定した間隔でファイルをチェックします（「*」指定時は監視しているそれぞれのファイルをチェックします）。そのため，ある監視時刻から次回監視時刻までの間に，監視している一つのファイルに対して監視条件に合うファイル更新が複数回行われた場合でも，最後に行われた更新だけを検知します。また，多数のジョブで一度にファイル更新を検知すると，それぞれのジョブでイベント発行のための処理を行うため，次回監視時刻が遅れることがあります。次回監視時刻が遅延している間に，監視条件に合うファイル更新が複数回行われた場合でも，最後に行われた更新だけを検知します。

  短い監視間隔を指定した場合，多数のジョブを監視していると，監視に監視間隔以上の時間が掛かることがあります。

- ファイル監視ジョブは，監視対象ファイルの状態の変更を検知した場合，クローズチェックによって監視対象ファイルをオープンしているプロセスがないかどうかをチェックしています。そのとき，ファイルがオープンされていないと判断した場合に，イベントが発生します。監視対象ファイルがオープンされている場合は，その時点でのイベントは発行しないで，再び監視中の状態になります。次回監視間隔の経過後に，再び監視対象ファイルをオープンしているプロセスがないかどうかをチェックします。つまり，監視対象ファイルがオープンされている間は，監視対象ファイルの状態の変更を検知していても，イベントは発行しません。なお，Windows で NFS サーバ上のファイルを監視する場合，または UNIX でネットワークを介したファイルを監視する場合，他ホストのプロセスがファイルをオープンしているかどうか確認できません。このため，監視中に他ホストのプロセスによってファイルがオープンされていても，イベントを発行します。

- ファイル監視ジョブの状態引き継ぎオプションを設定していて，退避した JP1/AJS3 のユニットを実行する場合や，バックアップした JP1/AJS3 のインストール先ディレクトリ下の環境を使用するときは，JP1/AJS3 サービスをコールドスタートしておく必要があります。JP1/AJS3 サービスをコールドスタートしないで，バックアップした JP1/AJS3 のファイル監視ジョブを実行した場合，退避前の監視状態から監視を始めます。

- 情報格納ファイルや情報格納フォルダに何らかの異常が発生して，引き継ぎ状態を格納するファイルの作成や書き込みができない場合，メッセージ KAVT2034-W を統合トレースログ出力してからジョブの実行を継続します。この場合，監視状態を引き継ぐためのファイルは生成されません。

- ファイル監視ジョブの状態引き継ぎを設定していて，起動条件のファイル監視ジョブを実行して，JP1/AJS3 サービスを強制終了させた場合は，ファイル監視ジョブの監視状態を引き継ぎますが，JP1/AJS3 サービスを計画終了させた場合には，ファイル監視ジョブの状態は引き継ぎません。

- 次に示すディレクトリには，ファイル監視プロセスの起動中，随時読み込み，書き込み処理を行っているファイルがあります。そのため，次に示すディレクトリにあるファイルは監視対象に指定しないでください。

  物理ホストでの運用時

  Windows で，インストール先フォルダがデフォルトまたはシステムで保護されたフォルダ配下の場合

  %ALLUSERSPROFILE%¥Hitachi¥JP1¥JP1_DEFAULT¥JP1AJS2¥jp1ajs2¥sys

  「%ALLUSERSPROFILE%」のデフォルトは「システムドライブ¥ProgramData」です。

  「システムで保護されたフォルダ」とは，次のパスを指します。

  ・「システムドライブ¥Windows」配下

  ・「システムドライブ¥Program Files」配下

  ・「システムドライブ¥Program Files (x86)」配下

  Windows で，インストール先フォルダが上記以外の場合

  JP1/AJS3 のインストール先フォルダ¥jp1ajs2¥sys

  UNIX の場合

  /var/opt/jp1ajs2/sys

  論理ホストでの運用時

  Windows の場合

  共有フォルダ¥jp1ajs2¥sys

  UNIX の場合

  共有ディレクトリ/jp1ajs2/sys

- Windows でファイル監視ジョブを使用する場合で，次の二つの条件を満たすとき，予期したファイルとは異なるファイルの変更を検知することがあります。

  1. ファイル監視ジョブの監視対象ファイル名を，拡張子は 3 文字で固定して指定し，その他の部分はワイルドカードで指定している場合。

     (例)

     「C:¥Temp¥*.txt」のように監視対象ファイルを指定している場合。

  2. ファイル監視ジョブの監視フォルダの下に，拡張子が 4 文字以上，かつ拡張子の 3 文字目までが，1 で指定した監視対象ファイル名の拡張子と一致するファイルがある場合。

     (例)

     「*.txt」という監視対象ファイル名を監視している場合（ワイルドカードを使用した前方一致の監視条件）で，監視フォルダの下に，「aaa.txt」，「aaa.txta」，「aaa.txtab」という三つのファイルがあり，どれかが更新されたとき，どのファイルが更新されても，ファイル監視ジョブによって監視条件と一致したと見なされてしまいます。拡張子の指定のしかたで，どのようなファイルが監視対象となるかの例を次の表に示します。

表 7-5　拡張子による監視対象ファイルの検知の有無

監視対象ファイル名	更新されるファイル名			
	aaa.tx	aaa.txt	aaa.txta	aaa.txtab
C:¥Temp¥*.tx	○	×	×	×
C:¥Temp¥*.txt	×	○	○	○
C:¥Temp¥*.txta	×	×	○	×
C:¥Temp¥*.txtab	×	×	×	○

（凡例）

○：更新があった場合，イベントが発行される。

×：更新があった場合，イベントが発行されない。

- 転送ツールを使用して，監視対象ファイルの更新などをしている場合，ファイルを上書きしたときでも，「削除」，または「作成」という監視条件が成立したと見なされることがあります。意図しない監視条件で成立するのは，使用しているツールの内部処理にファイルを削除する処理が入っている場合です。削除する処理と監視間隔のタイミングが一致すると，ファイル監視ジョブにファイルが削除されたと見なされることが原因です。この現象の発生を回避するためには，監視対象ファイルが常に上書きされる運用であっても，「作成」と「最終書き込み時刻変更」という監視条件を使用して監視することを推奨します。
- 起動条件に指定されたファイル監視ジョブが監視中のとき,ファイル監視ジョブの状態引き継ぎオプションを無効から有効に変更し，JP1/AJS3 を再起動するとメッセージ KAVT2031-E および KAVT2034-W が統合トレースログに出力されます。これは，JP1/AJS3 が再起動前にファイル監視ジョブの状態引き継ぎオプションが設定されていなかったため，再起動前のファイル監視ジョブの監視状態を引き継げなかったことを意味します。再起動前の状態を引き継ぐことができないので，ファイル監視ジョブが監視中のときには，ファイル監視ジョブの状態引き継ぎオプションを無効から有効に変更しないでください。
- Windows でファイル監視ジョブを使用する場合，監視対象ファイル名に「*.*」を指定すると，拡張子のないファイル名のファイルも検知します。

  例えば，監視対象ファイル名に「C:¥temp¥*.*」を指定した場合，C:¥temp 下の次に示すファイル名のファイルが更新されるとイベントが発行されます。

  - abc
  - abc.txt
  - abc.txt.txt
- 監視対象ファイルがシンボリックリンクの場合は，監視できません。ただし，監視対象ファイル名にシンボリックリンクしたディレクトリが含まれていても，ファイル自体がシンボリックリンクでなければ監視できます。
- ファイル監視ジョブは，監視対象ファイルの状態に変更がないかを監視間隔ごとに調べます。変更を検知したファイルについては，指定された監視条件（作成，サイズ変更，または最終書き込み時刻変更）

に関係なく，クローズチェックを実施します。Windows 環境のクローズチェックでは，監視対象ファイルを書き込みモードでオープンします。オープンできた場合はすぐにクローズして，監視対象ファイルをオープンしているプロセスがないと判断します。そのため，クローズチェックによるファイルのオープンからクローズまでの間，ほかのプログラムは監視対象ファイルにアクセスできません。したがって，イベントジョブや起動条件付きジョブネットで監視中のファイルをユーザープログラムから操作する場合は，ファイルアクセスの失敗を考慮し，ファイルアクセスのリトライ処理を組み込むなどの対策が必要です。

- Windows 環境で監視対象ファイルに読み取り専用属性が設定されている場合は，監視対象ファイルを書き込みモードでオープンできないため，クローズチェックが実施できず，監視条件に合致する監視対象ファイルの状態の変更を検知してもイベントを発行しません。監視対象ファイルに読み取り専用属性が設定されている場合は，統合トレースログにメッセージ KAVT2036-W が出力されるので，監視対象ファイルの読み取り専用属性を解除して，書き込みモードでオープンできるようにしてください。
- Windows のファイルシステムでは，次のファイル名が使用されます。
  - ユーザーが任意の文字列を指定したロングファイル名
  - Windows が自動的に生成するショートファイル名（8.3 形式）

  JP1/AJS3 のファイル監視ジョブも，これらのファイル名を監視対象とします。そのため，ファイル監視ジョブは，監視対象ファイル名と Windows が自動的に生成したショートファイル名が一致したファイルについてもイベントを検知します。監視対象ファイル名とショートファイル名が一致した場合，一見すると意図しないファイルについてイベントを検知したように見えます。

  ファイル監視ジョブが意図しないファイルを検知した場合は，コマンドプロンプトで次のコマンドを実行して監視対象ファイルがあるフォルダ配下のファイルのショートファイル名を出力し，監視対象ファイル名が意図しないファイルのショートファイル名と一致していないか確認してください。

```
dir 監視対象ファイルがあるフォルダのフルパス /x
```

  ショートファイル名をファイル監視ジョブの監視対象にさせないようにするには，監視対象ファイル名を次のように指定してください。

  - 監視対象ファイルに総称名を指定する場合は，ファイル名の固定文字列部分を 8 文字以上にする。

    ショートファイル名を監視対象にしない場合と監視対象にする場合の監視対象ファイル名の定義例を，次の表に示します。

表 7-6　ショートファイル名を監視対象にしない場合と監視対象にする場合の監視対象ファイル名の定義例

監視対象ファイル群		監視対象ファイル名で指定する総称名
ロングファイル名	ショートファイル名	
C:¥temp¥ABCDEFGH001.log	C:¥temp¥ABCDEF~1.log	ショートファイル名を監視対象にしない場合 C:¥temp¥ABCDEFGH*.log 固定文字列部分「ABCDEFGH」が 8 文字以上です。
C:¥temp¥ABCDEFGH002.log	C:¥temp¥ABCDEF~2.log	
C:¥temp¥ABCDEFGH003.log	C:¥temp¥ABCDEF~3.log	
C:¥temp¥ABCDXXXXXXX.log	C:¥temp¥ABCDEF~4.log	

監視対象ファイル群		監視対象ファイル名で指定する総称名
ロングファイル名	ショートファイル名	
C:¥temp¥ABCDXXXXXXX.log	C:¥temp¥ABCDEF~4.log	ショートファイル名を監視対象にする場合 C:¥temp¥ABCDEF*.log 固定文字列部分「ABCDEF」が 8 文字未満です。

- 監視対象ファイル名に完全名を指定する場合は，ショートファイル名形式のファイル名を指定しない。

  ショートファイル名形式のファイル名とは，8.3 形式のショートファイル名をロングファイル名として命名したファイル名のことです。

  補足事項

  監視対象ファイル名にショートファイル名形式のファイル名を指定する場合は，指定するファイルと同じパス上にはショートファイル名形式のファイルだけを配置するようにしてください。

- Linux の場合で，ファイル監視ジョブを実行しているホストと，NFS などのネットワークを介したファイルシステムとの接続が切断されたとき，ファイル監視ジョブが実行中のままになることがあります。この場合，正しくファイルの状態を監視できないおそれがあります。

  ファイル監視ジョブのクローズチェックにタイムアウト値を設定することで，接続の切断によってファイル監視ジョブが実行中のままになった場合に，メッセージを出力できます。メッセージを出力する場合は，環境設定パラメーターCloseCheckTimeout およびCloseCheckWarnLogInterval を設定してください。環境設定パラメーターCloseCheckTimeout の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.6.2(32) CloseCheckTimeout」を，環境設定パラメーターCloseCheckWarnLogInterval の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.6.2(33) CloseCheckWarnLogInterval」を参照してください。

7.6.3　メール受信監視ジョブの注意事項

メール受信監視ジョブの注意事項（使用する前に知っておいた方がよいこと）を次に示します。

メール受信監視ジョブを使った例を次に示します。

- システム管理者からのメールを受信したときだけ後続ジョブを実行する。
- 「Error」というタイトルのメールを受信したときだけ後続ジョブを実行する。

メール受信監視では，メールを受信したときに条件が成立します。メール受信の詳細については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド　2.6 メール受信監視ジョブの定義」を参照してください。

注意事項（UNIX 限定）

メールシステムと連携する環境から，メールシステムと連携しない環境に変更するには，jajs_config コマンドで定義キー[JP1_DEFAULT※¥JP1AOMAGENT¥mail_link]の下にあるExecMode を「U」から「N」に変更したあと，JP1/AJS3 を再起動してください。

注※

論理ホストの場合は，論理ホスト名になります。

7.6.4　ログファイル監視ジョブの注意事項

ログファイル監視ジョブの注意事項（使用する前に知っておいた方がよいこと）を次に示します。

なお，ログファイル監視ジョブの動作の概要については，「2.4.4(3) ログファイルの更新を契機に処理を実行する（ログファイル監視ジョブ）」を参照してください。

注意事項

- ログファイル監視ジョブは，JP1/Base のログファイルトラップ機能を前提に動作します。
- ログファイル監視ジョブを実行する前に，JP1/Base のログファイルトラップ管理サービスと JP1/Base のイベントサービスを起動しておいてください。

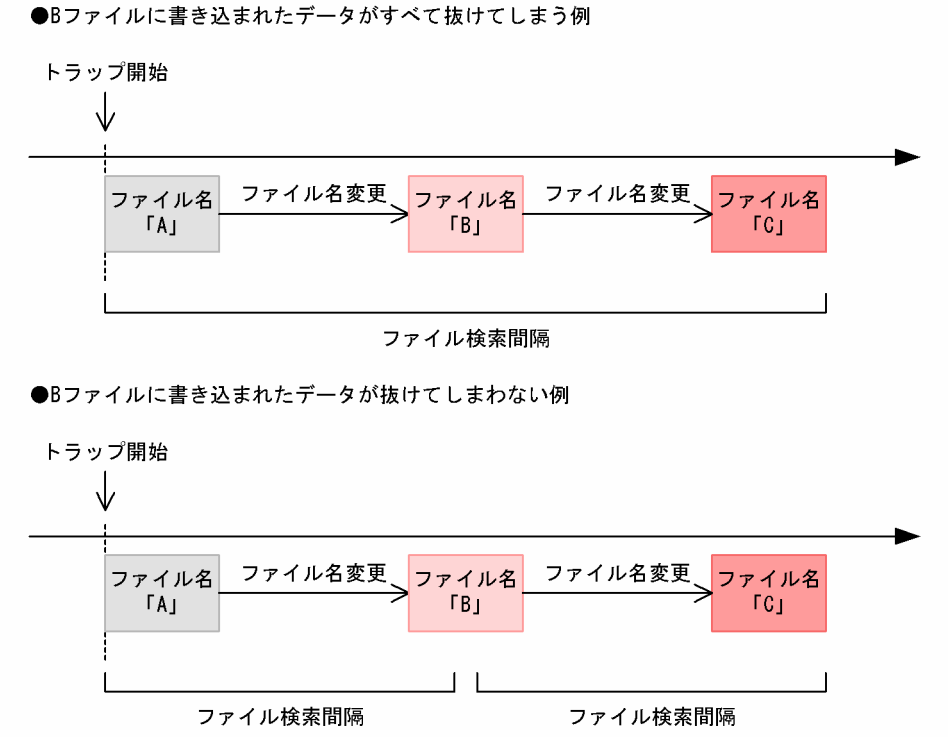
  JP1/Base のログファイルトラップ管理サービスと JP1/Base のイベントサービスが起動されていない場合，ログファイル監視ジョブは「実行中」状態になりますが，実際には監視されません。実際の監視は，JP1/Base イベントサービスと JP1/Base のログファイルトラップ管理サービスが起動されてから開始されます。

  JP1/Base のログファイルトラップ機能の詳細については，マニュアル「JP1/Base 運用ガイド」を参照してください。
- ログファイル監視ジョブの実行中に，JP1/Base のログファイルトラップ管理サービスまたは JP1/Base のイベントサービスを停止させたり，JP1/Base の`jevlogstop` コマンドを実行してログファイル監視ジョブが起動しているログファイルトラップを終了させたりしないでください。このような操作をした場合は，監視対象のログデータが書き込まれてもイベントを検知しないことがあります。
- ログファイル監視ジョブの実行時には，JP1/Base のログファイルトラップ機能を使用するために必要なメモリー，ディスク，およびシステム資源が，ログファイル監視ジョブ単位で必要になります。

  JP1/Base のリソースの見積もりについては，マニュアル「JP1/Base 運用ガイド」を参照してください。
- ジョブ実行時にマウントされたり，アンマウントされたりする環境にあるログファイルを監視対象にしないでください。このようなファイルを監視対象にした場合，監視処理が正常に動作しなくなったり，ファイルが新規に作成されたと誤認してログファイルを先頭から読み込んでしまったりすることがあります。
- ログファイルの出力形式で「SEQ2」を選択した場合，ファイル検索間隔の間にファイル名の変更が 2 回以上行われると取得メッセージが消失してしまうので，ファイル検索間隔の指定には注意してください。

  SEQ2 で監視中に名称変更したファイルの編集を実施したい場合，新たなファイルが作成されたあと，指定したファイル検索間隔の時間が経過してから行ってください。

図 7-18 「SEQ2」を選択した場合に取得メッセージが消失する例としない例



- ログファイル監視ジョブは，次に示すログファイルの出力形式を監視できません。
  - ・HTRACE
  - ・SEQ3
  - ・UPD
- ログファイル監視ジョブで監視するログファイルの文字コードは，JP1/AJS3 起動時の文字コードと同じ文字コードにしてください。
- ファイル出力形式を「SEQ2」と定義しているジョブを，JP1/AJS2 - View 06-00～06-71 の詳細設定で参照した場合，ファイル出力形式には何も表示されません。また，このときに詳細設定画面の［OK］ボタンをクリックすると，「SEQ2」と定義している内容が「SEQ」に上書きされるため注意してください。
- 同じログファイル（syslog を含みます）を参照するログファイル監視ジョブを複数個定義すると，ログファイルアクセス回数が増えて負荷が大きくなります。この場合，JP1/Base のログファイルトラップ管理サービスを使用して一つのログファイルトラップの動作定義をしておきます。そして，JP1/Base の JP1 イベント変換機能を使用して，ログファイルの更新内容を JP1 イベント化し，JP1 イベント化した JP1 イベントを，JP1 イベント受信監視ジョブで監視することを推奨します。
- Windows のネットワークドライブで参照できるログファイルや，UNIX の NFS などのネットワークを介したファイルシステムへのマウントで参照できるログファイルを監視対象にしないでください。これらのファイルを監視対象にした場合，ネットワークの切断によってログファイルを監視できなくなったり，他システムで使用中かどうか判断できないためログファイルを監視できなくなったりします。

- 次に示すファイルを監視対象にしないでください。これらのファイルを監視対象にした場合，監視処理が正常に動作しなくなることがあります。

  ・スペシャルファイル，デバイスファイル

  ・ログファイル中の 1 行の終了文字以外でバイナリーデータを含むレコードがあるファイル

  ・ファイル名が特定できないファイル

- スケジューラーログを監視対象にした場合，監視中に「ajsalter -c COPY」を実行すると監視処理が正常に動作しなくなります。

- ログファイル監視ジョブを開始したときに，監視対象のログファイルが見つけられなかった場合，監視対象のファイルが見つかるまで検索されます。そのため，監視開始から，ログファイルを見つけるまでの間に書き込まれたログデータについては，監視対象から外れている場合があるので注意してください。

- ログファイル監視ジョブ実行時に，オプション指定によって，ファイルがなかった場合にログファイル監視ジョブを異常終了とすることもできます。この場合は，監視開始時にファイルがないと，検索しないでログファイル監視ジョブを終了します。

- 正規表現で指定する項目で，拡張正規表現を使用したい場合は，JP1/Base の設定を行ってください。

- 正規表現で指定する項目では，指定した文字列は部分一致で条件一致となります。完全一致させたい場合は，完全名を明示的に指定した正規表現を使用してください。Windows の正規表現については，マニュアル「JP1/Base 運用ガイド」を参照してください。UNIX の正規表現については，UNIX のドキュメントを参照してください。

  JP1/Base が前提としている Linux では，ログファイルトラップ機能で使用する正規表現で日本語は処理できません。そのため，JP1/AJS3 のログファイル監視ジョブでも「.」（ピリオド）二つが日本語 1 文字に一致するなどの動作になります。正規表現については，マニュアル「JP1/Base 運用ガイド」を参照してください。

- トラップするデータに，OR 条件で複数のログデータを指定したログファイル監視ジョブを起動条件で実行した場合で，ログファイルに書き込まれたログデータに，指定した複数のログデータが含まれているとき，一つのログデータに対して複数回，監視条件が成立します。このような場合，一つ目のログデータを指定したあとのログデータでは，指定したいログデータを OR 条件で指定し，前に指定したログデータすべてを AND 条件と Not 条件を組み合わせて指定するようにしてください。

  例えば，「Error」「Warning」「Information」「Notice」を監視する場合，次に示すように指定します。

  ```
  lftpd="Error";
  lftpd="Warning":!"Error";
  lftpd="Information":!"Error":!"Warning";
  lftpd="Notice":!"Error":!"Warning":!"Information";
  ```

- ログファイル名を相対パスで指定した場合，次に示すディレクトリが通常，ログファイルのカレントディレクトリとなります。

  Windows の場合

システムフォルダ¥system32

UNIX の場合

jajs_spmd コマンドの実行ディレクトリ

- Windows では，多数のログファイル監視ジョブを同時に実行すると，JP1/Base のエラーメッセージを出力し，ログファイル監視ジョブが異常終了することがあります。このような場合，マニュアル「JP1/Base 運用ガイド」を参照して，JP1/Base のエラーメッセージの対処方法に従って対処してください。
- FQDN 形式のイベントサーバ名を持つイベントサーバが稼働するホストでログファイル監視ジョブを実行する場合は，実行エージェントの JP1/AJS3 に FQDN 形式のイベントサーバ名オプション（環境設定パラメーターEventServerName）を設定し，ログファイル監視ジョブが使用するイベントサーバ名（FQDN 形式）を指定してください。設定手順については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.3.17 DNS を使ったシステムでのイベントサーバ名の設定」（Windows の場合），またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.3.16 DNS を使ったシステムでのイベントサーバ名の設定」（UNIX の場合）を参照してください。

7.6.5　Windows イベントログ監視ジョブの注意事項

Windows イベントログ監視ジョブの注意事項（使用する前に知っておいた方がよいこと）を次に示します。

なお，Windows イベントログ監視ジョブの動作の概要については，「2.4.4(4) Windows イベントログの受信を契機に処理を実行する（Windows イベントログ監視ジョブ）」を参照してください。

- Windows イベントログ監視ジョブを使用するには，JP1/Base の動作定義ファイルにイベントログトラップ機能の動作を設定する必要があります。
- Windows イベントログ監視ジョブを実行する前に，JP1/Base のイベントログトラップサービスと JP1/Base のイベントサービスを起動しておいてください。

  JP1/Base のイベントログトラップサービスと JP1/Base のイベントサービスが起動されていない場合，Windows イベントログ監視ジョブは「実行中」状態のまま，Windows イベントログを受信してもジョブが終了されません。
- 監視できる Windows イベントは，JP1/Base のイベントログトラップ機能の仕様に従います。

  JP1/Base のイベントログトラップ機能の詳細および動作定義については，マニュアル「JP1/Base 運用ガイド」を参照してください。
- Windows イベントが分類できなかった場合，Windows のイベントビューアには「なし」と表示されますが，JP1/Base のイベントログトラップ機能を使って JP1 イベント化すると「None」として扱われます。そのため，JP1/AJS3 の Windows イベントログ監視ジョブの定義項目「分類」では，「なし」ではなく「None」と定義してください。「なし」という文字列を指定しても条件は成立しませんので注意してください。

- JP1/Base のイベントログトラップ機能は，定義内容を変えないままで運用すると，「エラー」と「警告」だけが監視されるようになっています。「詳細」，「情報」，「重大」，「失敗の監査」，および「成功の監査」を監視したい場合は，JP1/Base のイベントログトラップ機能の動作定義ファイルに，監視したいイベントについての定義を追加してください。
- Windows イベントログ監視ジョブの定義項目「分類」に指定した監視条件と Windows イベントの分類が完全に一致した場合に，監視条件が成立します。また，指定できる監視条件は 255 バイトまでですが，Windows イベントログには 255 バイトよりも長い情報が登録される場合があります。この場合は，Windows イベントログ監視ジョブの定義項目「分類」で指定した文字列と，255 バイトまでの Windows イベントの分類の情報とを比較し，一致した場合に条件が成立します。
- 正規表現で指定する項目では，指定した文字列は部分一致で条件一致となります。完全一致させたい場合は，完全名を明示的に指定した正規表現を使用してください。Windows の正規表現については，マニュアル「JP1/Base 運用ガイド」を参照してください。
- Windows イベントログ監視ジョブの定義項目「説明」に改行を含む文字列を指定した場合，イベントを検知できないことがあります。改行を含む Windows イベントの説明を監視する場合は，正規表現を使用して改行コードを「.*¥n」で指定してください。Windows の正規表現については，マニュアル「JP1/Base 運用ガイド」を参照してください。

  （例）次の文字列を監視する場合

  ```
  「TEST1を開始します
   TEST2を開始します」
  ```

  次のように指定します。

  ```
  「TEST1を開始します.*¥nTEST2を開始します」
  ```

- ジョブの実行先ホストの OS や JP1/AJS3 のバージョンによっては，次のエラーになります。

  ジョブの実行先ホストの OS が UNIX の場合

  ジョブの状態は「起動失敗」になり，メッセージ KAVT0567-E が出力されます。

  ジョブの実行先ホストの OS が Windows Server 2003 の場合

  Windows Server 2003 以外でだけ指定できる監視条件を指定している場合，エラーになります。

  Windows Server 2003 以外でだけ指定できる監視条件は，次のとおりです。

  ログ種別：「任意のログ種別」

  イベントの種別：「詳細」および「重大」

  ・ジョブの実行先ホストの JP1/AJS3 がバージョン 10-00 以降のとき

  「詳細」あるいは「重大」だけ，または「詳細」と「重大」の両方だけが指定されている場合は，ジョブの状態は「起動失敗」になり，メッセージ KAVT0591-E が出力されます。

  「詳細」または「重大」が指定されているが，それ以外の項目も指定している場合は，ジョブは実行され，エラーになりません。

  ・ジョブの実行先ホストの JP1/AJS3 がバージョン 10-00 より前のとき

  ジョブは異常終了して，メッセージ KAVT1013-E が出力されます。

7.6.6　実行間隔制御ジョブの注意事項

実行間隔制御ジョブの注意事項（使用する前に知っておいた方がよいこと）について説明します。

実行間隔制御ジョブを使った例を次に示します。

- ジョブネットを実行登録して，その時刻から 30 分後に後続ジョブを実行する。

  実行登録方法については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　4. 業務の実行」を参照してください。

実行間隔制御の定義では，待ち時間を 1～1,440（単位：分）で指定します。

注意事項

- 実行間隔制御ジョブで指定する待ち時間は，ジョブの実行から終了までの時間ではなく，実行間隔制御プロセスでの待ち時間を意味します。通信の状態などによって，指定した待ち時間と誤差が生じる場合があります。
- 実行間隔制御ジョブが実行中の状態で，実行間隔制御ジョブを実行したエージェントに対応するホストの JP1/AJS3 サービスを再起動した場合は，再起動後から指定した待ち時間をカウント（監視）し，満了後に監視条件が成立します。

  例えば，待ち時間に「60」（単位：分）を定義した実行間隔制御ジョブが「実行中」状態のときに JP1/AJS3 サービスを再起動した場合は，再起動するまでに経過した時間に関係なく，再起動してから 60 分後に監視条件が成立します。

  次の場合は，JP1/AJS3 サービスの再起動後もジョブの状態は「実行中」となります。そのため，監視の途中で JP1/AJS3 サービスを再起動すると，実行間隔制御ジョブを実行してから監視条件の成立までに掛かる時間が，ジョブに指定した待ち時間よりも長くなるため，注意してください。

  ・起動条件に定義した実行間隔制御ジョブを実行している場合

  ・イベントジョブ実行継続オプションを有効にしたマネージャーホストから実行間隔制御ジョブを実行している場合

  イベントジョブ実行継続オプションの詳細については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　8.2.1 JP1/AJS3 のサービスが停止してもイベントジョブの実行を継続させる」を参照してください。
- 次のような状態で，起動条件に実行間隔制御ジョブを定義して，「開始直後に満了」に「する」を指定した起動条件付きジョブネットを実行登録した場合は，起動条件が成立するまで時間が掛かることがあります。

  ・起動条件付きジョブネットや，イベントジョブを定義したジョブネットを，一度に大量に実行登録している状態

  ・大量にイベントを検知している状態

7.6.7　引き継ぎ情報定義時の注意事項

イベントジョブの引き継ぎ情報定義時の注意事項について説明します。

- 引き継ぎ情報は，イベントジョブを実行したホストと後続のジョブを実行するホストが異なっても引き継げます。このとき，ホスト間の文字コードが異なる場合は引き継ぎ情報を後続のホストの文字コードに変換します。ただし，引き継ぎ情報に含まれる文字がサポート範囲外の場合，JP1/AJS3 が誤動作するおそれがあります。
- JP1/AJS3 で使用できる文字の範囲については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 2.4.2 システムで使用する言語種別と文字コードについて検討する」を参照してください。なお，サポート範囲外の文字はスタンドアロン構成であっても使用できません。引き継ぎ情報の中に含めないようにしてください。
- 複数のイベントジョブを，一つの後続ジョブに関連づけた場合，すべてイベントジョブで受信した情報を後続ジョブに引き継げます。ただし，複数のイベントジョブで，同じマクロ変数名を定義してある場合は，受信情報が上書きされます。
- 一つのイベントジョブの引き継ぎ情報に，同じ名称のマクロ変数が複数定義された場合，最初に定義された情報が引き継がれます。

  ＜JP1 イベント受信監視イベントジョブでの例＞

  引き継ぎ情報に次の二つを定義します。

  - ?AJS2111?:EVID（マクロ変数「?AJS2111?」にイベント ID を引き継ぐように指定）
  - ?AJS2111?:EVMSG（マクロ変数「?AJS2111?」にメッセージ情報を引き継ぐように指定）

  この場合，「?AJS2111?」で引き継がれる情報はイベント ID になります。
- コマンドラインには，エスケープシーケンスを含むデータを引き継がないでください。また，空白を含んだデータを引き継いだ場合，意図しない動作をすることがあるため，マクロ変数を「"（ダブルクォーテーションマーク）」で囲んで定義してください。
- マクロ変数を後続ジョブのコマンドラインに指定する場合には，引き継ぐ情報に空白または「'（シングルクォーテーションマーク）」が含まれていると，情報を正しく引き継げません。また，マクロ変数はイベントジョブの後続ジョブを実行するエージェントホスト上で，コマンドラインに引き継いだ情報を展開した形で実行されます。引き継ぎ情報を定義する場合には，実行時にコマンドの引数として扱える形の情報を引き継ぐ場合だけ定義してください。
- イベントジョブから引き継ぐ情報がない場合，またはイベントジョブが実行されなかった場合は，後続ジョブで定義したマクロ変数には情報が引き継がれません。この場合，ジョブ実行時にマクロ変数名を「?AJS2111?」と定義すると，文字列「?AJS2111?」が渡されます。
- イベントジョブの受信情報を引き継ぎ情報として，標準ジョブやアクションジョブのパラメーターに引き継ぐ場合で，引き継ぐ情報に「”（ダブルクォーテーションマーク）」が含まれるとき，「¥」を前に付けて指定する必要があります。「¥」を付けない場合，受信情報を引き継いでも「”（ダブルクォーテーションマーク）」が無視されるなどの現象が発生します。

  これを防止するためには，「”（ダブルクォーテーションマーク）」を含むデータを指定したとおりに引き継ぎ情報として引き継がせるオプションを設定してください。

  設定については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド 6.3.4 「"（ダブルクォーテーションマーク）」を含むデータをイベントジョブの引き継ぎ情報に使用するための設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド

15.3.4 「"（ダブルクォーテーションマーク）」を含むデータをイベントジョブの引き継ぎ情報に使用するための設定」（UNIX の場合）を参照してください。

- 引き継ぎ情報をジョブのコマンドラインで使う場合は，引き継ぐ情報に「"（ダブルクォーテーションマーク）」などの文字列が入っていても，変換などは実行されないでそのまま後続ジョブに引き継がれます。そのため，各 OS のコマンドラインの制約でジョブが正しく実行されないことがあります。特殊な文字が入っている引き継ぎ情報を使う場合には，コマンドラインに直接展開しないで，環境変数に引き継ぐようにしてください。
- 起動条件に定義したイベントジョブにマクロ変数を定義した場合は，起動条件の成立によって起動されたジョブネット全体に情報が引き継がれます。
- マクロ変数名と引き継ぎ情報のサイズの合計が 4,096 バイトを超えないようにしてください。特に，起動条件を AND 条件で定義している場合は，起動条件内に定義したすべてのイベントジョブのマクロ変数名および引き継ぎ情報をマージします。このため，マクロ変数名と引き継ぎ情報のサイズの合計が 4,096 バイトを超えないように注意する必要があります。
- JP1/AJS3 09-50 以降にマクロ変数を指定できるようになった定義項目に，マクロ変数を指定しているジョブがある場合，次の点に注意する必要があります。
  - JP1/AJS3 - Manager のバージョンが 09-50 より前の環境にコピーして使用した場合，マクロ変数は展開されません。
  - 引き継ぎ情報設定ジョブを，JP1/AJS3 - Manager のバージョンが 09-50 より前の環境にコピーして使用した場合，引き継ぎ情報設定ジョブの実行時にエラーになります。

7.6.8 イベントジョブを実行したままサービスを再起動する場合の注意事項

イベントジョブ（起動条件に設定しているものを含む）を実行したまま，次のような操作を行った場合，スケジュール制御とイベント・アクション制御間では，実行中のイベントジョブについて互いの管理状態の整合性を取るための通信，および関連の処理をします。そのため，イベントジョブが大量に「実行中」状態だった場合は，それらの処理量が多くなり，高い負荷が掛かった状態になります。

高負荷状態になるような操作

- スケジューラーサービスを停止し，ウォームスタートまたはホットスタートする。
- マネージャーホストの JP1/AJS3 サービスを停止し，ウォームスタートまたはホットスタートする。
- エージェントホストの JP1/AJS3 サービスを停止し，再起動する。
- `jajs_maintain` コマンドを実行する。

高負荷状態になるのを回避するには，次のように操作してください。

回避方法

- 上記の操作を行う前にイベントジョブを使用したジョブネットを強制終了し，上記操作後に実行登録し直す。

- 上記の操作を行う前にイベントジョブを強制終了し，上記操作後に再実行する。
- 上記の操作後は，30 分～1 時間程度の時間をおいてからジョブを操作する。
- 上記の操作後は，30 分～1 時間程度の時間をおいてから監視対象イベントが発生するようにする。

なお，高負荷状態になると，次のような現象が発生することがあります。

高負荷状態によって発生する現象

1. 上記の操作直後に実行登録したイベントジョブ（起動条件に設定しているものを含む）が「実行中」状態になるまでに長時間掛かる。※1
2. 「実行中」状態のイベントジョブや，「監視中」状態の起動条件付きジョブネットを強制終了しても，終了するまでに長時間掛かる。※1
3. 「実行中」状態のイベントジョブの状態を変更しても，状態が変わるまでに長時間掛かる。※1
4. 監視条件が成立しても，イベントを検知するまでに長時間掛かる。※1
5. 上記の操作直後に実行登録したイベントジョブ（起動条件に設定しているものを含む）が「キューイング」状態のままとなる。※2
6. 「実行中」状態のイベントジョブや，「監視中」状態の起動条件付きジョブネットを強制終了しても，終了しない。※2
7. 監視条件が成立しても，イベントを検知しない。※2

注※1

現象 1～4 は，イベント・アクション制御の制限値に近い状態で高負荷状態になる操作をすると発生するおそれがあります。イベント・アクション制御の制限値については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 付録 B.8 イベント・アクション制御の制限値」を参照してください。

注※2

現象 5～7 は，イベント・アクション制御の制限値を超えた状態で高負荷状態になる操作をすると発生するおそれがあります。イベント・アクション制御の制限値については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 付録 B.8 イベント・アクション制御の制限値」を参照してください。

イベントジョブを大量に実行したまま高負荷状態になる操作をすると，スケジューラー制御とイベント・アクション制御の間で大量の通信が発生し，通信エラー時のリトライに備えて生成・管理する未通知情報の数が増加します。保持できる未通知情報には上限が設けてあります。これは，未通知情報を上限なく保持してしまうと，その処理による高負荷が生じてほかの処理ができなくなり，ジョブの実行やイベントの検知の遅れが発生するためです。保持できる未通知情報の上限を超過した場合は，古いものから破棄されます。結果として，破棄された未通知情報の内容によって，現象 5～7 のどれかが発生します。

なお，JP1/AJS3 では未通知情報の数の上限は公開していません。高負荷状態になる操作によって発生する未通知情報の数からイベント・アクション制御の制限値を算出しています。制限値を超えないように運用してください。

上記の現象が発生した場合は，次の回復策を実施してください。

回復方法

現象 1～4 の場合

処理が終わるまで待ってください。ジョブ数にもよりますが，目安としては 30 分～1 時間程度掛かります。

現象 5 の場合

該当のイベントジョブまたはジョブネットを強制終了し，実行登録し直してください。

現象 6 の場合

イベントジョブの場合は，状態変更して終了させてください。起動条件付きジョブネットの場合は，再度，強制終了してください。

現象 7 の場合

該当のイベントジョブまたは起動条件付きジョブネットを強制終了し，実行登録し直してください。そのあと，再度イベントを発生させてください。

7.6.9 JP1/AJS3 が発行するイベントやメッセージを監視する

JP1/AJS3 のメッセージを監視する場合，メッセージの最後に付加情報が追加されるおそれがあるため，メッセージ ID を監視するか，またはメッセージの前文を監視するようにしてください。メッセージの全文で監視すると，メッセージが検出できない場合があります。

JP1/AJS3 のイベントジョブに，JP1/AJS3 が発行する JP1 イベントや Windows イベント，ログファイルに出力されるメッセージを監視するように指定しても，JP1/AJS3 の状態によってはイベントとして正しく受信できなかったり，監視条件の設定によってはイベントジョブとして無限に実行してしまったりする場合があります。具体的には，次のような現象が発生する場合があります。

- イベントジョブは通常のジョブと同様に，ジョブネット中のジョブとして実行されます。JP1/AJS3 が起動され，ジョブネットが起動されて初めてイベントの監視が始まります。ジョブネット開始イベントは，最初のジョブが起動したあとに発行されるため，イベントジョブを実行している JP1/AJS3 のジョブネット開始イベントは監視できません。
- JP1/AJS3 にトラブルが発生し，イベントジョブが定義されているジョブネットが異常終了した場合，イベントジョブ自体も異常終了するので，イベントは監視できなくなります。
- イベントジョブも，ジョブの延長で JP1 イベントや Windows イベントを発行したり，ログファイルにメッセージを出力したりします。イベントジョブが発行した JP1 イベント，Windows イベント，およびログファイルに対するメッセージを，そのイベントジョブ自身が監視することになるため，監視が無限に続いてしまいます。

上記の対処方法として，JP1/AJS3 が発行するイベントやログファイル中のメッセージを契機にアクションを実行する場合は，JP1/IM の自動アクション機能か，またはイベントジョブを実行しているマネージャーとは別の JP1/AJS3 のマネージャーを使用してください。JP1/AJS3 が発行するイベントやログファ

イル中のメッセージを契機にコマンドなどを実行する場合については，JP1/AJS3 のマネージャーが複数ある場合（論理ホストを幾つかに分けて JP1/AJS3 を多重起動している場合を含む）と，JP1/AJS3 のマネージャーが一つだけの場合（JP1/AJS3 を多重起動していない場合）とに分けて次に説明します。また，イベントやメッセージの監視に関する注意事項も説明します。

なお，この節で説明する内容は，JP1/AJS3 自身が発行する JP1 イベント，Windows イベント，ログファイル・syslog などに出力するメッセージを監視する場合を対象としています。JP1 イベント送信ジョブを使用する場合や，ユーザージョブとして JP1 イベントを送信する場合は，対象外です。

(1) JP1/AJS3 のマネージャーが複数ある場合（論理ホストを幾つかに分けて JP1/AJS3 を多重起動している場合を含む）

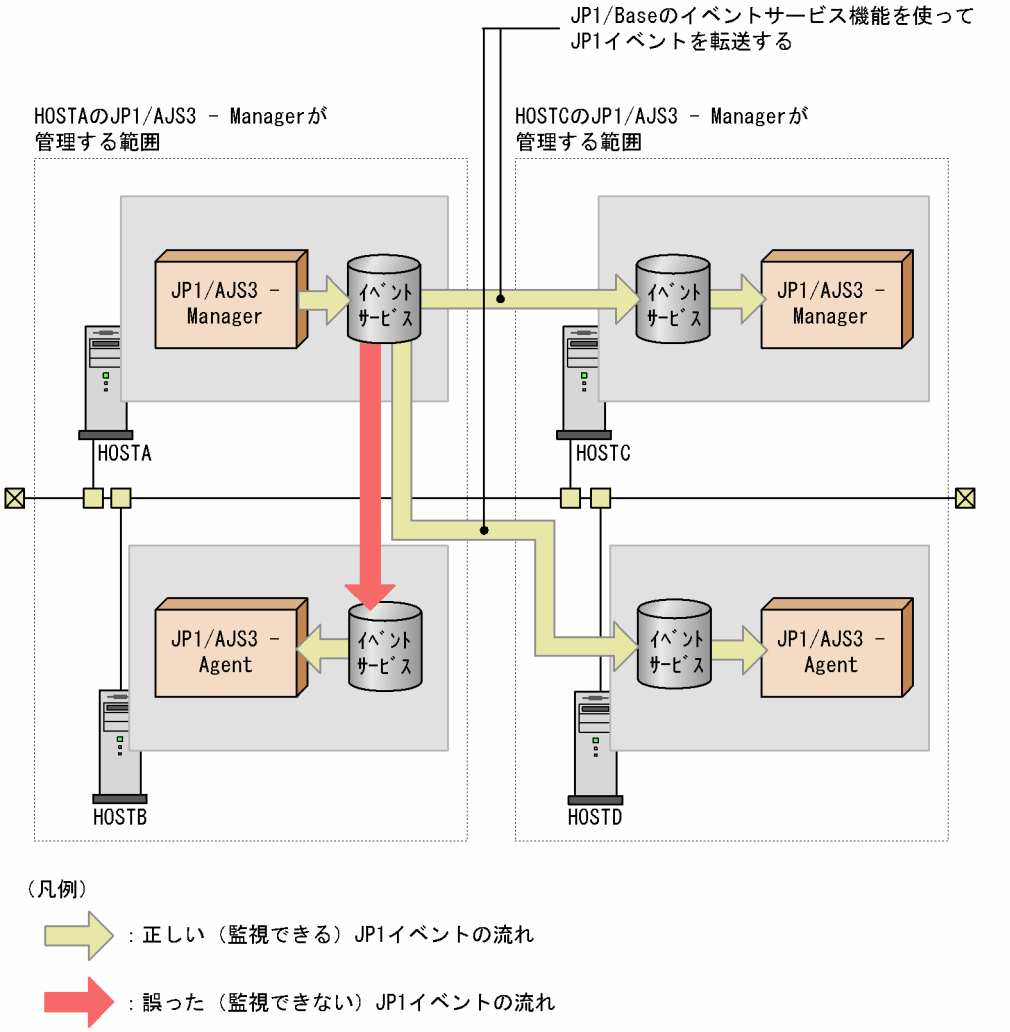
JP1/AJS3 のマネージャーが複数ある場合，または論理ホストを幾つかに分けて JP1/AJS3 を多重起動している場合に，ほかの論理ホスト上の JP1/AJS3 が発行するイベントやログファイル中のメッセージを監視するときの対処方法を説明します。なお，このとき，JP1/IM は不要です。

(a) ほかの論理ホスト上の JP1/AJS3 が発行する JP1 イベントを監視するとき，または JP1/AJS3 のマネージャーが複数あるとき

標準では，他ホスト（論理ホスト）上の JP1/AJS3 が発行する JP1 イベントは監視できません。ただし，監視対象の JP1/AJS3 が発行する JP1 イベントを，JP1/Base のイベントサービス機能を使って転送すれば監視できます。このとき，JP1 イベントを，その JP1 イベントを発行した JP1/AJS3 が管理するホストとは別のホスト（論理ホスト）上のイベントサービスに転送してください。その JP1 イベントが登録された JP1/AJS3 が管理するホスト上のイベントサービスに転送しても監視できません。

JP1 イベントの流れの例を次の図に示します。

図 7-19　JP1 イベントの流れ



上記の図の場合，HOSTA の JP1/AJS3 が発行する JP1 イベントを監視したい場合は，JP1 イベントを HOSTC または HOSTD のイベントサービスに転送してください。HOSTB のイベントサービスに転送しても監視できません。

(b)　ほかの論理ホスト上の JP1/AJS3 が発行する Windows イベントを監視するとき

Windows イベントは，JP1/Base のイベントログトラップ機能が動作する論理ホスト上だけで監視できます。

JP1/AJS3 が発行する Windows イベントを監視する場合は，論理ホストを分け，監視したい JP1/AJS3 とは別のマネージャーから監視してください。この場合，監視される JP1/AJS3 からは Windows イベントログ監視ジョブは使用できません。

(c)　ほかの論理ホスト上の JP1/AJS3 が出力するログファイル中のメッセージを監視するとき

監視対象のログファイルを，ほかの論理ホストの JP1/AJS3 から参照してください。

(2) JP1/AJS3 のマネージャーが一つだけの場合（JP1/AJS3 を多重起動していない場合）

JP1/AJS3 のマネージャーが一つだけの場合に，JP1/AJS3 が発行するイベントやログファイル中のメッセージを監視するときの対処方法を説明します。なお，このとき，JP1/IM が必要です。

(a) JP1/AJS3 が発行する JP1 イベントを監視するとき

JP1/IM の自動アクション機能を使用してください。

(b) JP1/AJS3 が発行する Windows イベントを監視するとき

JP1/Base の Windows イベントトラップ機能，および JP1/IM の自動アクション機能を使用してください。

(c) JP1/AJS3 が出力するログファイル中のメッセージを監視するとき

JP1/Base のログファイルトラップ機能，および JP1/IM の自動アクション機能を使用してください。

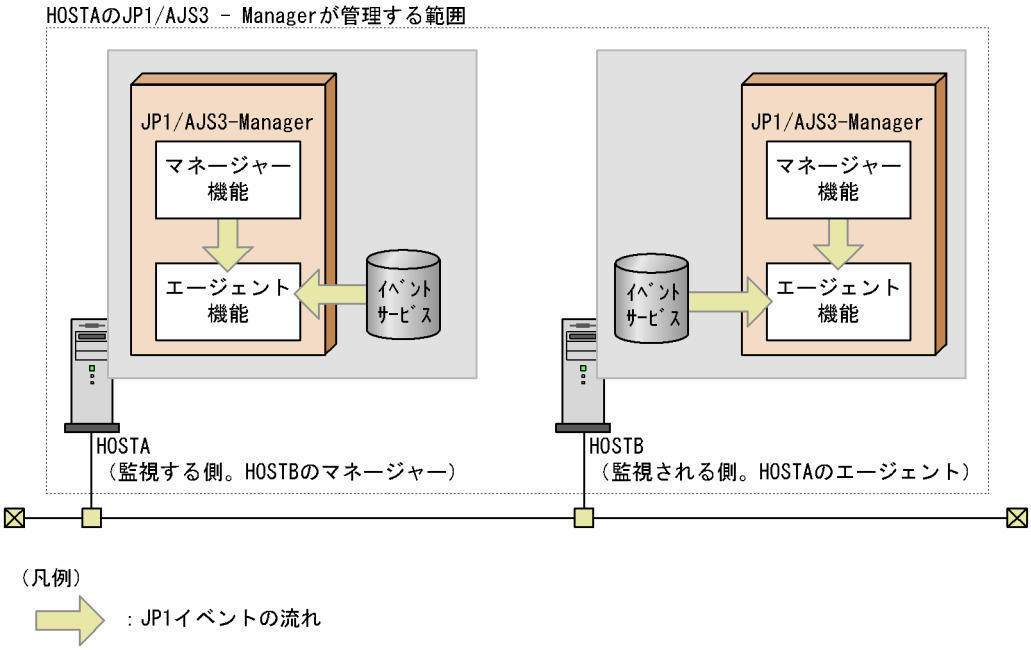
(3) イベントやメッセージの監視に関する注意事項

イベントやメッセージの監視に関する注意事項を次に示します。

(a) 2 台の JP1/AJS3 - Manager ホストでイベントジョブを用いてホストを監視する場合の制限事項

次に示す図のように，ある JP1/AJS3 - Manager ホスト（HOSTA）を HOSTB のマネージャー（監視する側）として，別の JP1/AJS3 - Manager ホスト（HOSTB）を HOSTA のエージェント（監視される側）として設定し，イベントジョブを用いて HOSTA から HOSTB を監視する場合の制限事項を次に示します。

図 7-20　2 台の JP1/AJS3 - Manager ホストでのイベントジョブを用いたホストの監視 (1)



制限事項

- HOSTB の JP1/AJS3 が起動していない場合，HOSTB では，イベントジョブを用いた監視はできません。

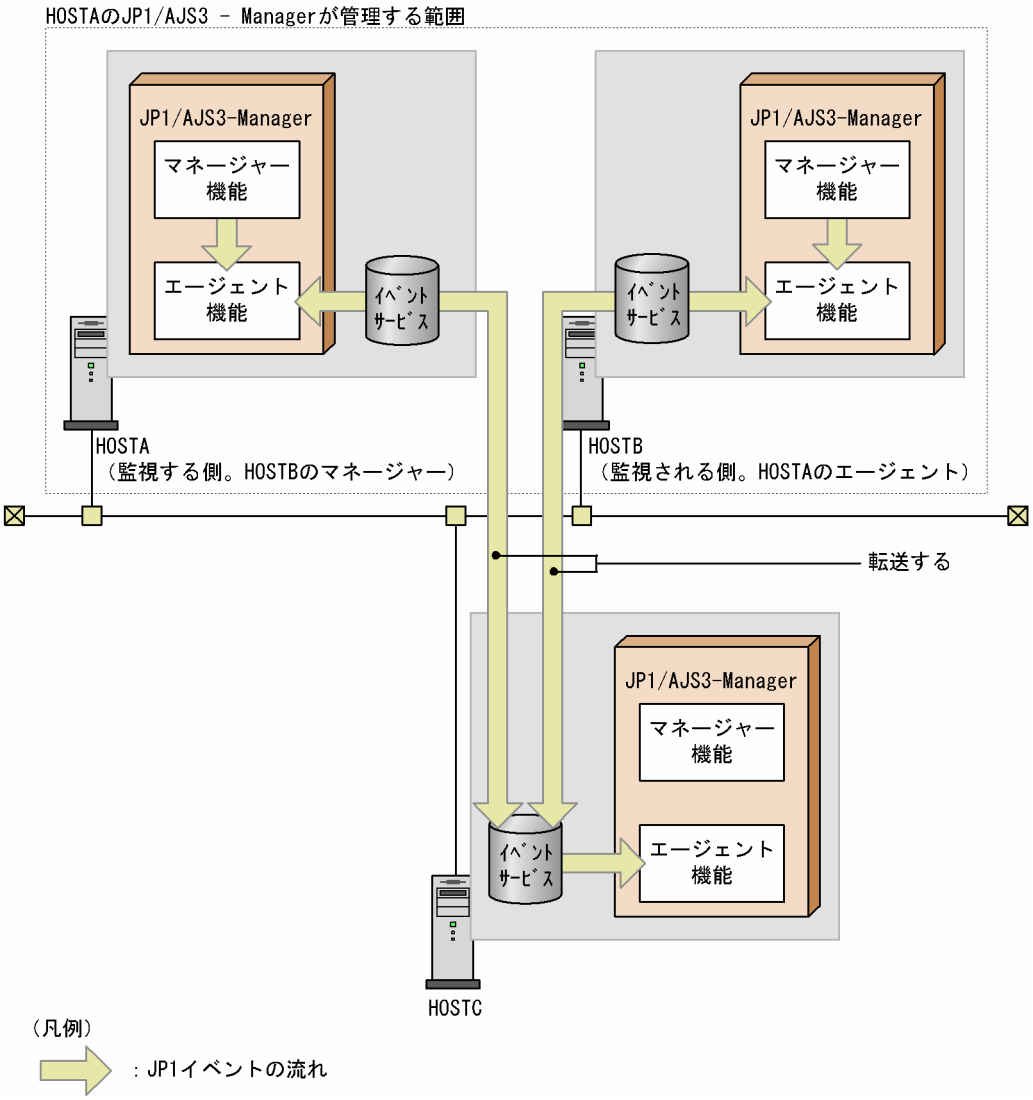
  マネージャーにはエージェントの機能も含まれています。そのため，HOSTA が監視している HOSTB のエージェントは，HOSTB のマネージャーの起動・停止と連動して起動・停止します。

- HOSTB の JP1/AJS3 が異常終了した場合，HOSTA に情報を通知できません（HOSTA から HOSTB の監視はできません）。

  マネージャーにはエージェントの機能も含まれています。そのため，HOSTA が監視している HOSTB のエージェントは，HOSTB のマネージャーの異常終了と連動して停止します。

この構成でホストを監視したい場合は，次の図に示すように，イベントを，監視する側および監視される側の JP1/AJS3 - Manager ホストとは別の JP1/AJS3 - Manager ホストに転送してください。

図 7-21　2 台の JP1/AJS3 - Manager ホストでのイベントジョブを用いたホストの監視 (2)

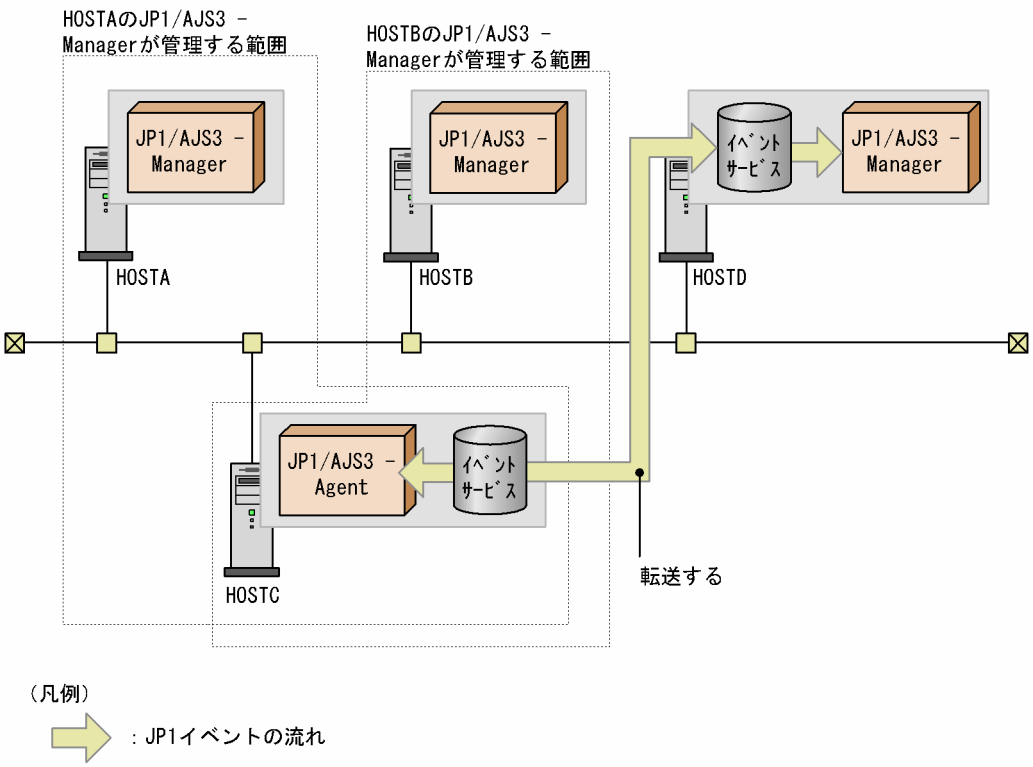


(b)　複数台の JP1/AJS3 - Manager ホストが 1 台の JP1/AJS3 - Agent ホストをエージェントとして設定している場合のイベントやメッセージの監視

複数台の JP1/AJS3 - Manager ホストが 1 台の JP1/AJS3 - Agent ホストをエージェントとして設定している場合，エージェントの JP1/AJS3 - Agent は同一のログファイル（エージェントが出力している，JP1/AJS3 のログファイル）の情報を監視します。そのため，マネージャーが異なっていても，該当するエージェントを管理するマネージャーからは正しく監視できません。

この構成でイベントやメッセージを監視したい場合は，次の図に示すように，イベントやメッセージを，JP1/AJS3 - Agent ホストを管理する JP1/AJS3 - Manager ホストとは別の JP1/AJS3 - Manager ホストに転送してください。

図 7-22　複数台の JP1/AJS3 - Manager ホストが 1 台の JP1/AJS3 - Agent ホストをエージェントとして設定している場合のイベントやメッセージの監視



7.7　アクションジョブ使用時の注意事項

アクションジョブ使用時の注意事項（使用する前に知っておいた方がよいこと）について説明します。

注意事項

- アクションジョブには「実行時のユーザー」を指定できません。詳細については各アクションジョブの注意事項を参照してください。
- アクションジョブは，ジョブ実行多重度の影響を受けます。そのため，ジョブ実行多重度に設定した値以上のジョブをエージェントホストで同時に実行しようとした場合，ジョブは「実行待ち」状態となります。ジョブ実行多重度については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 2.5.4 ジョブ実行多重度の検討」を参照してください。

7.7.1　JP1 イベント送信ジョブの注意事項

JP1 イベント送信ジョブの注意事項（使用する前に知っておいた方がよいこと）について説明します。送信する JP1 イベントには，重大度，イベント ID，および送信先ホスト名を指定します。JP1 イベントを送信したあと，送信した JP1 イベントが送信先ホストのイベントサーバに到達したかどうかを確認することもできます。送信先ホストには，JP1/Base が必要です。

重要

JP1 イベント送信ジョブを実行する前に，送信元と送信先で，JP1/Base イベントサービスを起動しておいてください。

JP1 イベント送信ジョブを使ったジョブネットの例を次に示します。

- あるジョブが異常終了したときに JP1 イベントを送信する。別のジョブネットに，その JP1 イベントを受信したらジョブネットを起動するように起動条件を定義しておき，ジョブが異常終了したときだけジョブネットを起動する。
- ジョブが終了したら JP1 イベントを JP1/IM - Manager がインストールされたサーバに送信するように定義しておき，送信した JP1 イベントを JP1/IM - View で確認する。

注意事項

- JP1 イベントの到達確認は，送信したイベントが送信先イベントサーバへ到達したことを確認できるまで待つ機能であり，送信失敗時にリトライを行う機能ではありません。そのため，送信先のイベントサービスが停止している状態でイベント送信ジョブを実行した場合，「確認間隔」，「確認回数」の指定有無に関係なく，すぐに終了します。
- JP1 イベントの到達確認をしない場合は，次に示すイベントサーバをイベント送信先ホスト名に指定してもエラーにはなりません。

  ・JP1/Base のイベントサーバとして定義されていないイベントサーバ

・起動していないイベントサーバ

・ネットワーク障害などで，JP1 イベントが届かないイベントサーバ

- イベント送信先ホスト名に自ホスト以外のホスト名を指定して送信した JP1 イベントは，自ホストのイベントサーバからは取得できません。
- イベント送信先ホスト名を指定して，JP1 イベントを他ホストのイベントサーバに送信する場合，JP1/Base のイベントサーバ設定ファイル（conf）で設定した「forward-limit」によるリトライの対象にはなりません。
- イベント送信先ホスト名，メッセージ，または拡張属性の値にマクロ変数を指定する場合に，引き継ぐ情報に「"（ダブルクォーテーションマーク）」が含まれていると，情報を正しく引き継げなかったり，エラーが発生したりする原因となります。マクロ変数は，引き継ぐ情報を意識して，正常に動作できる情報を引き継ぐ場合だけ，指定するようにしてください。
- イベント送信先ホスト名を指定しない場合，JP1 イベントは JP1 イベント送信ジョブ実行ホストのイベントサーバに送信されます。
- アクションジョブには「実行時のユーザー」を指定できません。JP1 イベント送信ジョブを実行する［登録ユーザー］または［所有ユーザー］にユーザーマッピングされているプライマリー OS ユーザーが JP1 イベントの発行元ユーザーとなります。
- 環境変数JP1_HOSTNAME に自ホストにない論理ホスト名が設定されている場合は，JP1 イベント送信ジョブの実行ホストが論理ホストであっても，イベント送信元ホスト名は物理ホスト名になります。
- JP1 イベント送信ジョブは，JP1/Base のイベントサーバの機能を利用しています。このため，複数 LAN 環境で JP1 イベントの送信経路などを設定したい場合は，JP1/Base 側で通信経路の設定（conf ファイルの設定）が必要です。JP1/Base 側での通信経路の設定方法については，マニュアル「JP1/Base 運用ガイド」を参照してください。
- JP1/Base のイベントサービス環境の設定で，API 設定ファイルのserver パラメーターの通信タイプに「keep-alive」を設定してください。「close」を設定していると，JP1/AJS3 が起動時に発行する JP1 イベントを発行できなくなったり，統合トレースログにメッセージ KAVT1040-E が出力されて JP1 イベント送信ジョブが異常検出終了したりするなど，正常に動作しない問題が発生することがあります。設定方法および API 設定ファイルについては，マニュアル「JP1/Base 運用ガイド」を参照してください。

7.7.2 メール送信ジョブの注意事項

メール送信ジョブの注意事項（使用する前に知っておいた方がよいこと）について説明します。メールシステムと連携している場合，メール送信します。Windows，UNIX どちらのホストへもメールを送信できます。

メール送信ジョブを定義したジョブネットの例を次に示します。

- ジョブが警告終了したときにメールを送信する。
- ジョブネット中のジョブがすべて終了したらメールを送信する。

メール送信ジョブで実行できる機能の詳細については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド　2.7 メール送信ジョブの定義」を参照してください。

注意事項

アクションジョブには「実行時のユーザー」を指定できません。メール送信ジョブを UNIX ホストで実行する場合は，メール差出人とするユーザーをプライマリー OS ユーザーとしてユーザーマッピングした JP1 ユーザーでログインし，実行登録してください。

7.7.3　JP1/Cm2 状態通知ジョブの注意事項

JP1/Cm2 状態通知ジョブの注意事項（使用する前に知っておいた方がよいこと）について説明します。JP1/Cm2/NNM または HP NNM に任意の状態を通知します。JP1/Cm2/NNM または HP NNM との連携については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド　付録 A JP1/Cm2/NNM または HP NNM を使った監視」を参照してください。

注意事項

JP1/Cm2 状態通知ジョブを UNIX ホストで実行する場合は，スーパーユーザー権限が必要になります。アクションジョブには「実行時のユーザー」を指定できないため，スーパーユーザーをプライマリー OS ユーザーとしてユーザーマッピングした JP1 ユーザーでログインし，実行登録してください。

7.7.4　ローカル電源制御ジョブおよびリモート電源制御ジョブの注意事項

ローカル電源制御ジョブおよびリモート電源制御ジョブの注意事項（使用する前に知っておいた方がよいこと）を次に示します。

- ローカル電源制御ジョブは，JP1/Power Monitor と連携して，マネージャーホストまたはエージェントホストを終了します。
- リモート電源制御ジョブは，JP1/Power Monitor と連携して，ネットワーク上の JP1/Power Monitor のエージェントホストを起動・終了します。制御できるホストは，JP1/Power Monitor のマネージャー・エージェント構成のホストです。JP1/AJS3 のマネージャー・エージェント構成には依存しません。
- リモート電源制御ジョブを実行すると，ジョブを実行するホスト（エージェント）上の JP1/Power Monitor に対して，リモート電源制御を要求します。そのあと，JP1/Power Monitor によってリモート電源制御が行われます。設定された次回電源投入時刻は，リモート電源制御ジョブが実行されるホストの実行ユーザーのタイムゾーンで指定された時刻で動作します。
- リモート電源制御ジョブを実行するホストには，JP1/Power Monitor のリモート電源連携の，マネージャーホストの設定が必要です。

補足事項

イベントジョブ実行継続オプションを使用していると，エージェントホストの JP1/AJS3 が終了してもマネージャーホスト上のイベントジョブは「実行中」状態のままとなります。そのため，計画終了などでマネージャーホスト上のジョブの終了を待って停止する運用の場合は計画停止できません。イベントジョブ実行継続オプションと併用する場合は，実行中のイベントジョブの終了を待つか，イベントジョブを強制終了する，またはスケジュールなどによってイベントジョブが終了している状態で停止するなど，計画的な運用が必要です。

(1) ローカル電源制御ジョブを定義したジョブネットの例

ローカル電源制御ジョブを定義したジョブネットの例を次に示します。

- 先行するジョブが終了したら，実行ホストで指定されたマネージャーホストまたはエージェントホストを終了させる。

なお，JP1/AJS3 からローカル電源制御ジョブを実行するのと，JP1/Power Monitor のカレンダーで計画的に停止するのは，同じ動作です。

ローカル電源制御ジョブで実行できる機能の詳細については，マニュアル「JP1/Power Monitor」を参照してください。

注意事項

ローカル電源制御ジョブを実行する場合の OS ユーザーについては，「(3) 電源制御ジョブ実行時の OS ユーザーについての注意事項」を参照してください。アクションジョブには「実行時のユーザー」を指定できないため，ローカル電源制御ジョブを実行する場合は，Administrators 権限またはスーパーユーザー権限があるユーザーをプライマリー OS ユーザーとしてユーザーマッピングした JP1 ユーザーでログインし，実行登録してください。

(2) リモート電源制御ジョブを定義したジョブネットの例

リモート電源制御ジョブを定義したジョブネットの例を次に示します。

- ジョブの実行前にエージェントホスト（JP1/Power Monitor のエージェントホスト）を起動して，そのホストでジョブを実行する。ジョブが終了したら，ホストも終了する。
- 先行ジョブが異常終了したら，そのジョブを実行していたホスト（JP1/Power Monitor のエージェントホスト）を再起動して，ジョブを実行する。

リモート電源制御ジョブで実行できる機能の詳細については，マニュアル「JP1/Power Monitor」を参照してください。

注意事項

リモート電源制御ジョブを実行する場合の OS ユーザーについては，「(3) 電源制御ジョブ実行時の OS ユーザーについての注意事項」を参照してください。アクションジョブには「実行時のユーザー」を指定できないため，リモート電源制御ジョブを実行する場合は，Administrators 権限またはスーパーユー

ザー権限があるユーザーをプライマリー OS ユーザーとしてユーザーマッピングした JP1 ユーザーでログインし，実行登録してください。

(3)　電源制御ジョブ実行時の OS ユーザーについての注意事項

JP1/Power Monitor と連携する電源制御ジョブ（ローカル電源制御ジョブおよびリモート電源制御ジョブ）は，次の表に示す OS ユーザーで実行する必要があります。

表 7-7　電源制御ジョブを実行する場合の OS ユーザー

OS	OS ユーザー	
Windows	UAC が有効	Administrator ユーザー（ビルトイン Administrator アカウント）※
	UAC が無効	Administrator ユーザー（ビルトイン Administrator アカウント），または Administrators グループユーザー
UNIX	スーパーユーザー	

注※

「UAC 機能有効時に Administrators 権限でジョブを実行するための設定」を有効にすることで，Administrators グループの OS ユーザーでも電源制御ジョブを実行できます。詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.2.22 UAC 機能有効時に Administrators 権限でジョブを実行するための設定」を参照してください。

JP1/Power Monitor の動作する OS については，マニュアル「JP1/Power Monitor」を参照してください。

7.8 フレキシブルジョブ使用時の注意事項

フレキシブルジョブ使用時の注意事項を，エージェントホストの OS ごとに次に示します。

OS 共通の注意事項

- **中継エージェント，一斉配信エージェント，および宛先エージェントに指定できる OS**

  フレキシブルジョブの中継エージェント，一斉配信エージェントおよび宛先エージェントに指定できる OS は，Windows と Linux だけです。また，Windows および Linux 以外の OS でフレキシブルジョブを定義する場合には，中継エージェントを指定する必要があります。

- **フレキシブルジョブの強制終了**

  フレキシブルジョブを強制終了すると，ジョブ自体は強制終了状態になりますが，宛先エージェントの実行プログラムは強制終了できません。

- **物理ホストと論理ホストに関する注意事項**

  フレキシブルジョブは物理ホストでだけ実行できます。宛先エージェントに論理ホスト名を指定しても，物理ホストで実行されます。

- **ユーザーマッピングに関する注意事項**

  中継エージェントおよび宛先エージェントの両方でユーザーマッピングできるようにする必要があります。中継エージェントを使用する場合，宛先エージェントのユーザーマッピング定義の［サーバホスト名］に中継エージェントのホスト名または「*（アスタリスク）」を設定する必要があります。

- **環境設定パラメーター FXJOB_MONITOR_TIMEOUT に関する注意事項**

  ・フレキシブルジョブは，環境設定パラメーター`FXJOB_MONITOR_TIMEOUT` に指定した時間（デフォルト値は 60 分）を経過すると，ジョブの状態が異常検出終了となります。60 分以上掛かるジョブがある場合，環境設定パラメーター`FXJOB_MONITOR_TIMEOUT` に適切な値を設定してください。

  ・フレキシブルジョブ実行時にエラーが発生した場合，ジョブが異常検出終了となるまでに，環境設定パラメーター`FXJOB_MONITOR_TIMEOUT` に指定した時間（デフォルト値は 60 分）が掛かる場合があります。

- **通信エラーが発生した場合の注意事項**

  通信エラーが発生した場合，フレキシブルジョブの状態は異常検出終了になりますが，宛先エージェントでは正常にユーザープログラムが実行されていることがあります。フレキシブルジョブが異常検出終了になった場合は，ユーザープログラムの実行結果を参照して対処してください。

- **標準入力ファイルからのデータの入力に対応していないコマンド**

  標準入力ファイルからのデータの入力に対応していないコマンドは実行できません。標準入力ファイルからのデータの入力に対応していないコマンドの例としては，Windows の`timeout` コマンドがあります。

- **サービスの停止に関する注意事項**

  フレキシブルジョブを実行する時間帯に，中継エージェントや宛先エージェントで，JP1/AJS3 Autonomous Agent サービスや JP1/AJS3 Autonomous AgentMessenger サービスを停止しないでください。停止した場合，フレキシブルジョブが異常検出終了します。また，環境設定パラメー

ターFXJOB_MONITOR_TIMEOUTに指定した時間（デフォルト値は60分）が経過するまで実行中のままになる場合もあります。

- **フレキシブルジョブ実行時のワークパス**

  フレキシブルジョブ実行時のワークパスは変更できません。フレキシブルジョブ実行時のワークパスを次に示します。

  ・Windowsでインストール先フォルダがデフォルトまたはシステムで保護されたフォルダ配下の場合

  %ALLUSERSPROFILE%¥Hitachi¥JP1¥JP1_DEFAULT¥JP1AJS2¥tmp¥AJSFX_WORK_00000001

  ・Windowsでインストール先が上記以外の場合

  JP1/AJS3のインストール先フォルダ¥tmp¥AJSFX_WORK_00000001

  ・UNIXの場合

  /var/opt/jp1ajs2/tmp/AJSFX_WORK_00000001

- **フレキシブルジョブの再実行**

  ロードバランサーを使用する場合，フレキシブルジョブを再実行またはリトライすると，実行先は再度振り分けられます。

  一斉実行機能を使用する場合，フレキシブルジョブを再実行すると，すべての宛先エージェントで再実行されます。

- **リモートジョブネットとの組み合わせについて**

  リモートジョブネットとフレキシブルジョブを組み合わせて使用することは推奨しません。組み合わせて使用した場合，次に示すように，リモートジョブネットの定義だけでなく，フレキシブルジョブの実行要求も転送されます。このため，障害発生時に問題の切り分けが難しくなるおそれがあります。

1. マネージャーホストのリモートジョブネットの定義が，別のマネージャーホストに転送される。
2. 転送先のマネージャーホストから中継エージェントにフレキシブルジョブの実行要求が行われる。
3. 中継エージェントからロードバランサーに実行プログラムの実行要求が行われる。
4. ロードバランサーから宛先エージェントに実行プログラムの実行要求が転送される。

- **フレキシブルジョブの後続ジョブに引き継ぎ情報設定ジョブを指定している場合の注意事項**

  フレキシブルジョブの後続の引き継ぎ情報設定ジョブが引き継げる標準出力ファイルの内容は，4,096バイトまでです。4,096バイトを超える値を出力しても，使用できるのは4,096バイト分までです。4,096バイトの制限は，中継エージェントおよび宛先エージェントの両方で実施します。このため，中継エージェントと宛先エージェントの両方の文字コードで，4,096バイト以内となるようにしてください。なお，4,096バイトから4,097バイト目のデータがマルチバイト文字の場合は，「?」に置き換えられます。

- **フレキシブルジョブにOSのシャットダウンコマンドを登録しないこと**

  フレキシブルジョブに自ホストや中継エージェントや宛先エージェントに対するOSのシャットダウンコマンドを登録して実行しないでください。WindowsのシャットダウンはJP1/AJS3の停止完了を待たないため，JP1/AJS3が稼働中にシャットダウンを実行すると，JP1/AJS3のデータファイルが壊れるなど問題が発生するおそれがあります。ジョブの自動運用でシャットダウンしたい場

合は，JP1/Power Monitor をインストールし，アクションジョブの「ローカル電源制御ジョブ」を実行する運用を検討してください。

また，手動でシステムを終了する場合，JP1/Power Monitor の電源制御コマンド（aompwcon コマンド）を実行してください。JP1/Power Monitor をインストールしないで手動でシャットダウンを実行する場合は，JP1/AJS3 のサービスを手動で停止したあとにシャットダウンを実行してください。

- **一斉実行機能利用時の，宛先エージェント上でのジョブの状態確認**

  フレキシブルジョブの一斉実行を使用している場合，JP1/AJS3 - View では宛先エージェント上でのジョブの実行状態は確認できません。宛先エージェント上でのジョブの状態を細かく管理したい場合は，標準ジョブを使用してください。

- **一斉実行機能利用時の実行結果詳細**

  フレキシブルジョブの一斉実行を同期モードで使用している場合に，JP1/AJS3 - View で確認できる実行結果は，各宛先エージェントの標準エラー出力をすべてマージした内容になります。そのため，マージされる実行結果のサイズが巨大にならないように，個々の標準エラー出力は最大 384 バイトで制限され，385 バイト以降は打ち切られます。

- **一斉実行機能で使用する IP アドレス**

  フレキシブルジョブの一斉実行を使用する場合，一斉配信エージェントおよび一斉配信先の宛先エージェントで使用する IP アドレスは，互いに通信できる IP アドレスを使用してください。

- **一斉実行機能利用時のジョブ実行時間**

  通信できない一斉配信先の宛先エージェントが存在し，かつ通信できない時間が環境設定パラメーターFXBC_MANAGEDAGT_REMOVEDTIME に設定されている値を超えている場合に一斉実行すると，ジョブは正常に実行されて終了するにもかかわらず，次の理由からジョブ実行終了まで時間が掛かることがあります。

一斉配信エージェントから一斉配信先の宛先エージェントへの接続がタイムアウトと判定されるまで時間が掛かるが，通信できない一斉配信先の宛先エージェントが削除され，存在しないものとして処理が続行されるため。

- **接続元制限機能の利用**

  接続元制限機能を利用する場合，マネージャーホストから中継エージェントへの接続は，中継エージェントのエージェント用接続許可設定ファイルで接続を制限できます。しかし，宛先エージェントへの接続，および一斉配信エージェントへの接続は，接続元制限機能で制限できません。

  接続元制限については，マニュアル「JP1/Automatic Job Management System 3 設計ガイド（システム構築編） 2.3.9 JP1/AJS3 へ接続するホストの制限」を参照してください。

- **ホスト名の長さについての注意事項**

  フレキシブルジョブを使用する場合，次のホストのホスト名は，128 バイト以下になるようにしてください。

  ・マネージャーホスト

  ・中継エージェント

  ・一斉配信エージェント

  ・宛先エージェント

エージェントホストが Windows の場合の注意事項

- **UAC 機能についての注意事項**

  Windows の UAC 機能を有効にしていると，ビルトイン Administrator を除く，すべての Administrators グループの OS ユーザーは，Administrators 権限が無効な状態で動作します。このため，フレキシブルジョブでは，UAC 機能が有効の場合，Administrators 権限を必要とするジョブは実行できません。

- **システムのリソース不足の発生を少なくするための対応**

  ある数以上のジョブを同時に実行すると，システムの環境によってはシステムのリソース（デスクトップヒープ領域）不足が発生し，エラーになることがあります。こうした場合の対応として次のことを検討してください。

  ・JP1/AJS3 サービスのアカウントをユーザーアカウントにしてください。JP1/AJS3 サービスのアカウントを，ほかのサービスのアカウントやログオンユーザーと異なるユーザーアカウントにすれば，デスクトップヒープ領域を共用しないで運用できます。

  ・フレキシブルジョブを実行するときによく使用する OS ユーザーを，JP1/AJS3 のサービスのアカウントと同じにすることで，デスクトップヒープ領域の消費を少なくできます。

- **アプリケーションファイル名に空白文字が含まれている場合の対応**

  ファイルタイプ（拡張子）に関連づけられているアプリケーションファイル名に空白文字が含まれるときは，エクスプローラのファイルタイプの編集で確認し，アプリケーションファイル名を「"」（ダブルクォーテーションマーク）で囲んでください。

- **フレキシブルジョブで指定するファイル名を 254 バイト以内にする**

  次に示すファイル名に文字数が 255 バイト以上のファイル名を指定した場合，ジョブが「起動失敗」状態または「異常検出終了」状態になることがあります。

  ・ジョブの実行ファイル名

  ・ジョブの環境変数ファイル名

  ・ジョブの終了判定のファイル名

  上記のファイル名には，254 バイト以内のファイル名を指定してください。

- **ユーザープロファイルを必要とするジョブについての注意事項**

  フレキシブルジョブではユーザープロファイルを読み込みません。このため，フレキシブルジョブではユーザープロファイルを必要とするジョブは実行できません。

エージェントホストが Linux の場合の注意事項

- **フレキシブルジョブで設定できる終了コードについて**

  フレキシブルジョブで設定できる終了コードの値は 0～255 です。ジョブの起動失敗や，標準出力データまたは標準エラー出力データの取得に失敗した場合は，終了コードは-1 となります。

- **フレキシブルジョブが「起動失敗」状態にならないようにするための注意**

  JP1/AJS3 サービスの起動ユーザーおよびフレキシブルジョブの実行 OS ユーザーに，次に示すファイルおよび格納先ディレクトリに対する書き込み権限および読み込み権限がないと，フレキシブルジョブが「起動失敗」状態になることがあります。そのため，JP1/AJS3 サービスの起動ユーザー

に，次に示すファイルおよび格納先ディレクトリに対する書き込み権限および読み込み権限を設定してください。

・実行 OS ユーザーのホームディレクトリ

・ジョブ実行制御のログファイル※

注※

ジョブ実行制御で使用するログファイルの詳細については，マニュアル「JP1/Automatic Job Management System 3 トラブルシューティング　1.2.5 ログファイルおよびディレクトリ一覧」を参照してください。

- **フレキシブルジョブ実行時のリソース制限値についての注意事項**

  UNIX 版 JP1/AJS3 では，JP1/AJS3 起動時のリソース制限値がジョブ実行時にも有効になります。そのため，リソースの制限値を設定したい場合は，JP1/AJS3 の起動ユーザーである root ユーザーに対して設定してください。ただし，実行するフレキシブルジョブに制限値を指定した場合は，指定した値が有効になります。詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5 ジョブ実行環境設定」を参照してください。

  ファイルサイズの制限値を変更する例を次に示します。

1. **root ユーザーのログインプロファイル（通常は[/.profile]（$HOME/.profile））に次に示す記述を設定する。**

   fsize には必要なファイルサイズを設定します。無制限にする場合は，`unlimited` を設定してください。

   ulimit -f fsize

2. **root ユーザーでログインする。**

3. **JP1/AJS3 サービスを root ユーザーで起動する。**

   fsize の値が有効になります。

注意事項

OS のリソース設定ファイル（`/etc/security/limits.conf`）の値は，telnet 接続などで起動される `login` コマンド経由のプロセスでだけ有効になります。JP1/AJS3 から起動されるジョブは，サービスから起動されるプロセスであるため OS のリソース設定ファイルの設定は有効になりません。

- **フレキシブルジョブ実行時の環境変数 LANG についての注意**

  次に示す環境変数`LANG` には同じ文字コード種別を設定してください。

  ・JP1/AJS3 Autonomous Agent サービス起動時の環境変数`LANG`

  ・JP1/AJS3 Autonomous Agent Messenger サービス起動時の環境変数`LANG`

  ・root ユーザーのログインプロファイルの環境変数`LANG`

- **JP1/AJS3 サービスを自動起動している場合の注意**

  JP1/AJS3 サービスを自動起動している場合，root ユーザーのログインプロファイルは読み込みません。そのため，ログインプロファイルで root ユーザーのリソース制限値を変更する設定にしても，手動でログインして JP1/AJS3 サービスを起動した際に設定されるリソース制限値と値が異なることがあります。この場合，ジョブ実行環境の環境設定パラメーターでリソース制限値を設定し

てください。ジョブ実行環境の環境設定パラメーターについては，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5 ジョブ実行環境設定」を参照してください。

また，リソース制限値は JP1/AJS3 の自動起動スクリプト（/etc/opt/jp1ajs2/jajs_start）にも記載できます。その場合は，十分な検証を実施した上で運用してください。

なお，フレキシブルジョブ実行時のグループ ID は，root ユーザーでログインしたときに設定されるグループ ID と異なります。詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　5.4.12 ジョブ実行時のグループ ID（UNIX 限定）」を参照してください。

- **端末を必要とするユーザープログラムをジョブとして実行する場合の注意**

  UNIX 版 JP1/AJS3 で端末を必要とするユーザープログラムをジョブとして実行すると，正しく動作しない（フレキシブルジョブが異常終了する）場合があります。

- **OS ユーザーを登録または更新する場合の注意**

  フレキシブルジョブ実行中に，システム管理者権限でpasswd コマンドなどによって OS ユーザーを登録または更新しないでください。OS ユーザーの登録または更新が終了したあとにジョブを実行してください。

マネージャーホストが Linux で中継エージェントを指定していない場合，または中継エージェントホストが Linux の場合の注意事項

- **ログインシェルの注意事項**

  JP1 ユーザーにマッピングされている OS ユーザーのログインシェルには，bash またはsh を指定する必要があります。

7.9 HTTP 接続ジョブ使用時の注意事項

HTTP 接続ジョブ使用時の注意事項を，エージェントホストの OS ごとに次に示します。

なお，HTTP 接続ジョブが起動失敗したり，異常終了したりする場合の要因や注意事項については，マニュアル「JP1/Automatic Job Management System 3 トラブルシューティング　2.6.2 標準ジョブ，HTTP 接続ジョブ，アクションジョブ，およびカスタムジョブのトラブルへの対処」もあわせて参照してください。

エージェントホストが Windows の場合の注意事項

- **システムのリソース不足の発生を少なくするための対応**

  ある数以上のジョブを同時に実行すると，システムの環境によってはシステムのリソース（デスクトップヒープ領域）不足が発生し，エラーになることがあります。こうした場合の対応として次のことを検討してください。

  ・JP1/AJS3 サービスのアカウントをユーザーアカウントにしてください。JP1/AJS3 サービスのアカウントを，ほかのサービスのアカウントやログオンユーザーと異なるユーザーアカウントにすれば，デスクトップヒープ領域を共用しないで運用できます。

  ・HTTP 接続ジョブを実行するときによく使用する OS ユーザーを，JP1/AJS3 のサービスのアカウントと同じにすることで，デスクトップヒープ領域の消費を少なくできます。

- **HTTP 接続ジョブが「起動失敗」状態にならないようにするための注意**

  JP1/AJS3 サービスの起動ユーザーに次に示す権限がないと，HTTP 接続ジョブが「起動失敗」状態になることがあります。そのため，JP1/AJS3 サービスの起動ユーザーに，次に示す権限を設定してください。

  ・HTTP 接続ジョブの標準出力ファイル名の読み取り権限および書き込み権限

  ・HTTP 接続ジョブの標準エラー出力ファイル名の読み取り権限および書き込み権限

- **ジョブで指定するファイル名を 254 バイト以内にする**

  次に示すファイル名に文字数が 255 バイト以上のファイル名を指定した場合，ジョブが「起動失敗」状態または「異常検出終了」状態になることがあります。

  ・ジョブの標準出力ファイル名

  ・ジョブの標準エラー出力ファイル名

  ・ジョブの終了判定のファイル名

  上記のファイル名には，254 バイト以内のファイル名を指定してください。

エージェントホストが Linux の場合の注意事項

- **HTTP 接続ジョブが「起動失敗」状態にならないようにするための注意**

  JP1/AJS3 サービスの起動ユーザーおよび HTTP 接続ジョブの実行 OS ユーザーに，次に示すファイルおよび格納先ディレクトリに対する書き込み権限および読み込み権限がないと，ジョブが「起動失敗」状態になることがあります。そのため，JP1/AJS3 サービスの起動ユーザーに，次に示すファイルおよび格納先ディレクトリに対する書き込み権限および読み込み権限を設定してください。

  ・HTTP 接続ジョブの標準出力ファイル

・HTTP 接続ジョブの標準エラー出力ファイル

・実行 OS ユーザーのホームディレクトリ

・ジョブ実行制御のログファイル※

注※

ジョブ実行制御で使用するログファイルの詳細については，マニュアル「JP1/Automatic Job Management System 3 トラブルシューティング　1.2.5 ログファイルおよびディレクトリ一覧」を参照してください。

- **ジョブ実行時のリソース制限値についての注意事項**

UNIX 版 JP1/AJS3 では，JP1/AJS3 起動時のリソース制限値がジョブ実行時にも有効になります。そのため，リソースの制限値を設定したい場合は，JP1/AJS3 の起動ユーザーである root ユーザーに対して設定してください。ただし，実行する HTTP 接続ジョブに制限値を指定した場合は，指定した値が有効になります。詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5 ジョブ実行環境設定」を参照してください。

ファイルサイズの制限値を変更する例を次に示します。

1. **root ユーザーのログインプロファイル（通常は[/.profile]（$HOME/.profile））に次に示す記述を設定する。**

   fsize には必要なファイルサイズを設定します。無制限にする場合は，`unlimited` を設定してください。

   ulimit -f fsize

2. **root ユーザーでログインする。**

3. **JP1/AJS3 サービスを root ユーザーで起動する。**

   fsize の値が有効になります。

注意事項

OS のリソース設定ファイル（`/etc/security/limits.conf`）の値は，telnet 接続などで起動される `login` コマンド経由のプロセスでだけ有効になります。JP1/AJS3 から起動される HTTP 接続ジョブは，サービスから起動されるプロセスであるため OS のリソース設定ファイルの設定は有効になりません。

- **JP1/AJS3 サービスを自動起動している場合の注意**

JP1/AJS3 サービスを自動起動している場合，root ユーザーのログインプロファイルは読み込みません。そのため，ログインプロファイルで root ユーザーのリソース制限値を変更する設定にしても，手動でログインして JP1/AJS3 サービスを起動した際に設定されるリソース制限値と値が異なることがあります。この場合，ジョブ実行環境の環境設定パラメーターでリソース制限値を設定してください。ジョブ実行環境の環境設定パラメーターについては，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5 ジョブ実行環境設定」を参照してください。

また，リソース制限値は JP1/AJS3 の自動起動スクリプト（`/etc/opt/jp1ajs2/jajs_start`）にも記載できます。その場合は，十分な検証を実施した上で運用してください。

なお，HTTP 接続ジョブ実行時のグループ ID は，root ユーザーでログインしたときに設定されるグループ ID と異なります。詳細については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　5.4.12 ジョブ実行時のグループ ID（UNIX 限定）」を参照してください。

- OS ユーザーを登録または更新する場合の注意

  HTTP 接続ジョブ実行中に，システム管理者権限で`passwd` コマンドなどによって OS ユーザーを登録または更新しないでください。OS ユーザーの登録または更新が終了したあとに HTTP 接続ジョブを実行してください。

7.10　ジョブ定義についての注意事項

ジョブ定義についての注意事項を次に示します。

7.10.1　標準出力ファイル・標準エラー出力ファイルの注意事項

標準出力ファイルおよび標準エラー出力ファイルについての注意事項を次に示します。

(1)　標準出力ファイル・標準エラー出力ファイルを定義する場合

ジョブ（キューレスジョブを含む PC ジョブ，UNIX ジョブ，フレキシブルジョブ，および HTTP 接続ジョブ）の実行時に出力する標準出力ファイル・標準エラー出力ファイルを定義する場合，次の注意事項があります。

1. ジョブネットから実行するジョブはデフォルトで標準エラー出力をジョブの実行結果として取得します。標準出力はジョブの実行結果として取得されません。標準エラー出力の内容は，［JP1/AJS3 - View］ウィンドウの［実行結果詳細］ダイアログボックスで表示できます。なお，［JP1/AJS3 - View］ウィンドウの［実行結果詳細］ダイアログボックスには，JP1/AJS3 が標準エラー出力に出力するエラーメッセージも表示されます。
2. 標準出力・標準エラー出力の内容を任意のファイルに出力したい場合は，ジョブを定義する際に標準出力ファイル名・標準エラー出力ファイル名を指定してください。
3. ［JP1/AJS3 - View］ウィンドウの［実行結果詳細］ダイアログボックスに標準出力の内容も表示する場合は，［標準出力ファイル名］と［標準エラー出力ファイル名］に同じファイル名を指定してください。
4. ［標準出力ファイル名］および［標準エラー出力ファイル名］に同じファイル名を指定する場合は，追加書きオプションの設定を一致させてください。一方に新規作成，もう一方に追加書きを設定すると，パラメーターエラーになり，ジョブが起動失敗します。
5. ［詳細定義－［**アイコン名**］］ダイアログボックスの［環境変数ファイル名］，［標準入力ファイル名］，［標準出力ファイル名］，および［標準エラー出力ファイル名］に，［スクリプトファイル名］または［実行ファイル名］と同じファイル名を指定している場合，ジョブが起動失敗または異常検出終了となるおそれがあります。これらの項目には，［スクリプトファイル名］または［実行ファイル名］と同じファイル名を指定しないでください。
6. 同時に実行する複数のジョブの定義に，同じ標準出力ファイルや標準エラー出力ファイルを指定してジョブを実行すると，標準出力や標準エラー出力の出力結果が上書きされます。また，標準出力や標準エラー出力を指定したジョブを多重起動して同時に実行した場合も，標準出力や標準エラー出力の出力結果が上書きされます。

   同時に実行するジョブの［標準出力ファイル名］，［標準エラー出力ファイル名］には，それぞれ別のファイルを指定してください。

7. ジョブの［標準出力ファイル名］または［標準エラー出力ファイル名］に，アクセスできないネットワークファイル名を指定した場合，ジョブが起動失敗または異常検出終了します。正しいネットワークファイル名を指定してください。
8. ［標準出力ファイル名］または［標準エラー出力ファイル名］にネットワークファイル名を指定している場合，指定したファイルには，次に示す権限が必要です。

   Windows の場合

   実行ホストの JP1/AJS3 サービス（キューレスジョブの場合は JP1/AJS3 Queueless Agent サービス）に設定されているアカウントで作成，参照および更新する権限

   UNIX の場合

   ジョブの実行 OS ユーザーに指定されているアカウントで作成，参照および更新する権限
9. ［標準出力ファイル名］を指定した場合，指定しなかった場合に比べてジョブネットの処理性能に影響することがあります。

補足事項

`jpqjobsub` コマンドから実行するサブミットジョブの場合はデフォルトでは標準出力・標準エラー出力を保存しません。`jpqjobsub` コマンドのオプションに任意のファイル名を指定してください。

(2) 標準出力ファイル・標準エラー出力ファイルに大量データを出力する場合

JP1/AJS3 では，ジョブ（キューレスジョブを除く標準ジョブ，アクションジョブ，またはカスタムジョブ）の終了時にエージェントホストからマネージャーホストに標準出力ファイルまたは標準エラー出力ファイルを転送しています。標準出力ファイルまたは標準エラー出力ファイルに数メガバイトを超えるような大量データを出力するジョブを実行すると,ファイルのデータ解析処理またはファイルのデータ転送で負荷が掛かり，CPU 使用率の増加や JP1/AJS3 が使用するメモリー使用量の増加など，ジョブの実行が遅延するだけでなく，システム全体の処理に影響を与えるおそれがあります。これらの問題は，マネージャーホストとエージェントホストの両方で発生します。

また，追加書きオプションを設定すると，ジョブ実行のたびに出力データが蓄積され，転送するファイルのサイズが単調増加します。この場合も，ファイルのデータ解析処理またはファイルのデータ転送で負荷が掛かり，ジョブが異常終了したり，エージェントホストからマネージャーホストに対するデータ転送が遅延したりすることがあります。この場合，追加書きオプションの設定を無効にするか，または標準出力データファイルや標準エラー出力データファイルを定期的に削除，退避するようにしてください。

標準出力ファイルや標準エラー出力ファイルのファイルサイズが原因となって発生する問題を回避する方法として，JP1/AJS3 では標準出力ファイルや標準エラー出力ファイルのファイルサイズに上限値を設定できます。この設定を行うことで，マネージャーホストでファイルサイズの上限値を超えたデータの破棄や，上限値に達したことを示す警告メッセージを出力できます。これによって，一部のジョブの処理で JP1/AJS3 のジョブ実行全体に影響が出ないようにすることができます。設定方法の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.2.7 ファイル受信制限をするための設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.2.7 ファイル受信制限をするための設定」（UNIX の場合）を参照してください。

また，エージェントホストの設定を行うことで同様にファイルサイズの上限値を超えたデータの破棄や，上限値に達したことを示す警告メッセージを出力できます。これによって，一部のジョブの処理で JP1/AJS3 のジョブ実行全体に影響が出ないようにすることができます。設定方法の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.2.27 ファイル送信制限をするための設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.2.24 ファイル送信制限をするための設定」（UNIX の場合）を参照してください。

上記のように，標準出力ファイルや標準エラー出力ファイルのファイルサイズに上限値を設定していない状態で，ジョブが正常に動作しない場合は，次に示す点に注意して確認した上，それぞれの対処方法で対応してください。

- 標準出力データが大量に出力される

  PC ジョブや UNIX ジョブの場合，ジョブの定義で［標準出力ファイル名］を明示的に指定している場合にだけファイルの転送を行っており，標準出力データを大量に出力するとジョブが異常終了することがあります。この場合，ジョブの定義の［標準出力ファイル名］には何も指定しないでください。ジョブのバッチファイルやシェルスクリプト内で，標準出力をリダイレクトしてください。

  ジョブの定義の［標準出力ファイル名］に何も指定しなければ，標準出力ファイルはマネージャーホストに転送されません。

- 標準エラー出力データが大量に出力される

  PC ジョブや UNIX ジョブの場合，ジョブ定義の［標準エラー出力ファイル名］に NULL デバイスを指定してください。UNIX 上で実行するジョブの場合は"/dev/null"を記述し，Windows 上で実行するジョブの場合は"NUL"を記述してください。ジョブ定義の［標準エラー出力ファイル名］にファイル名を明示的に指定している場合は，バッチファイルやシェルスクリプト内で標準エラー出力をリダイレクトしてください。

  この設定をすると，標準エラー出力ファイルはマネージャーホストに転送されません。その場合は，JP1/AJS3 - View の［実行結果詳細］ダイアログボックスで標準エラー出力ファイルの内容を参照できなくなります。

補足事項

1. この現象が発生すると，エージェントホストのジョブ実行環境のワークディレクトリに「A_JPQ*_**ジョブ番号**」という一時ファイルが残ることがあります。エージェントホストのジョブ実行環境のワークディレクトリは，[{JP1_DEFAULT|**論理ホスト名**}¥JP1NBQAGENT¥Process]の環境設定パラメーターWorkPath に指定したフォルダです。環境設定パラメーターWorkPath のデフォルト値などの詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5.2(37) WorkPath（エージェントプロセス用）」を参照してください。

   なお，異常検出終了したジョブと同じジョブ番号の一時ファイルは削除しても問題ありません。

2. サブミットジョブで標準出力データ，または標準エラー出力データを大量に出力する場合，マネージャーホストのジョブ実行環境のワークディレクトリに作成される一時ファイル（M_JPQSTDE_**ジョブ番号**，または M_JPQSTDO_**ジョブ番号**）のデータも大量になります。通常は，ジョブ情報削除処理実行時に保存日数を経過しているジョブの一時ファイルは，自動的に削除されます。しかし，一時ファイルによるディスク容量不足などが発生した場合には，ジョブの終了後に手動で削除

できます。また，ディスク容量不足が慢性的に発生する場合には，ジョブ情報保存日数の縮小や，ジョブのスクリプトファイル内で標準出力・標準エラー出力のリダイレクトを検討してください。

一時ファイルを削除した場合には，jpqjobget コマンドを使用して標準出力・標準エラー出力ファイルを参照できなくなります。マネージャーホストのジョブ実行環境のワークディレクトリは，[{JP1_DEFAULT|論理ホスト名}¥JP1NBQMANAGER¥Process]の環境設定パラメーターWorkPath に設定したフォルダです。環境設定パラメーターWorkPath のデフォルト値などの詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.5.2(1) WorkPath（マネージャープロセス用)」を参照してください。

ジョブの詳細定義の［パラメーター］にリダイレクトを指定する方法とその例を，次に示します。

表 7-8　ジョブの詳細定義の［パラメーター］にリダイレクトを指定する方法と例（Windows の場合）

定義内容	指定方法	指定例
標準出力ファイル，標準エラー出力ファイルが異なる場合（実行ファイルが exe の場合）	実行ファイル名：test.exe パラメーター：なし 標準出力ファイル：out.txt 標準エラー出力ファイル：err.txt	実行ファイル名：cmd.exe パラメーター： /C test.exe >out.txt 2>err.txt 標準出力ファイル：なし 標準エラー出力ファイル：なし
標準出力ファイル，標準エラー出力ファイルが異なる場合（実行ファイルが bat の場合）	実行ファイル名：test.bat パラメーター：なし 標準出力ファイル：out.txt 標準エラー出力ファイル：err.txt	実行ファイル名：test.bat パラメーター：>out.txt 2>err.txt 標準出力ファイル：なし 標準エラー出力ファイル：なし
標準出力ファイル，標準エラー出力ファイルが同名の場合	実行ファイル名：test.bat パラメーター：なし 標準出力ファイル：out.txt 標準エラー出力ファイル：out.txt	実行ファイル名：test.bat パラメーター：>out.txt 2>&1 標準出力ファイル：なし 標準エラー出力ファイル：なし

表 7-9　ジョブの詳細定義の［パラメーター］にリダイレクトを指定する方法と例（UNIX の場合）

定義内容	指定方法	指定例
標準出力ファイル，標準エラー出力ファイルが異なる場合	スクリプトファイル名：test.sh パラメーター：なし 標準出力ファイル：out.txt 標準エラー出力ファイル：err.txt	スクリプトファイル名：test.sh パラメーター：>out.txt 2>err.txt 標準出力ファイル：なし 標準エラー出力ファイル：なし
標準出力ファイル，標準エラー出力ファイルが同名の場合	スクリプトファイル名：test.sh パラメーター：なし 標準出力ファイル：out.txt 標準エラー出力ファイル：out.txt	スクリプトファイル名：test.sh パラメーター：>out.txt 2>&1 標準出力ファイル：なし 標準エラー出力ファイル：なし

(3) 標準出力ファイル・標準エラー出力ファイルのファイル受信制限機能とファイル送信制限機能について

標準出力ファイルまたは標準エラー出力ファイルに数メガバイトを超えるような大量データを出力するジョブを実行すると，ファイルのデータ解析処理またはファイルのデータ転送で負荷が掛かり，CPU 使用率の増加や JP1/AJS3 が使用するメモリー使用量の増加など，ジョブの実行が遅延するだけでなく，システム全体の処理に影響を与えるおそれがあります。そのため JP1/AJS3 11-00 以降では，デフォルトの機能としてマネージャーホストでファイル受信制限を行い，エージェントホストでファイル送信制限を行います。

ファイル受信制限機能とファイル送信制限機能は，ファイルサイズ上限を超えたデータの破棄と，上限値に達したことを示すメッセージの出力を行います。マネージャーのファイル受信制限のサイズは 5 メガバイト（5,242,880 バイト），エージェントのファイル送信制限のサイズは 3 メガバイト（3,145,728 バイト）です。

ファイルサイズ上限を超えた時の動作とファイルサイズ上限値については変更できます。設定方法の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.2.7 ファイル受信制限をするための設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.2.7 ファイル受信制限をするための設定」（UNIX の場合）を参照してください。また，エージェントのファイルサイズ上限を超えた時の動作とファイルサイズ上限値についても変更できます。設定方法の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.2.27 ファイル送信制限をするための設定」（Windows の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.2.24 ファイル送信制限をするための設定」（UNIX の場合）を参照してください。

補足事項

マネージャーホストの JP1/AJS3 をバージョンアップインストールした場合，ファイル受信制限機能はバージョンアップ前の環境設定パラメーターの値を引き継ぎます。エージェントホストの JP1/AJS3 をバージョンアップインストールした場合，ファイル送信機能はファイルサイズ上限を超えたデータを破棄せず，上限値に達したことを示すメッセージを出力します。

(4) その他の注意事項

1. 標準出力データや標準エラー出力データの出力先に指定できるファイルは，テキスト形式のファイルだけです。
2. 標準出力データや標準エラー出力データには，外字・機種依存文字を使用しないでください。これらの文字を使用した場合，JP1/AJS3 - View では正しく表示できません。
3. 標準出力データ・標準エラー出力データは，エージェントホスト上に一時的にファイルとして作成され，ジョブ終了時にマネージャーホストに転送されます。これらの一時ファイルはマネージャーホスト，エージェントホストのジョブ実行環境のワークディレクトリに，それぞれ次に示す名前で作成されます。

   マネージャーホスト（サブミットジョブを実行した場合）

   M_JPQSTDE_**ジョブ番号**

M_JPQSTDO_ジョブ番号

エージェントホスト

A_JPQSTDE_*_ジョブ番号

A_JPQSTDO_*_ジョブ番号

それぞれのホストで削除されるタイミングは，次のとおりです。

マネージャーホスト

QUEUE ジョブ・サブミットジョブについての一時ファイルはジョブ情報削除処理の実行時に削除され，それ以外のジョブの一時ファイルはジョブ終了時に削除されます。

エージェントホスト

一時ファイルはジョブ終了時に自動的に削除されます。

なお，キューレスジョブの場合は，標準エラー出力データだけが一時的にファイルとしてエージェントホスト上に保存されます。このファイルの内容は，キューレスジョブ終了時にマネージャーホストへ転送されます。このファイルは，キューレスジョブ終了時，またはキューレスエージェントサービス起動時に自動的に削除されます。

フレキシブルジョブで実行したユーザープログラムの標準出力データ，および標準エラー出力データは常にマネージャーホストに転送されます。転送されたファイルのファイル名は，ajsshow コマンドの-i オプションに「%so」を指定して実行することで確認できます。

4. ［標準エラー出力ファイル名］を指定し，［追加書き］を選択しているジョブの実行後に JP1/AJS3 - View で［実行結果詳細］ダイアログボックスの内容を表示した場合，ジョブの実行先サービスによって次の違いがあります。
   - ［実行先サービス］に［標準］を指定したジョブの場合
     ［標準エラー出力ファイル名］に指定したファイルの内容が表示されます。
   - ［実行先サービス］に［キューレス］を指定したジョブの場合
     ジョブ実行時に標準エラー出力に出力された内容が表示されます。

7.10.2　ジョブの実行優先順位に関する注意事項

JP1/AJS3 では，ジョブ（キューレスジョブを含む PC ジョブ，UNIX ジョブ，HTTP 接続ジョブ，JP1/AJS3 上で実行する QUEUE ジョブ，およびカスタムジョブ）に対して実行優先順位を指定します。また，フレキシブルジョブ，アクションジョブでは，上位ジョブネットに優先順位を設定することで実行優先順位を指定します。

指定できる実行優先順位は［なし］，および［1］～［5］です。ジョブの実行優先順位，およびその上位のジョブネットの優先順位に［なし］が定義されている場合は，環境設定パラメーターDEFAULTPRIORITY に指定された値を仮定します。また，ジョブの実行優先順位，ジョブネットの優先順位に［なし］が定義され，環境設定パラメーターDEFAULTPRIORITY が設定されていない場合，JP1/AJS3 から実行するジョブの実行優先順位は，［1］が仮定されます。環境設定パラメーターDEFAULTPRIORITY の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2(108) DEFAULTPRIORITY」

を参照してください。次に示す注意事項を参照し，システム環境や運用に応じてジョブの実行優先順位，およびジョブネットの優先順位を変更する必要がないかどうかを検討してください。

Windows，UNIX それぞれについて，JP1/AJS3 がジョブの実行優先順位に従ってジョブに設定する値の詳細について次に説明します。

(1) ジョブの実行優先順位と JP1/AJS3 がジョブに設定する値（UNIX の場合）

UNIX の場合，ジョブの実行優先順位に従って nice 値を設定します。また，ジョブの実行先サービスに［標準］を指定した場合と［キューレス］を指定した場合とで設定の基準が異なります。

(a) ジョブの実行先サービスに［標準］を指定した場合

ジョブの実行先サービスに［標準］を指定した場合，JP1/AJS3 から実行するジョブに設定する nice 値は，JP1/AJS3 のサービスを起動（`jajs_spmd` コマンドを実行）したときの nice 値※が基準となります。ジョブに設定する nice 値は，この基準となる nice 値と指定した実行優先順位に従って増減して設定します。

注※

JP1/AJS3 サービスを起動したときの nice 値は，プロセスを起動したときに OS がデフォルトで設定する値です。また，システムが制限する nice 値の最大値，最小値は，OS によって異なります。プロセスを起動したときのデフォルトの nice 値やシステムが制限する nice 値の最大値，最小値については，各 OS のドキュメントや UNIX の参考文献を参照してください。

HP-UX で，JP1/AJS3 のサービスを起動したときのデフォルトの nice 値が「20」の場合を例に，ジョブの実行先サービスに［標準］を指定した場合のジョブの実行優先順位とジョブに設定する nice 値の対応を次の表に示します。

表 7-10 ジョブの実行優先順位とジョブに設定する nice 値（ジョブの実行先サービスに［標準］を指定した場合）

ジョブの 実行優先順位	ジョブに設定する nice 値	JP1/AJS3 サービスの nice 値を 「20」とした場合
1	nice 値+20	39
2	nice 値+10	30
3	nice 値	20
4	nice 値-10	10
5	nice 値-20	0

実行優先順位の指定がない場合，ジョブの実行優先順位は環境設定パラメーター`DEFAULTPRIORITY` で設定された値となり，JP1/AJS3 サービスの nice 値（デフォルト「20」）に増分値「20」を加算して，「39」の nice 値を設定します。これは，システムが制限する範囲（0～39）を超えて nice 値を設定しようとした際に最大値で 39，最小値で 0 に設定値が制限されるためです。

(b) ジョブの実行先サービスに［キューレス］を指定した場合

ジョブの実行先サービスに［キューレス］を指定した場合，JP1/AJS3 サービスの nice 値は基準となりません。nice 値には実行優先順位の低い順から 39，30，20，10，0（Linux の場合は，19，10，0，-10，-20）のどれかが設定されます。nice 値を変更する場合は，これらの nice 値に対応したジョブの実行優先順位を指定してください。

ジョブの実行先サービスに［キューレス］を指定した場合のジョブの実行優先順位とジョブに設定する nice 値の対応を次の表に示します。

表 7-11　ジョブの実行優先順位とジョブに設定する nice 値（ジョブの実行先サービスに［キューレス］を指定した場合）

ジョブの実行優先順位	ジョブに設定する nice 値※
1	39
2	30
3	20
4	10
5	0

注※

Linux の場合は，システムが制限する nice 値の範囲が-20～19 であり，実行優先順位の低い順から 19，10，0，-10，-20 のどれかが設定されます。

(2) ジョブの実行優先順位と JP1/AJS3 がジョブに設定する値（Windows の場合）

Windows の場合，ジョブの実行優先順位は 3 段階で設定します。ジョブの実行優先順位と Windows の基本優先度の対応を次の表に示します。基本優先度は Windows タスクマネージャーで参照することができます。

表 7-12　ジョブの実行優先順位と Windows の基本優先度

ジョブの実行優先順位	ジョブに設定する基本優先度
1	低
2	
3	通常
4	高
5	

JP1/AJS3 がジョブのプロセスを起動する際に，ジョブの実行優先順位から設定する優先順位クラスについて，次に説明します。

- 実行優先順位が［1］または［2］の場合は，システムがアイドル状態のときに実行されます。対話処理と比較して優先度が低くなります（Windows で規定される IDLE_PRIORITY_CLASS を設定します）。
- 実行優先順位が［3］の場合は，一般的なプロセスとして実行されます。対話処理と同等の優先度になります（Windows で規定される NORMAL_PRIORITY_CLASS を設定します）。
- 実行優先順位が［4］または［5］の場合は，上記の優先順位クラスを割り当てられたプロセスのスレッドより先に実行されます。対話処理と比較して優先度が高くなります（Windows で規定される HIGH_PRIORITY_CLASS を設定します）。

(3) ジョブの実行優先順位の変更方法

JP1/AJS3 を経由しないでサービスからプロセスを直接起動した場合，または端末から直接プロセスを起動した場合には，Windows のときには，基本優先度は「通常」で対話処理の優先順位になります。また，UNIX のときには，nice 値は「20」（Linux の場合は「0」）で実行されます。JP1/AJS3 から実行するジョブをこれらのプロセスと同等の実行優先順位で実行するには，ジョブの実行優先順位に「3」を設定する必要があります。ジョブの実行優先順位を「3」に変更する方法について説明します。

(a) キューレスジョブを含む PC ジョブ，UNIX ジョブ，HTTP 接続ジョブ，JP1/AJS3 上で実行する QUEUE ジョブ，およびカスタムジョブの場合

次に示すどちらかの方法でジョブの実行優先順位を変更できます。

1. 各ジョブの［詳細定義－［アイコン名]］ダイアログボックスの［定義］タブの［実行優先順位］に［3］を指定する。

2. 各ジョブの［詳細定義－［アイコン名]］ダイアログボックスの［定義］タブの［実行優先順位］には［なし］を指定し，ルートジョブネットなどの上位ジョブネットの［詳細定義－［ジョブネット]］ダイアログボックスの［定義］タブの［優先順位］に［3］を指定する。

(b) フレキシブルジョブ，アクションジョブの場合

次に示す方法でジョブの実行優先順位を変更できます。

1. ルートジョブネットなどの上位ジョブネットの［詳細定義－［ジョブネット]］ダイアログボックスの［定義］タブの［優先順位］に［3］を指定する。

(c) ジョブの実行優先順位および上位ジョブネットの優先順位に［なし］が設定されている場合

次に示す方法で，ジョブの実行優先順位に仮定する値を変更できます。

1. 環境設定パラメーター DEFAULTPRIORITY に［3］を設定する。

環境設定パラメーターの設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　4.2 環境設定パラメーターの設定」（Windows の場合），またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　14.2 環境設定パラメーターの設定」（UNIX の場合）を参照してください。

2. 各ジョブの［詳細定義－［アイコン名]］ダイアログボックスの［定義］タブの［実行優先順位]，およびルートジョブネットなどの上位ジョブネットの［詳細定義－［ジョブネット]］ダイアログボックスの［定義］タブの［優先順位］に，［なし］を指定する。

(d)　ジョブの実行優先順位を変更するときの注意事項

UNIX の場合，ジョブの実行ユーザーがスーパーユーザー権限を持たないとき，ジョブの実行優先順位に「4]，「5」を指定すると，ジョブの実行時に権限エラーとなるため，注意してください。

また，ジョブの実行優先順位をジョブプロセス以外の他プロセスより高く設定すると，「2.1.4(2) 実行優先順位」に記載している状態とは逆に JP1/AJS3 から起動したジョブプロセスが CPU を占有し，他プロセスに影響を与えることも考えられます。ジョブの実行優先順位を他プロセスより高く設定する場合は，システム環境や運用方法を十分考慮した上で設定してください。

7.10.3　ジョブの終了コードを確認する

ジョブの終了コードは，JP1/AJS3 - View の［モニタ詳細－［**アイコン名**]］ダイアログボックス，または`jpqjobget` コマンドで出力したジョブ情報の EXITCODE などで確認できます。ジョブの終了コードには，通常，正常終了では「0]，異常検出終了では起動したジョブプロセスの終了コードが設定されます。ただし，次の表に示す場合は，JP1/AJS3 によって終了コードが設定されます。

表 7-13　JP1/AJS3 によって終了コードが設定される条件とその終了コード

条件	JP1/AJS3 によって設定される終了コード
PC ジョブ，UNIX ジョブ，フレキシブルジョブ，HTTP 接続ジョブ，アクションジョブ，カスタムジョブ，または JP1/AJS3 をホストに指定した QUEUE ジョブで，次の状態になる場合。 • 起動失敗 • 強制終了（打ち切り時間指定による打ち切り終了も含みます） • 異常検出終了※	-1
UNIX ジョブで，JP1/AJS3 で予期しないエラーが発生した場合。	255

注※

ジョブプロセスが終了したあとの JP1/AJS3 側のあと処理で何か問題が発生した場合（例えば，マネージャーホストへの結果ファイルの転送失敗など）は，ジョブプロセスのリターンコードが JP1/AJS3 によって「-1」に上書きされ，異常検出終了になります。

注意事項

- PC ジョブおよび実行ホストが Windows のフレキシブルジョブの場合は，起動したジョブプロセス自身がリターンコードとして「-1」を返すこともできます。また，UNIX ジョブおよび実行ホストが Linux のフレキシブルジョブでは負の値で終了すると，終了コードは，256 + 負の値と扱われます。例えば，ジョブプロセスが「-1」で終了した場合，終了コードは，256 - 1 = 255 となります。したがって，ジョブプロセスによって設定された終了コードなのか，JP1/AJS3 によって設定された終了コードなのかを判別する必要がある場合は，ユーザーアプリケーションなどで設定する終了コードには「-1」および「255」を使用しないようにしてください。
- UNIX ジョブおよび実行ホストが Linux のフレキシブルジョブの場合は，ジョブの実行に先立ってログインスクリプトが実行されます。したがって，ログインスクリプト内で処理が終了した場合は，ログインスクリプトで設定された終了コードが設定されます。ログインスクリプト内で処理が異常終了することがないようにしてください。なお，ログインスクリプトの変更方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　13.4.2 ログインスクリプトの変更」を参照してください。
- フレキシブルジョブの場合に，終了判定方法で次のどれかを選択し，選択した終了条件が満たさなかったときは，終了コードは「1」になります。
  - ・[ファイルが存在すれば正常]
  - ・[ファイルが更新されれば正常]

ジョブの終了コードには，ユーザーアプリケーションや JP1/AJS3 以外に OS が設定するものがあります。代表的な終了コードを次の表に示します。これらの終了コードは OS のアップグレードなどによって変更になる場合もあるため，使用している OS の技術情報を確認してください。

表 7-14　OS が設定するジョブの終了コードの例

OS	終了コード	原因	対処方法
Windows	259 または-1	結果ファイルのオープンに失敗していることが考えられます。この場合は，同時に次に示すどちらかのメッセージが出力されます。 • 「KAVU3284-W 内部処理（**論理ホスト名**）でシステムコールエラーが発生しました（要因個所：**要因個所**，要因番号 1：0x2013000a，システムコール名：CreateFile，要因番号 2：**要因番号 2**）」 • 標準エラー出力に「プロセスはファイルにアクセスできません。別のプロセスが使用中です。」	ジョブとして実行するプログラム内でオープンするファイル，またはバッチファイル内のリダイレクションでオープンするファイルを，ジョブを登録するときの標準出力ファイル，または標準エラー出力ファイルとして指定しないでください。ただし，プログラム内で関数を使用してオープンする場合に限って，共有の読み取りまたは共有の書き込みを許可する設定でオープンすることによって回避することもできます。
	128	デスクトップヒープの領域不足が考えられます。この場合は，同時	デスクトップヒープ領域の消費を少なくしたい場合は，ジョブを実行するユーザーのアカウントを，「JP1/AJS3

OS	終了コード	原因	対処方法
Windows	128	に次に示すどちらかのメッセージが出力されます。 • 「XXXX.XXX - アプリケーションエラー：アプリケーションを正しく初期化できませんでした。」 • 「XXXX.XXX の初期化に失敗しました。」	サービス」のアカウントと同じにして運用してください。
UNIX	シグナル番号+128	ジョブプロセスから起動したプログラムがシグナルを受信したことが考えられます。※	次に示すどちらかの方法で対処してください。 • ジョブプロセスから起動するプログラムがシグナルを受信しないようにする。 • ジョブプロセスから起動するプログラムにシグナルハンドラーなどを設定して，シグナルの種類によって終了コードを切り分けるなどの処理を追加する。

注※

ジョブプロセス自身がシグナルを受信すると，実行先サービスが［通常］のジョブは，終了コード「-1」で終了状態「強制終了」になります。また，実行先サービスが［キューレス］のジョブは，終了コード「-1」で終了状態「異常検出終了」になります。

(1) フレキシブルジョブの終了コード

フレキシブルジョブを実行したときの終了コードを次の表に示します。

なお，フレキシブルジョブの一斉実行を使用している場合，複数ある宛先エージェントでの終了コードのうち，最も大きい値が終了コードになります。ただし，終了コードが-1 の宛先エージェントがあるとき，終了コードは-1 になります。

表 7-15 フレキシブルジョブの終了コード

終了コード	内容	対応するメッセージ
ユーザーアプリケーション（実行ファイル）が設定する終了コード	終了判定が「常に正常」,「常に異常」および「しきい値による判定」のとき，JP1/AJS3 の処理が正常に終了した。	KAVS8138-I KAVS8140-I
0	終了判定が「ファイルが存在すれば正常」または「ファイルが更新されれば正常」のとき，判定条件を満たした。	
1	終了判定が「ファイルが存在すれば正常」または「ファイルが更新されれば正常」のとき，判定条件を満たさなかった。	
-1	監視終了時間を超過した。	KAVS8101-E

終了コード	内容	対応するメッセージ
-1	リソースが不足している。	KAVS8102-E
	JP1/AJS3 Autonomous Agent Messenger サービスと通信不可。	KAVS8100-E
	システムエラー。	KAVV8106-E
	実行ファイルが存在しない。	KAVS8108-E
	実行ファイルに対して権限がない。	KAVS8109-E

(2) HTTP 接続ジョブの終了コード

HTTP 接続ジョブを実行したときの終了コードを次の表に示します。

表 7-16 HTTP 接続ジョブ実行時の終了コード一覧

終了コード	内容	対応するメッセージ
0	終了コードを指定している場合，正常終了（0 に指定した HTTP ステータスコード）。 終了コードを指定していない場合，正常終了（HTTP ステータスコード 200）。	KAVS8050-I
1～9	終了コードを指定している場合，正常終了で指定した HTTP ステータスコードに対応する終了コード。	KAVS8050-I
10	正常終了（終了コード指定に定義していない HTTP ステータスコードで，かつ HTTP ステータスコードが 200 以外）。	KAVS8050-I
20	タイムアウトが発生した。	KAVS8061-E
30	接続先ホストの名前解決に失敗した。	KAVS8054-E
40	ファイルアクセスでその他のエラーが発生した。	KAVS8053-E
60	パラメーターエラー。 ジョブの定義項目に誤りがある（不正な長さ，不正な値）。	KAVS8051-E
64	接続設定ファイルの構文エラー。	KAVS8052-E
70	URL の指定に誤りがある。 URL の参照先にリソースがない。 GET メソッド呼び出し時の送信情報ファイルに改行コードが含まれている。	KAVS8060-E
80	ファイルが存在しない。 ディレクトリを指定した。	KAVS8053-E
81	ファイルアクセスで権限エラーが発生した。 ディレクトリを指定した。	KAVS8053-E
90	環境設定パラメーターの読み込みに失敗した。	KAVS8063-E
101	接続先ホストへの接続に失敗した。	KAVS8056-E
102	接続先ホストとの通信でエラーが発生した。	KAVS8057-E

終了コード	内容	対応するメッセージ
106	レスポンスデータが接続設定ファイルで指定した最大ファイルサイズを超えている。	KAVS8059-E
110	HTTPS 認証の処理でエラーが発生した。 • HTTPS 認証に失敗 • ハンドシェイク失敗 • 証明書が不正 • 証明書ファイルのアクセスエラー • CRL ファイルが不正 • CRL ファイルのアクセスエラー • 発行元（CN（Common Name）または SAN（subjectAltName））のチェックに失敗 • 証明書の期限切れ • 証明書の失効	KAVS8062-E
120	メモリー不足が発生した。	KAVS0902-E
124	予期せぬエラーが発生した。 HTTP 接続ジョブの実行に必要なライブラリがロードできなかった。	KAVS8058-E KAVS0990-E

(3) イベントジョブおよびアクションジョブの終了コード

イベントジョブおよびアクションジョブが，実行しているとき（ジョブプロセスが起動されてから終了するまで）に設定される終了コードについては，「付録 A イベントジョブおよびアクションジョブ実行時の戻り値一覧」を参照してください。

7.11　スケジューラーサービスの起動性能について

スケジューラーサービスの起動性能は，保存世代数を管理したり，停止前の状態と起動モードに応じて状態遷移したりするため，主に実行登録したルートジョブネットやその配下のユニット数，世代数，およびスケジューラーサービス停止時のユニット状態に影響を受けます。そのため，スケジューラーサービスの起動性能を検証するには，運用に合わせてルートジョブネットを実行登録し，保存世代数まで世代を保存した状態で，実行中のジョブやジョブネットが多いピーク時に，スケジューラーサービスを再起動して，起動に掛かる時間を確認してください。

スケジューラーサービスの起動開始と起動完了は，統合トレースログに出力される対象スケジューラーサービスのメッセージ KNAD3705-I とメッセージ KNAD3602-I で確認できます。

ただし，起動条件付きジョブネットの世代が滞留し続けるなどで，検証時よりも世代数が多い場合は，検証した結果よりもスケジューラーサービスの起動に掛かる時間が増加します。

環境設定パラメーターCONDGENWARNINT や環境設定パラメーターMULTIMONWARNNUM を設定すると，起動条件付きジョブネットの世代が増え続けていることをメッセージ KAVS1157-W やメッセージ KAVS1156-W の出力によって確認できます。メッセージを出力しないで確認する場合は，次のようにajsshow コマンドを実行することで世代数を確認できます。

```
ajsshow -F スケジューラーサービス名 -g a -i "%JJ %## %CC" -ERT /
```

また，環境設定パラメーターFLSTARTPERFIMP を有効にすることでスケジューラーサービスの起動処理を改善できるため，メッセージの出力やajsshow コマンドによる確認と合わせて設定することを検討してください。環境設定パラメーターの詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.4.2 スケジューラーサービスの環境設定パラメーターの詳細」を参照してください。ajsshow コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajsshow」を参照してください。

補足事項

起動モードによって実行される処理が異なります。ホットスタートでは，保存世代数管理と状態遷移のための処理をするのに対し，ウォームスタートでは，さらにスケジュール再計算をします。また，ウォームスタートに対し，ディザスターリカバリースタートでは，さらに状態遷移がすべて完了してから起動完了します。そのため，使用する起動モードで起動に掛かる時間を確認してください。

8 定義内容の事前チェック

JP1/AJS3 には，ジョブの定義内容に誤りがないかをチェックする機能があります。

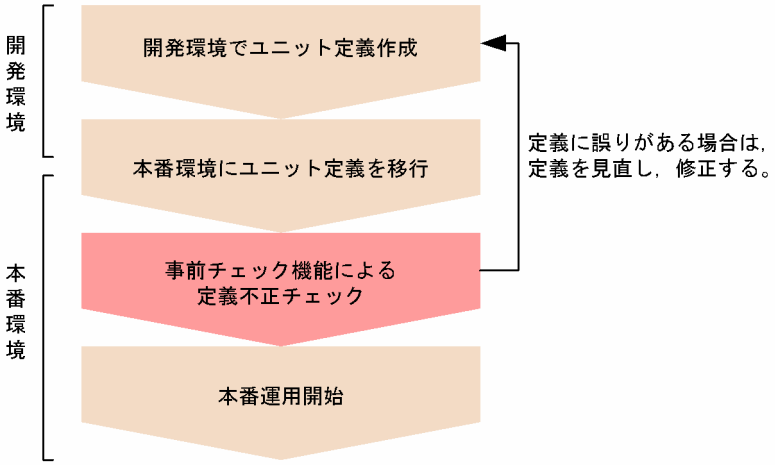
この章では，JP1/AJS3 本番運用前の定義内容チェックの流れやチェック項目，注意事項などについて説明します。

8.1　本番運用前の JP1/AJS3 定義内容のチェック

JP1/AJS3 では，本番運用時に障害の発生を抑えるため，ジョブの定義内容に不正がないかどうかを本番運用の開始前にチェックできます。

定義内容の事前チェックを実施する場合の運用までの流れを次の図に示します。

図 8-1　定義内容の事前チェックを実施する場合の運用までの流れ



定義内容の事前チェックは，ajschkdef コマンドで行います。ajschkdef コマンドを実行すると，定義内容の事前チェック結果格納ファイルにチェック結果が出力されるので，このファイルを確認してください。定義内容の事前チェック結果格納ファイルは，デフォルトでは次に示す場所に出力されます。

Windows で，インストール先フォルダがデフォルトまたはシステムで保護されたフォルダ配下の場合

%ALLUSERSPROFILE%¥Hitachi¥JP1¥JP1_DEFAULT¥JP1AJS2¥log¥ajscheckfile.txt

「%ALLUSERSPROFILE%」のデフォルトは「システムドライブ¥ProgramData」です。

「システムで保護されたフォルダ」とは，次のパスを指します。

・「システムドライブ¥Windows」配下

・「システムドライブ¥Program Files」配下

・「システムドライブ¥Program Files (x86)」配下

Windows で，インストール先フォルダが上記以外の場合

JP1/AJS3 - Manager のインストール先フォルダ¥log¥ajscheckfile.txt

UNIX の場合

/var/opt/jp1ajs2/log/ajscheckfile.txt

ajschkdef コマンドの詳細については，マニュアル「JP1/Automatic Job Management System 3 コマンドリファレンス　3. 通常の運用で使用するコマンド　ajschkdef」を参照してください。

なお，定義内容の事前チェックを行うには，定義内容の事前チェック機能をセットアップしておく必要があります。セットアップ方法，および環境の設定方法については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　6.5.1 JP1/AJS3 定義内容の事前チェック機能の設定」（Windows

の場合）またはマニュアル「JP1/Automatic Job Management System 3 構築ガイド　15.5.1 JP1/AJS3 定義内容の事前チェック機能の設定」（UNIX の場合）を参照してください。

8.1.1　定義内容の事前チェック機能の概要と注意事項

定義内容の事前チェック機能でチェックできる項目，事前チェックの流れ，および使用に当たっての注意事項を説明します。

(1)　チェック項目

定義内容の事前チェック機能でのチェック項目を次の表に示します。

表 8–1　定義内容の事前チェック機能でのチェック項目

チェック項目の分類	内容		
ジョブ実行順序	• 実行順序関係がループしていないかどうかをチェックします。 • 判定ジョブに従属ユニットが条件接続されているかどうかをチェックします。 • 従属ユニットに条件接続以外の関連定義がされていないかどうかをチェックします。		
ジョブネット詳細定義	ジョブネットコネクタ	接続範囲※1	接続範囲の指定が接続先のジョブネットと一致するかどうかをチェックします。
		接続ホスト名※1	指定したホストに接続できるかどうかをチェックします。
		接続サービス名	指定したスケジューラーサービスが存在するかどうかをチェックします。
		接続先のジョブネット名	次についてチェックします。 • 未指定ではないか。 • 指定しているユニットの種別がルートジョブネットまたはプランニンググループであるか。 • 指定したユニットが存在するか。 • 指定したユニットに，実行順序制御をすると設定されているか。 • 指定したユニットが他ホストを指定していないか。 • 指定したユニットがほかのスケジューラーサービスを指定していないか。

チェック項目の分類	内容		
ジョブネット詳細定義	ジョブネットコネクタ	接続先のジョブネット名	• 指定したユニットがほかのジョブネットコネクタを指定していないか。
	待ち合わせ条件	次についてチェックします。 • 指定した待ち合わせ対象ユニットがあるか。 • 指定した待ち合わせ対象ユニットのユニット種別は，待ち合わせ対象ユニットに指定できるユニット種別であるか。※2	
実行エージェント制限	マネージャー側で，実行エージェントが許可されているかどうかをチェックします。※3，※4，※5		
空ジョブ定義	UNIX ジョブで，［スクリプトファイル名］または［コマンド文］が指定されているかどうかをチェックします。 PC ジョブで，［実行ファイル名］が指定されているかどうかをチェックします。 JP1 イベント受信監視ジョブで，［イベント ID］が指定されているかどうかをチェックします。		
実行エージェント名	• UNIX ジョブ • PC ジョブ • アクションジョブ	実行先サービスが［標準］※6	次についてチェックします。 • マネージャー側で，実行エージェントが登録されているか。 • マネージャー側で，実行エージェントに設定されている実行ホスト名（エージェントホスト名）を解決できるか。 • エージェント側で，マネージャーのホスト名を解決できるか。※7
		実行先サービスが［キューレス］	次についてチェックします。 • マネージャー側で，エージェントのホスト名を解決できるか。 • エージェント側で，マネージャーのホスト名を解決できるか。※7
	イベントジョブ		次についてチェックします。 • マネージャー側で，実行エージェントが登録されているか。 • マネージャー側で，実行エージェントに設定されている実行ホスト名（エージェントホスト名）を解決できるか。 • エージェント側で，マネージャーのホスト名を解決できるか。※7

<table>
<tr><th>チェック項目の分類</th><th colspan="3">内容</th></tr>
<tr><td>実行エージェント名</td><td colspan="2">• ジョブネット※6
• カスタムジョブ※6
• フレキシブルジョブ※5，※6
• HTTP 接続ジョブ※6</td><td>次についてチェックします。
• マネージャー側で，実行エージェントが登録されているか。
• マネージャー側で，実行エージェントに設定されている実行ホスト名（エージェントホスト名）を解決できるか。
• エージェント側で，マネージャーのホスト名を解決できるか。※7</td></tr>
<tr><td>ユーザーマッピング</td><td colspan="3">エージェント側でユーザーマッピングが正しくできるかどうかについて，JP1 ユーザー，実行元ホスト，OS ユーザーを通常のジョブ実行時と同一の方法でチェックします。ただし，JP1 ユーザーにマッピングされている OS ユーザーの存在確認は行いません。
なお，実行ユーザー固定機能を使用している場合，ユニット属性プロファイルの設定内容に従って，固定されている JP1 ユーザーのユーザーマッピングをチェックします。※3，※5</td></tr>
<tr><td rowspan="6">ジョブ詳細定義</td><td rowspan="6">UNIX ジョブ</td><td>スクリプトファイル名※9</td><td>ファイルが Agent 上に存在するかチェックします。</td></tr>
<tr><td>環境変数ファイル名</td><td>1. ファイルが Agent 上に存在するかチェックします。
2. ファイルのアクセス権限をチェックします。</td></tr>
<tr><td>ワークパス</td><td>1. パスが Agent 上に存在するかチェックします。
2. パスのアクセス権限をチェックします。</td></tr>
<tr><td>標準入力ファイル名</td><td>1. ファイルが Agent 上に存在するかチェックします。
2. ファイルのアクセス権限をチェックします。</td></tr>
<tr><td>標準出力ファイル名</td><td>1. ファイル上位のディレクトリが Agent 上に存在するかチェックします。
2. ファイルが存在する場合はアクセス権限をチェックします。</td></tr>
<tr><td>標準エラー出力ファイル名</td><td>1. ファイル上位のディレクトリが Agent 上に存在するかチェックします。
2. ファイルが存在する場合はアクセス権限をチェックします。</td></tr>
</table>

チェック項目の分類	内容		
ジョブ詳細定義	UNIX ジョブ	実行時のユーザー	ユーザーマッピングでチェックします。
		転送元ファイル※9	1. ファイルが Manager 上に存在するかチェックします。 2. ファイルのアクセス権限をチェックします。
		転送先ファイル※9	1. ファイルが Agent 上に存在するかチェックします。 2. ファイルのアクセス権限をチェックします。
	PC ジョブ	実行ファイル名※9	ファイルが Agent 上に存在するかチェックします。
		環境変数ファイル名	ファイルが Agent 上に存在するかチェックします。
		ワークパス	パスが Agent 上に存在するかチェックします。
		標準入力ファイル名	ファイルが Agent 上に存在するかチェックします。
		標準出力ファイル名	ファイル上位のディレクトリが Agent 上に存在するかチェックします。
		標準エラー出力ファイル名	ファイル上位のディレクトリが Agent 上に存在するかチェックします。
		実行時のユーザー	ユーザーマッピングでチェックします。
		転送元ファイル※9	ファイルが Manager 上に存在するかチェックします。
		転送先ファイル※9	ファイルが Agent 上に存在するかチェックします。
	JP1 イベント受信監視ジョブ	イベント ID	イベント ID の形式に合致するかチェックします。
		実行前のイベント検索	検索前時間は範囲内かどうかチェックします。
	ファイル監視ジョブ	監視対象ファイル名	ファイル名の形式をチェックします。
	メール受信監視ジョブ	プラットフォーム	PC/UNIX が実行先ホストと一致しているかチェックします。

チェック項目の分類	内容		
ジョブ詳細定義	アクションジョブ共通	プラットフォーム	PC/UNIX が実行先ホストと一致しているかチェックします。※8
実行ファイル権限※9，※10，※11	ジョブ実行 OS ユーザーが該当する実行ファイルに対して実行権限および読み取り権限を持つかどうかを，通常のジョブ実行時と同一の方法でチェックします。ただし，UNIX 上で実行されるキューレスジョブについては，ジョブ実行 OS ユーザーがスーパーユーザー権限を持つ OS ユーザー(root ユーザー)の場合は実行権限のチェックはしません。 また，ジョブ実行 OS ユーザーのプライマリーグループでだけ実行ファイル権限のチェックを行います。		

注※1

［接続範囲］に［別サービス］を指定して接続先ホストで接続元制限機能を有効にしている場合は，接続先ホストのマネージャー用接続許可設定ファイルに，ジョブネットコネクタが定義されているホストの IP アドレスを設定する必要があります。設定しないで定義内容の事前チェックを実施した場合，チェック結果の「NUMBER OF CHECKUNITS」に表示される「チェックが完了したユニット数」が，ジョブネットコネクタの分だけ少なくなり，メッセージ KAVS3431-I が統合トレースログに出力されます。

注※2

ルートジョブネット，およびその配下のユニットを待ち合わせ対象ユニットに指定している場合は，起動条件の使用有無に関係なくチェックします。

プランニンググループ配下のネストジョブネットやジョブを待ち合わせ対象ユニットに指定している場合は，定義不正はチェックされません。

待ち合わせ条件による実行順序のループはチェックされません。

注※3

プロファイルの状態（有効/無効）に関係なく，`ajschkdef` コマンドで指定したプロファイルの内容に基づいてチェックします。

注※4

実行エージェントが実行エージェントプロファイルに設定されているかどうかをチェックします。このため，チェック対象のジョブが実行エージェントプロファイルでジョブ実行時に制限確認しない設定（`JobExec=off`）であっても，チェックします。

注※5

フレキシブルジョブの場合，エージェントを中継エージェントに読み替えてください。宛先エージェントはチェックしません。

注※6

実行エージェントグループ名が指定されている場合はチェックしません。

注※7

エージェントとマネージャー間の通信で NAT（ネットワークアドレス変換）を使用している場合は，マネージャーホスト名から解決された IP アドレスがエージェントホストとマネージャーホストで異な

るため，この項目のチェック結果はエラーになります。設定に問題がなければエラーを無視してください。

注※8

MSMQ 送信ジョブは UNIX 環境で実行できないため，プラットフォームの指定はできません。プラットフォームに「PC」が指定されたと仮定してチェックします。

注※9

相対パス，および UNC パスが指定された場合はチェックしません。

注※10

PC ジョブ（Windows ホストの場合）はチェックしません。

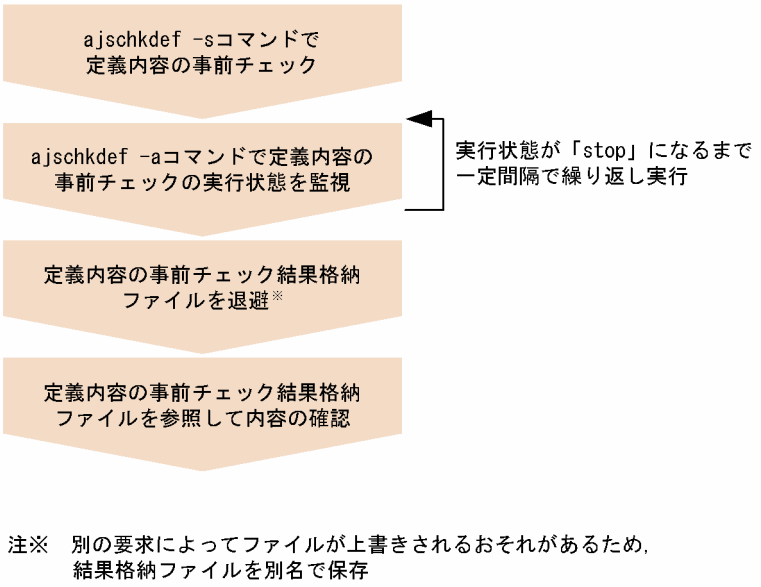
注※11

フレキシブルジョブはチェックしません。

(2) 事前チェックの流れ

定義内容の事前チェックの流れを次に示します。

図 8-2 定義内容の事前チェックの流れ



(3) 注意事項

定義内容の事前チェックを実施する際の注意事項を次に示します。

1. 定義内容の事前チェックは，本番業務が実行中でない時間帯に実行してください。本番業務の実行中に定義内容の事前チェックを実行した場合，次に示す現象が発生するおそれがあります。
   - スケジューラーデータベースへのアクセス競合によって，業務実行性能が劣化する
   - システムへの負荷が一時的に集中したことによって，ジョブ実行処理でエラーが発生する

2. 定義内容の事前チェックを実施するために，マネージャー側の JP1/AJS3 サービスを起動しておく必要があります。
3. 定義内容の事前チェックの多重実行はできません。
4. UNIX ジョブの［スクリプトファイル名］，または PC ジョブの［実行ファイル名］にパラメーターも同時に指定している場合，定義内容の事前チェックではエラーとして検出されます。
5. イベントジョブの引き継ぎ情報はチェックの対象外です。
6. マクロ変数はチェックの対象外です。
7. 環境変数はチェックの対象外です。
8. ジョブ定義で「$変数名$」形式の変数を使用した項目は，チェック対象外です。なお，定義内容の事前チェックでは，「$」を 2 回以上使用している場合，「$変数名$」形式の変数を使用していると見なします。
9. UNIX の場合，JP1/AJS3 Check Manager サービスは，services ファイルに設定されている「jp1ajs2chkman」サービスのポート番号を使用しません。
10. スクリプトファイルのアクセス権限のチェックでは，運用時とは異なる方法でチェックを実施するため，実際には正常終了するジョブを，定義内容の事前チェックでエラーとして検出する場合があります。
11. メッセージ KAVS3400～KAVS3431 は，「統合トレースログ」または「標準出力」だけに出力されます。
12. Windows で，環境変数ファイル，標準入力ファイル，標準出力ファイル，および標準エラー出力ファイルにフォルダを指定して，キューレスエージェントサービスで実行した場合には正常終了しますが，定義内容の事前チェックではエラーとして検出されます。
13. UNIX で，環境変数ファイルにディレクトリを指定して，キューレスエージェントサービスで実行した場合には異常終了しますが，定義内容の事前チェックではエラーとして検出されます。
14. ajschkdef コマンドで-A オプションを指定した場合，実行ファイル権限チェックにジョブ実行 OS ユーザーの情報が必要なため，「ユーザーマッピング」も前提としてチェックされます。ただし，Windows の場合，［詳細定義－［PC Job]］ダイアログボックスの［実行時のユーザー］に指定されたユーザーで，OS のログオン確認はチェックしません。
15. 次に示す条件が重なる場合は，一時ファイルを作成して定義内容の事前チェックを実施します。
    - UNIX ジョブの［標準出力ファイル名］，［標準エラー出力ファイル名］，または［転送先ファイル名］に実在しないファイルを指定する
    - 上記ジョブを対象に，-D オプションを指定して定義内容の事前チェックを実施する

    一時ファイルは，上記の条件に該当するジョブの［標準出力ファイル名］，［標準エラー出力ファイル名］，または［転送先ファイル名］で指定した場所に作成されます。そのため，［標準出力ファイル名］，［標準エラー出力ファイル名］，または［転送先ファイル名］で指定したファイルの上位ディレクトリは，更新日時が変わることがあります。

    なお，一時ファイルはチェック終了後に削除されます。
16. 次に示すユニットおよびその配下のユニットは，チェックしません。

- リモートジョブネット
- マネージャージョブネット
- マネージャージョブグループ

17. ジョブネットリリース機能を使用している場合は，リリース状態が「適用中」の定義内容について事前チェックを実施します。リリース状態が「リリース待ち」の定義内容はチェックできないため，リリース元ジョブネットをチェックしてください。
18. 接続元制限機能を有効にしていても，マネージャーホストが接続先のエージェントホストに接続できるかどうかは，チェックされません。そのため，接続が許可されていないエージェントホスト上で実行するようにユニットが定義されている場合でも，定義内容が正しければ，事前チェックではエラーは検出されません。ただし，［接続範囲］に［別サービス］を指定したジョブネットコネクタは，接続先ホストでジョブネットコネクタが定義されているホストの接続が許可されていないと，定義内容をチェックできません。
19. チェック対象のジョブの実行先ホスト名（エージェントホスト名）に，エイリアスホスト名は使用できません。エイリアスホスト名を使用すると，チェック結果が正しく出力されない場合があります。

(4) クラスタ運用時の注意事項

クラスタ運用している場合の，定義内容の事前チェックを実施する際の注意事項を次に示します。

1. クラスタ環境で，論理ホスト上のユニットに対して定義内容の事前チェックをする場合は，論理ホストの実行系でチェックを実施してください。待機系ではチェックできません。
2. クラスタ環境で，実行系の論理ホスト上のユニットをチェックしているときにフェールオーバーが発生した場合，チェックが中断されます。フェールオーバー後に実行系になった論理ホストでチェックが再開されることはありません。

付録

付録 A　イベントジョブおよびアクションジョブ実行時の戻り値一覧

イベントジョブおよびアクションジョブを実行したときの戻り値を次の表に示します。

表 A-1　イベントジョブ実行時の戻り値一覧

戻り値	説明	実行後または打ち切り後の ジョブの状態
0	正常終了しました。または，常に正常終了です。	正常終了
1	メッセージが一致して正常終了しました。	正常終了
2	詳細情報が一致して正常終了しました。	正常終了
3	メッセージが一致しないで正常終了しました。	正常終了
4	詳細情報が一致しないで正常終了しました。	正常終了
5	イベントの引き継ぎ情報が上限を超えました。	正常終了
6	イベントの引き継ぎ情報作成時にエラーが発生しました。	正常終了
7	打ち切り時間に到達後，正常終了しました。	正常終了
12	打ち切り時間に到達後，警告終了しました。	警告終了
16	条件が成立しませんでした（計画未実行終了）。	計画未実行終了
17	打ち切り時間に到達後，強制終了しました。	強制終了
18	強制終了しました。	強制終了
20	常に異常終了です。	異常終了
21	メッセージが一致して異常終了しました。	異常終了
22	詳細情報が一致して異常終了しました。	異常終了
23	メッセージが一致しないで異常終了しました。	異常終了
24	詳細情報が一致しないで異常終了しました。	異常終了
25	打ち切り時間に到達後，異常終了しました。	異常終了
30	イベント・アクションエージェントが停止したため，終了しました。	異常終了
50	マネージャー・エージェント間で通信エラーが発生しました。	異常終了，または起動失敗
55	リソースが不足しました。または，指定したファイルにアクセスできませんでした。	異常終了
56	メモリーが不足しました。	異常終了
64	ファイルが読み込めませんでした（I/O エラー）。	異常終了
90	パラメーターに誤りがありました。	異常終了
91	必要な項目が定義されていませんでした。	異常終了，または起動失敗

戻り値	説明	実行後または打ち切り後の ジョブの状態
92	マクロ変数の指定に誤りがありました。	異常終了
93	プラットフォームの指定に誤りがありました。	異常終了
250	メールシステムまたはメッセージキューシステムが使用できない状態でした。	起動失敗
253	前提プログラムでオプションエラーが発生しました。	異常終了
254	前提プログラムでエラーが発生しました。	異常終了
255	システムエラーが発生しました。	異常終了，または起動失敗

注

起動条件全体に対して操作した場合（起動条件を強制終了させた場合など）や，起動条件全体に関係するエラーが発生した場合は，起動条件全体の各イベントの戻り値には，「0（ゼロ）」が設定されます。これに対して，特定のイベントに対するエラーが発生した場合（定義が誤っていてエラーになった場合など）は，表に示した戻り値が設定されます。

表 A-2　JP1 イベント送信ジョブ実行時の戻り値一覧

戻り値	説明
0	正常終了しました。
1	引数エラー
100	転送に失敗しました。 イベントの到達確認をするときだけ返されます。JP1/Base の`jevsendd` コマンドの戻り値が，3（転送に失敗した）の場合に対応しています。
120	処理は続行中です（最大到達待ち時間内に到達確認ができなかった場合）。 イベントの到達確認をするときだけ返されます。JP1/Base の`jevsendd` コマンドの戻り値が，2（処理は続行中）の場合に対応しています。
150	メモリー不足
255	その他のエラー

注

戻り値 0 以外の場合は，ジョブの実行結果詳細のメッセージで要因を特定してください。

表 A-3　メール送信ジョブ実行時の戻り値一覧（Windows）

戻り値	説明
0	正常終了しました。
90	パラメーターエラー（定義項目エラー）
91	パラメーターエラー（定義内容エラー）

戻り値	説明
92	パラメーターエラー（必要な項目の未定義エラー）
94	パラメーターエラー（定義内容長エラー）
95	本文テキストファイルなし，添付ファイルなし，または添付ファイルリストなし
99	JP1/AJS3 メール監視プロセス未起動エラーまたは JP1/AJS3 Mail サービス未起動エラー
100	環境設定読み込みエラー
101	環境設定未定義エラー
102	JP1/AJS のインストール失敗
103	環境設定の定義情報が壊れています。
104	Outlook を使用しないメール送信ジョブ実行時に，存在しないプロファイルを指定
105	Outlook を使用しないメール送信ジョブ実行時に，プロファイルへのアクセスに失敗
120	JP1/AJS のメール監視プロセスまたはメール監視サービスへの通信に失敗しました。
121	メール情報の送信に失敗しました。
150	メモリー不足
151	リソース不足（ファイル関連）
152	リソース不足（プロセス関連）
153	メール送信ジョブがタイムアウトしました。
154	プロファイルが破損しています。
155	デスクトップヒープ不足発生
180	メール送信時スレッドの終了時終了状態の取得に失敗しました。
182	メールの送信に失敗しました。
255	システムエラーが発生しました。

表 A-4　メール送信ジョブ実行時の戻り値一覧（UNIX）

戻り値	説明
0	正常終了しました。
90	パラメーターエラー（定義項目エラー）
95	本文テキストファイル名がありません。
96	メール送信先アドレス指定なしエラー
97	メール送信先アドレス指定エラー
98	本文テキストファイルの読み込みに失敗
126	メール送信機能は使用できません。

戻り値	説明
130	メールを送信するためのテンポラリーファイルの作成に失敗しました。
131	メールを送信中にシグナルを受けメールの送信を中断しました。
150	メモリー不足
254	メールを送信時に使用するシステムコマンドに実行権限がありません。
255	システムエラーが発生しました。

表 A-5　メッセージキュー送信ジョブ実行時の戻り値一覧（Windows）

戻り値	説明
0	正常終了しました。
1	指定キューへの接続が部分的に失敗
3	デッドレターキューへ送信しました。
90	パラメーターエラー（定義項目エラー）
91	パラメーターエラー（定義内容エラー）
92	パラメーターエラー（必要な項目の未定義エラー）
94	パラメーターエラー（定義内容長エラー）
95	指定したファイルなし
98	JP1/AJS2 メッセージキュー監視プロセスが使用不可
99	JP1/AJS2 メッセージキュー監視プロセス未起動エラー
100	環境設定読み込みエラー
101	環境設定未定義エラー
102	JP1/AJS2 のインストール失敗
103	環境設定の使用するメッセージキューイングシステムの種別に誤りがあります。
120	JP1/AJS2 の提供している TP1/LiNK の SUP との通信に失敗しました。
121	指定キューへの接続に失敗しました。
123	メッセージキューアクセス時のエラー
150	リソース不足（プロセス関連）
151	リソース不足（ファイル関連）
152	リソース不足（スレッド関連）
153	ミューテックスの作成に失敗
154	jposupwth（TP1/Message Queue 連携時の MQ 監視用プロセス）が停止しました。
155	jposupwth（TP1/Message Queue 連携時の MQ 監視用プロセス）が起動失敗

戻り値	説明
156	イベントの作成に失敗
157	ファイルの作成に失敗しました。
158	メモリー不足
159	デッドレターキューへの送信に失敗
180	メッセージ送信時スレッドの終了時終了状態の取得に失敗しました。
181	ファイルサイズの取得に失敗しました。
182	ファイルの読み込みに失敗しました。
183	ファイルのクローズに失敗しました。
250	MQSeries サービスの起動時に別のイベントが発生
255	システムエラーが発生しました。

表 A-6　メッセージキュー送信ジョブ実行時の戻り値一覧（UNIX）

戻り値	説明
0	正常終了しました。
2	送信中に警告エラー
90	パラメーターエラー（定義項目エラー）
91	パラメーターエラー（定義内容エラー）
95	指定したファイルなし
103	環境設定の使用するメッセージキューイングシステムの種別に誤りがあります。
120	JP1/AJS2 の提供している TP1/Server Base の SUP との通信に失敗しました。
123	メッセージキューアクセス時のエラー
126	メッセージキュー送信処理は使用できません。
150	リソース不足（プロセス関連）
155	TP1/Message Queue 連携時の MQ 監視用プロセスが起動失敗
156	イベントの作成に失敗
157	ファイルの作成に失敗しました。
158	メモリー不足
255	システムエラーが発生しました。

表 A-7　MSMQ 送信ジョブ実行時の戻り値一覧

戻り値	説明
0	正常終了しました。

戻り値	説明
90	パラメーターエラー（定義項目エラー）
91	パラメーターエラー（オペランドエラー）
92	パラメーターエラー（定義内容エラー）
93	パラメーターエラー（必要な項目の未定義エラー）
94	メッセージ送信失敗
95	致命的なエラー MSMQ 連携機能は続行できない

表 A-8　JP1/Cm2 状態通知ジョブ実行時の戻り値一覧

戻り値	説明
0	正常終了しました。
16	パラメーターエラー（定義項目エラー）
17	パラメーターエラー（オペランドエラー）
18	パラメーターエラー（定義内容エラー）
19	パラメーターエラー（必要な項目の未定義エラー）
30	SNMP サービスがインストールされていません（Windows 限定）。
31	SNMP サービスが起動されていません（Windows 限定）。
32	SNMP サービスのコミュニティ名が設定されていません。
33	SNMP サービスのトラップ送信先が設定されていません。
34	状態通知に失敗しました。

表 A-9　ローカル電源制御ジョブ実行時の戻り値一覧

戻り値	説明
0	正常終了しました。
0 以外	ローカル電源制御に失敗しました。

表 A-10　リモート電源制御ジョブ実行時の戻り値一覧

戻り値	説明
0	正常終了しました。
0 以外	リモート電源制御に失敗しました。

表 A-11　（特記事項）アクションジョブ実行時の戻り値一覧※1

戻り値	説明
128	リソース不足※2
-1	起動失敗，強制終了，異常検出終了※3

注※1

アクションジョブのジョブプロセスが設定する戻り値以外で設定されるおそれがある戻り値です。

なお，アクションジョブは，次のジョブが該当します。

- JP1 イベント送信ジョブ
- メール送信ジョブ
- メッセージキュー送信ジョブ
- MSMQ 送信ジョブ
- JP1/Cm2 状態通知ジョブ
- ローカル電源制御ジョブ
- リモート電源制御ジョブ

注※2

この戻り値は，Windows 限定です。詳細については，「7.3 PC ジョブ使用時の注意事項」を参照してください。

なお，この戻り値は OS で設定される値のため，OS のバージョンによって戻り値が異なるおそれがあります。

注※3

詳細については，「7.10.3 ジョブの終了コードを確認する」を参照してください。

付録 B　イベントジョブで引き継ぐ情報

イベントジョブで引き継ぐ情報について説明します。

イベントジョブで受信した情報を後続のジョブまたはジョブネットで参照できます。[詳細定義－［**イベントアイコン名**］－［引き継ぎ情報]］ダイアログボックスで，受信したイベントの特定の引き継ぎ情報をマクロ変数に指定すると参照できます。マクロ変数に指定できる引き継ぎ情報は，各イベントジョブで異なります。マクロ変数の定義例については，「2.4.4(6) イベントジョブの受信情報の引き継ぎ」を参照してください。

各イベントジョブの引き継ぎ情報の一覧を，次の表に示します。

表 B-1　各イベントジョブの引き継ぎ情報一覧

アイコン名	変数	説明	Windows	UNIX	サイズ（バイト）
共通	CMTMOUT※1	ジョブが打ち切り終了したかどうか t：打ち切り終了 f：打ち切り終了以外	○	○	1
JP1 イベント受信監視	EVID	イベント ID（基本コード：拡張コード※2） イベント ID を「基本コード：拡張コード」形式にした文字列	○	○	17
	EVUSR	イベント発行元ユーザー名	○	○	20
	EVGRP	イベント発行元グループ名	○	○	20
	EVHOST	発行元イベントサーバ名	○	○	255
	EVIPADDR	イベント発行元 IP アドレス	○	○	39
	EVMSG	メッセージ情報	○	○	1,023
	EVDETAIL	イベント詳細情報※3	○	○	1,023
	EVSEV	拡張属性の重大度	○	○	11
	EV：**拡張属性名**※4	任意の拡張属性	○	○	4,085
	EVENV1～EVENV9※5	切り出したデータ	○※6	○	4,085
	EVUSRID	イベント発行元ユーザー ID	○	○	10
	EVGRPID	イベント発行元グループ ID	○	○	10
	EVPROCESSID	イベント発行元プロセス ID	○	○	10
	EVDATE	イベント発行年月日（yyyy/mm/dd 形式）	○	○	10

アイコン名	変数	説明	Windows	UNIX	サイズ（バイト）
JP1 イベント受信監視	EVTIME	イベント発行時間（hh：mm：ss 形式）	○	○	8
ファイル監視	FLFNAME	変更があったファイルのファイル名（完全名で指定）	○	○	Windows：258 UNIX：509
	FLCOND	次の成立した監視条件のどれか一つ c（作成） d（削除） s（サイズ変更） m（最終書き込み時間）	○	○	1
	FLCTIME	ファイル更新時刻（yyyy/mm/dd. hh：mm：ss 形式）	○	○	19
	FLSIZE	変更があったファイルのファイルサイズ	○	○	20
メール受信監視	MLRCVADDRESS	受信したメールの差し出し人※7	○	○	256※8
	MLRCVSUBJECT	受信したメールの件名	○	○	256※9
	MLRCVBODY	受信したメールの本文を保存したファイル名	○	×	258
	MLRCVATTACHFILEnn（nn は 01～20 のアラビア数字）	受信したメールに添付されていた添付ファイル名	○	×	258※10
	MLRCVATTACHLIST	受信したメールに添付されていたファイルの一覧のファイル名	○	×	258※11
	MLRCVMAILBODY	受信したメール自体を保存したファイル名（内部設定，1 ファイル）	×	○	452
	MLRCVTIME	受信したメールの到着日時	○	○	40
メッセージキュー受信監視	MQRCVCORRELATION	受信したメッセージの相関識別子	○	○	24
	MQRCVDISCRIMINATION	受信したメッセージのメッセージ識別子	○	○	24
	MQRCVQUEUE	メッセージ受信キュー名	○	○	48
	MQRCVMODELQUEUE	モデルキュー名	○	○	48

アイコン名	変数	説明	Windows	UNIX	サイズ（バイト）
メッセージキュー受信監視	MQRCVMESSAGEFILE	受信したメッセージをメッセージ構造体として格納したファイル名（内部設定，1 ファイル）	○	○	Windows：258 UNIX：452
MSMQ 受信監視	MSRCVQUEUEPATH	受信したメッセージのキューパス名	○	×	259
	MSRCVMUTUAL	受信したメッセージの相互関係	○	×	20
	MSRCVMESSAGELABEL	受信したメッセージのメッセージラベル	○	×	249
	MSRCVAPPLICATION	受信したメッセージのアプリケーション情報（HEX 形式の数値文字）	○	×	8
	MSRCVMESSAGEFILE	受信したメッセージをメッセージ構造体として格納したファイル名（内部設定，1 ファイル）	○	×	258
ログファイル監視	LFFNAME	トラップしたログファイルのファイル完全名	○	○	258
	LFDATA	トラップしたログデータ（511 バイトを超えた場合は切り捨てる）	○	○	511
Windows イベントログ監視	NELOG	ログ種別	○	×	255
	NEEVKIND	イベントの種類	○	×	20
	NESOURCE	ソース	○	×	255
	NECLASS	分類	○	×	4,085
	NEEVID	イベント ID	○	×	16
	NEDETAIL	説明（1,023 バイトを超えた場合は切り捨てる）	○	×	1,023
実行間隔制御	－	－	－	－	－

（凡例）

○：指定できる。

×：指定できない。

－：該当しない。

注 1

引き継ぎ情報に次の文字が含まれていた場合，すべて空白に置き換えられます。

「¥r」「¥n」「¥b」「¥f」

注 2

マクロ変数名と引き継ぎ情報のサイズの合計が 4,096 バイトを超えないように，事前にマクロ変数名のサイズや引き継ぎ情報のサイズを見積もった上で運用してください。特に，起動条件を AND 条件で定義している場合は，起動条件内に定義したすべてのイベントジョブのマクロ変数名および引き継ぎ情報をマージします。このため，マクロ変数名と引き継ぎ情報のサイズの合計が 4,096 バイトを超えないように注意してください。

注※1

イベントジョブが監視条件によって正常終了した場合，CMTMOUT へ"f"を格納します。打ち切り時間が経過した場合，CMTMOUT へ"t"を格納します。エラー発生によってイベントジョブが異常検出終了した場合，CMTMOUT へは何も格納しません。

注※2

拡張コードは，必ず「0」が設定されます。

注※3

イベント詳細情報とは，JP1 イベントの基本属性の詳細情報を指し，JP1 イベント中の詳細情報がテキストの場合だけ設定されます。JP1 イベントの属性については，マニュアル「JP1/Automatic Job Management System 3 運用ガイド　付録 A.2 JP1 イベントの属性」を参照してください。

注※4

拡張属性名は，JP1/Base のイベントサービスで決められた 32 バイト以内の文字列です。拡張属性名の詳細については，マニュアル「JP1/Base 運用ガイド」を参照してください。

注※5

切り出しデータは，ajsdefine コマンドに指定したパラメーターの指定順ではなく，次の順番で正規表現のチェックをして，切り出しがあれば，順に EVENV1～EVENV9 へ格納します。

- イベント発行元ユーザー名
- イベント発行元グループ名
- 発行元イベントサーバ名
- メッセージ情報
- 詳細情報
- 任意の拡張属性

任意の拡張属性内では，JP1 イベントの取得順に正規表現のチェックをして，切り出しがあれば，順に EVENV1～EVENV9 へ格納します。

注※6

拡張正規表現を使用している場合だけ設定されます。

注※7

Windows の場合，環境設定パラメーターNextAddress に指定した情報の種類が設定されます。「Address」を指定した場合には，メールアドレスが設定されます。「Nickname」を指定した場合には，表示名（ニックネーム）が設定されます。

Windows ホストでメールシステムと連携するために必要な設定については，マニュアル「JP1/Automatic Job Management System 3 連携ガイド　2.3.4 メールシステム連携のための環境設定をする」を参照してください。

注※8

メール受信監視ジョブを実行するホストが Windows の場合，バイト数に加えて文字数にも上限があります。文字数の上限は 255 文字です。

注※9

メール受信監視ジョブを実行するホストが Windows の場合，バイト数に加えて文字数にも上限があります。文字数の上限は 253 文字です。

注※10

受信したメールに添付されていたファイルを保存して，そのファイル名がフルパス名で設定されます。

複数のファイルが添付されていた場合，添付ファイルの順序は保証されません。また，21 個以上のファイルが添付されていた場合には，20 個のファイル名までしか引き継ぎできません。

注※11

受信したメールに添付されていたファイルを保存して，その保存したファイルのリストを作成します。作成されたリストファイル名がフルパス名で設定されます。

付録 C　HTTP 接続ジョブで使用するファイル

HTTP 接続ジョブで使用するファイルについて説明します。

付録 C.1　接続設定ファイル

接続設定ファイルには，Web API を呼び出すために必要な HTTP 接続に関する情報を設定します。

ファイルの内容にセキュリティ情報が含まれる場合は，ユーザー責任で管理するようにしてください。

接続設定ファイルの仕様を次の表に示します。

表 C-1　接続設定ファイルの仕様

項番	項目	説明
1	権限	ジョブ実行ユーザーにファイル読み取り権限が必要
2	文字コード	HTTP 接続ジョブを実行するエージェントホストで使用している文字コードと同じ文字コードで記載
3	最大サイズ	1,024 バイト

接続設定ファイルにはモデルファイルがあります。接続設定ファイルは，モデルファイルをコピーして編集するか，または記述形式に従って作成してください。

接続設定ファイルのモデルファイルの格納先を次に示します。

Windows の場合

JP1/AJS3 - Manager または JP1/AJS3 - Agent のインストール先フォルダ
¥conf¥ajshttpreq_network.conf.model

Linux の場合

/etc/opt/jp1ajs2/conf/ajshttpreq_network.conf.model

なお，接続設定ファイルは，HTTP 接続ジョブを実行するエージェントホスト上の任意の場所に格納してください。

(1)　記述形式

接続設定ファイルの記述形式を次に示します。

```
URL=接続先URL
HttpVer=HTTPバージョン
ConnectTimeout=接続タイムアウト時間
Timeout=HTTP接続ジョブ全体の処理時間の上限
MaximumReceivedDataSize=最大受信データサイズ
ClientBindAddrIP=クライアントバインドを行うIPアドレス
```

```
Header=追加ヘッダー
Authorization=WebサーバのBasic認証の有無
UserName=WebサーバのBasic認証に使用するユーザー名
UserPass=WebサーバのBasic認証に使用するパスワード
Proxy=プロキシサーバ名
ProxyAuthorization=プロキシサーバの認証の有無
ProxyUserName=プロキシサーバの認証に使用するユーザー名
ProxyUserPass=プロキシサーバの認証に使用するパスワード
CAFile=証明書ファイル名
CrlFile=証明書失効リストファイル名
```

コメントを記載する際は，行の先頭に「#」を記載してください。半角の空白文字やタブ文字で始まる行のうち，半角の空白文字やタブ文字の次の文字が「#」の行も，コメントとして扱われます。

(2) 記述項目

記述項目について説明します。接続設定ファイルは，各項目を改行文字で区切って記述してください。

(a) URL=接続先 URL

接続先の URL を，2,083 バイト以内の文字列で指定します。URL は，「http://」または「https://」で始まる必要があります。シングルバイト文字で指定する必要があります。

接続先 URL に，URL パラメーターを含めた指定はできません。接続先 URL に URL パラメーターを含めたい場合は，URL パラメーターを送信情報ファイルに指定してください。

規定のポート番号以外を使用する場合は，URL に含めた形式で記載が必要です。規定のポート番号を次に示します。

- http の場合：80
- https の場合：443

https の場合は「CAFile=証明書ファイル名」を指定する必要があります。

この項目は必須です。

(b) HttpVer=HTTP バージョン

HTTP のバージョンを次のどちらかから指定します。

1.0
　HTTP1.0 を使用してリクエストします。

1.1
　HTTP1.1 を使用してリクエストします。

この項目は省略できます。省略した場合，「1.1」が仮定されます。

(c) ConnectTimeout=接続タイムアウト時間

接続先ホストへの接続タイムアウト時間を，1～600（単位：秒）で指定します。

ここで指定した接続タイムアウト時間内に接続先ホストと通信できない場合，HTTP 接続ジョブは異常終了します。

この項目は省略できます。省略した場合，「10」が仮定されます。

注意事項

HTTP 接続ジョブが実行されるホストの OS の設定，およびシステムのネットワーク設定によって，この項目で指定した値より短い時間でタイムアウトし，HTTP 接続ジョブが異常終了する場合があります。OS の設定については，OS のドキュメントを参照してください。

(d) Timeout=HTTP 接続ジョブ全体の処理時間の上限

HTTP 接続ジョブ全体の処理時間の上限を，1～86,400（単位：秒）で指定します。

ここで指定した処理時間内に処理が終了しない場合，HTTP 接続ジョブは異常終了します。

この項目は省略できます。省略した場合，「600」が仮定されます。

(e) MaximumReceivedDataSize=最大受信データサイズ

Web API から取得した受信データの最大受信データサイズを，1～1,024（単位：メガバイト）で指定します。

指定した値は，受信した HTTP ヘッダーの「Content-Length」の値と比較されます。HTTP ヘッダーの「Content-Length」の方が大きい場合，データの受信を中断し，HTTP 接続ジョブは異常終了します。

受信した HTTP ヘッダーに「Content-Length」が指定されていない場合，データを受信しながらデータサイズをチェックします。HTTP ボディのデータサイズが指定した値に達した時点でデータの受信を中断し，HTTP 接続ジョブは異常終了します。

この項目は省略できます。省略した場合，「500」が仮定されます。

(f) ClientBindAddrIP=クライアントバインドを行う IP アドレス

クライアントバインドを行う場合，バインドする IP アドレスを 39 バイト以内の文字列で指定します。

IP アドレスは，IPv4 アドレスまたは IPv6 アドレスで指定できます。

この項目は省略できます。省略した場合，クライアントバインドを行いません。

注意事項

ファイアウォール通過用に送信時の IP アドレスを特定する場合，または特定の LAN だけを利用する場合は，クライアントバインドを行ってください。

(g) Header=追加ヘッダー

リクエスト時の HTTP ヘッダーに追加でヘッダーを設定する場合，追加するヘッダーを，1,024 バイト以内の文字列で指定します。

追加ヘッダーは複数行指定できます。HTTP 接続ジョブが設定する項目と同じ項目を設定した場合，「Header=追加ヘッダー」に指定された値が優先されます。

「Header=追加ヘッダー」の指定形式を次に示します。

Header=ヘッダー名:ヘッダー値

値を持ったカスタムヘッダーを指定する場合，ヘッダー名とヘッダー値の間に「：(コロン)」を指定します。例えば，ヘッダー名が「ABC」，ヘッダー値が「DEF」というヘッダーを指定したい場合は，「Header=ABC:DEF」と設定します。

Header=ヘッダー名:

HTTP 接続ジョブが設定するヘッダーを無効にしたい場合，ヘッダー名の終端に「：(コロン)」を指定します。例えば，「ABC」というヘッダーを無効にしたい場合は，「Header=ABC:」と設定します。

Header=ヘッダー名;

値を持たないカスタムヘッダーを設定する場合，ヘッダー名の終端に「；(セミコロン)」を指定します。例えば，「ABC」という値を持たないヘッダーを指定したい場合は，「Header=ABC;」と設定します。

この項目は省略できます。省略した場合，ヘッダーを追加しません。

注意事項

HTTP 接続ジョブが設定するヘッダーを無効にすると，HTTP 接続ジョブが正常に実行されなくなるおそれがあります。十分に検討してから実施してください。

(h) Authorization=Web サーバの Basic 認証の有無

Web サーバで Basic 認証を行うかどうかを，次のどちらかから指定します。

y

Web サーバで Basic 認証を行います。

n

Web サーバで Basic 認証を行いません。

「y」を指定した場合，「UserName=Web サーバの Basic 認証に使用するユーザー名」と「UserPass=Web サーバの Basic 認証に使用するパスワード」に指定された値を基に，リクエスト時の HTTP ヘッダーに Authorization ヘッダーを付与します。なお，「Header=追加ヘッダー」で Authorization ヘッダーを付与している場合，「Header=追加ヘッダー」に付与された値が有効になります。

Basic 認証以外の認証方式を使用する場合は，「Header=追加ヘッダー」で指定する必要があります。

接続先が Web サーバでの認証を必要としない場合に，「y」を指定したときの動作は，接続先の Web サーバの仕様に依存します。

この項目は省略できます。省略した場合，「n」が仮定されます。

(i)　UserName=Web サーバの Basic 認証に使用するユーザー名

Web サーバで Basic 認証を行う場合，認証に使用するユーザー名を 256 バイト以内の文字列で指定します。ユーザー名に「：(コロン)」は使用できません。

Web サーバで Basic 認証を行う場合，この項目は必須です。

(j)　UserPass=Web サーバの Basic 認証に使用するパスワード

Web サーバで Basic 認証を行う場合，認証に使用するパスワードを 256 バイト以内の文字列で指定します。

Web サーバで Basic 認証を行う場合，この項目は必須です。

(k)　Proxy=プロキシサーバ名

プロキシサーバを利用する場合，プロキシサーバ名を 2,083 バイト以内の文字列で指定します。

プロキシサーバで Basic 認証を行う場合，この項目は必須です。

(l)　ProxyAuthorization=プロキシサーバの Basic 認証の有無

プロキシサーバを利用する場合，プロキシサーバで Basic 認証を行うかどうかを，次のどちらかから指定します。

y
　プロキシサーバで Basic 認証を行います。

n
　プロキシサーバで Basic 認証を行いません。

「y」を指定した場合，「ProxyUserName=プロキシサーバの Basic 認証に使用するユーザー名」と「ProxyUserPass=プロキシサーバの Basic 認証に使用するパスワード」に指定された値を基に，リクエスト時の HTTP ヘッダーに ProxyAuthorization ヘッダーを付与します。なお，「Header=追加ヘッダー」で ProxyAuthorization ヘッダーを付与している場合，「Header=追加ヘッダー」に付与された値が有効になります。

Basic 認証以外の認証方式を使用する場合は，「Header=追加ヘッダー」で指定する必要があります。

接続先がプロキシサーバでの認証を必要としない場合に，「y」を指定したときの動作は，接続先のプロキシサーバの仕様に依存します。

この項目は省略できます。省略した場合，「n」が仮定されます。

(m)　ProxyUserName=プロキシサーバの Basic 認証に使用するユーザー名

プロキシサーバで Basic 認証を行う場合，認証に使用するユーザー名を 256 バイト以内の文字列で指定します。ユーザー名に「：(コロン)」は使用できません。

プロキシサーバで Basic 認証を行う場合，この項目は必須です。

(n)　ProxyUserPass=プロキシサーバの Basic 認証に使用するパスワード

プロキシサーバで Basic 認証を行う場合，認証に使用するパスワードを 256 バイト以内の文字列で指定します。

プロキシサーバで Basic 認証を行う場合，この項目は必須です。

(o)　CAFile=証明書ファイル名

「URL=接続先 URL」に「https://」で始まる文字列を指定した場合に，ルート証明書のファイル名を 511 バイト以内の文字列で指定します。ルート証明書は PEM 形式である必要があります。DER 形式を指定した場合，メッセージ KAVS8062-E が出力され，HTTP 接続ジョブは異常終了します。

「URL=接続先 URL」に「http://」で始まる文字列を指定した場合，この項目は省略できます。

(p)　CrlFile=証明書失効リストファイル名

サーバ証明書を検証する場合，証明書失効リストファイル名を 511 バイト以内の文字列で指定します。証明書失効リストファイルは PEM 形式である必要があります。DER 形式を指定した場合，メッセージ KAVS8062-E が出力され，HTTP 接続ジョブは異常終了します。

この項目は省略できます。省略した場合，証明書の失効状況のチェックは行われません。

(3)　モデルファイルの形式

接続設定ファイルのモデルファイルの形式を次に示します。

```
URL=
#HttpVer=1.1
#ConnectTimeout=10
#Timeout=600
#MaximumReceivedDataSize=500
#ClientBindAddrIP=
#Header=
#Authorization=n
#UserName=
#UserPass=
#Proxy=
#ProxyAuthorization=n
#ProxyUserName=
#ProxyUserPass=
#CAFile=
#CrlFile=
```

省略できる項目はコメント行になっています。任意の値を指定する場合は，指定する項目の「#」を削除してから，任意の値を指定してください。

(4)　注意事項

- 次の行は無視されます。
  - 空行
  - 半角の空白文字およびタブ文字だけの行
- 行の先頭または最後に半角の空白文字やタブ文字がある場合，半角の空白文字やタブ文字を除いた値を使用します。
- 同じ項目を複数回指定した場合は，「Header=追加ヘッダー」を除き，最後に指定した内容が有効になります。
- 指定されている記述項目ではない項目がある場合，HTTP 接続ジョブは異常終了します。
- 項目名を記述して，項目の値だけを省略した場合，HTTP 接続ジョブは異常終了します。

付録 C.2　送信情報ファイル

送信情報ファイルには，Web API へのリクエスト情報を設定します。ファイルの記述形式や記述項目は呼び出す Web API の仕様によって異なります。また，リクエスト時に送信する情報がない場合は，送信情報ファイルの作成は不要です。

ファイルの内容にセキュリティ情報が含まれる場合は，ユーザー責任で管理するようにしてください。

JP1/AJS3 - Manager と JP1/AJS3 - Agent のバージョンが 12-50 以降で，拡張モードを使用する場合，このファイルは使用されません。拡張モードについては，「2.4.12 Web 上の業務システムと連携する（HTTP 接続ジョブを使ったジョブネットの定義例）」を参照してください。

送信情報ファイルの仕様を次の表に示します。

表 C-2　送信情報ファイルの仕様

項番	項目	説明
1	権限	ジョブ実行ユーザーにファイル読み取り権限が必要
2	文字コード	接続設定ファイルの Header 項目で指定した Content-Type ヘッダーの charset パラメーターで指定する値に従う
3	最大サイズ	GET リクエストの場合：8 メガバイト GET リクエスト以外の場合：10 メガバイト

送信情報ファイルは，HTTP 接続ジョブを実行するエージェントホスト上の任意の場所に格納してください。

補足事項

- GET リクエストの場合，送信情報ファイルの内容は，URL パラメーターとして認識します。
- GET リクエスト以外の場合，送信情報ファイルの内容はメッセージボディとして認識します。なお，このメッセージボディには，URL パラメーターを指定できます。ただし，URL パラメーター以外のリクエスト情報は同時に指定できません。
- URL パラメーターを指定する場合は，改行文字を使用しないで，送信情報ファイルに次の形式で記載します。なお，行の先頭に「?」は不要です。

  パラメーター名=パラメーター値[&パラメーター名=パラメーター値&パラメーター名=…]
- リクエスト情報に URL エンコードが必要な値がある場合，送信情報ファイルには URL エンコードされた値で記述する必要があります。

付録 C.3　送信情報ファイル（URL パラメーター）

送信情報ファイル（URL パラメーター）には，Web API へのリクエスト情報の URL パラメーターを設定します。ファイルの記述形式や記述項目は呼び出す Web API の仕様によって異なります。また，リクエスト時に送信する情報がない場合は，送信情報ファイル（URL パラメーター）の作成は不要です。

このファイルは，JP1/AJS3 - Manager と JP1/AJS3 - Agent のバージョンが 12-50 以降で，拡張モードを使用する場合に作成します。拡張モードについては，「2.4.12 Web 上の業務システムと連携する（HTTP 接続ジョブを使ったジョブネットの定義例）」を参照してください。

ファイルの内容にセキュリティ情報が含まれる場合は，ユーザー責任で管理するようにしてください。

送信情報ファイル（URL パラメーター）の仕様を次の表に示します。

表 C-3　送信情報ファイルの仕様（URL パラメーター）

項番	項目	説明
1	権限	ジョブ実行ユーザーにファイル読み取り権限が必要
2	文字コード	接続設定ファイルの Header 項目で指定した Content-Type ヘッダーの charset パラメーターで指定する値に従う
3	最大サイズ	8 メガバイト

送信情報ファイル（URL パラメーター）は，HTTP 接続ジョブを実行するエージェントホスト上の任意の場所に格納してください。

補足事項

- URL パラメーターは，改行文字を使用しないで次の形式で記載します。なお，行の先頭に「?」は不要です。

  パラメーター名=パラメーター値[&パラメーター名=パラメーター値&パラメーター名=...]

- リクエスト情報に URL エンコードが必要な値がある場合，送信情報ファイル（URL パラメーター）には URL エンコードされた値で記述する必要があります。

付録 C.4　送信情報ファイル（メッセージボディ）

送信情報ファイル（メッセージボディ）には，Web API へのリクエスト情報のメッセージボディを設定します。ファイルの記述形式や記述項目は呼び出す Web API の仕様によって異なります。また，リクエスト時に送信する情報がない場合は，送信情報ファイル（メッセージボディ）の作成は不要です。

このファイルは，JP1/AJS3 - Manager と JP1/AJS3 - Agent のバージョンが 12-50 以降で，拡張モードを使用する場合に作成します。拡張モードについては，「2.4.12 Web 上の業務システムと連携する（HTTP 接続ジョブを使ったジョブネットの定義例）」を参照してください。

ファイルの内容にセキュリティ情報が含まれる場合は，ユーザー責任で管理するようにしてください。

送信情報ファイル（メッセージボディ）の仕様を次の表に示します。

表 C–4　送信情報ファイルの仕様（メッセージボディ）

項番	項目	説明
1	権限	ジョブ実行ユーザーにファイル読み取り権限が必要
2	文字コード	接続設定ファイルの Header 項目で指定した Content-Type ヘッダーの charset パラメーターで指定する値に従う
3	最大サイズ	10 メガバイト

送信情報ファイル（メッセージボディ）は，HTTP 接続ジョブを実行するエージェントホスト上の任意の場所に格納してください。

補足事項

リクエスト情報に URL エンコードが必要な値がある場合，送信情報ファイル（メッセージボディ）には URL エンコードされた値で記述する必要があります。

付録 C.5　ステータス格納ファイル

ステータス格納ファイルには，Web API から取得した受信データの HTTP ステータスコードが記述されます。ステータス格納ファイルは，HTTP ステータスコードが 1 以上の場合に，HTTP 接続ジョブが自動で作成するファイルです。

ステータス格納ファイルの仕様を次の表に示します。

表 C-5　ステータス格納ファイルの仕様

項番	項目	説明
1	権限	ファイルを作成するフォルダに，ジョブ実行ユーザーのファイル作成権限が必要 作成されるファイルの権限は次のとおりです。 Windows の場合 所有者：ジョブ実行ユーザー Linux の場合 パーミッション：644 オーナー：ジョブ実行ユーザー グループ：ジョブ実行ユーザーの所属グループ
2	文字コード	ASCII 文字
3	最大サイズ	100 バイト

(1)　記述形式

ステータス格納ファイルの記述形式を次に示します。

```
HTTP-Status-Code=HTTPステータスの値
```

付録 C.6　受信ヘッダー格納ファイル

受信ヘッダー格納ファイルには，Web API から取得した受信データの HTTP ヘッダー部分が記述されます。受信ヘッダー格納ファイルは，HTTP 接続ジョブが自動で作成するファイルです。ファイルへの記述形式は呼び出す Web API の仕様によって異なります。

ファイルの内容にセキュリティ情報が含まれる場合は，ユーザー責任で管理するようにしてください。

受信ヘッダー格納ファイルの仕様を次の表に示します。

表 C-6　受信ヘッダー格納ファイルの仕様

項番	項目	説明
1	権限	ファイルを作成するフォルダに，ジョブ実行ユーザーのファイル作成権限が必要 作成されるファイルの権限は次のとおりです。 Windows の場合 所有者：ジョブ実行ユーザー Linux の場合 パーミッション：644 オーナー：ジョブ実行ユーザー グループ：ジョブ実行ユーザーの所属グループ

項番	項目	説明
2	文字コード	呼び出す Web API の仕様に依存
3	最大サイズ	上限なし

HTTP 接続ジョブの定義項目の受信データ格納方式で，ヘッダーとボディを同じファイルに出力すると指定した場合，受信データのボディ部分も受信ヘッダー格納ファイルに記述されます。

付録 C.7　受信ボディ格納ファイル

受信ボディ格納ファイルには，Web API から取得した受信データの HTTP ボディ部分が記述されます。受信ボディ格納ファイルは，HTTP 接続ジョブが自動で作成するファイルです。ファイルへの記述形式は，呼び出す API の仕様によって異なります。

ファイルの内容にセキュリティ情報が含まれる場合は，ユーザー責任で管理するようにしてください。

受信ボディ格納ファイルの仕様を次の表に示します。

表 C-7　受信ボディ格納ファイルの仕様

項番	項目	説明
1	権限	ファイルを作成するフォルダに，ジョブ実行ユーザーのファイル作成権限が必要 作成されるファイルの権限は次のとおりです。 Windows の場合 所有者：ジョブ実行ユーザー Linux の場合 パーミッション：644 オーナー：ジョブ実行ユーザー グループ：ジョブ実行ユーザーの所属グループ
2	文字コード	呼び出す Web API の仕様に依存
3	最大サイズ	接続設定ファイルの「MaximumReceivedDataSize=**最大受信データサイズ**」で指定

受信ボディ格納ファイルは，HTTP 接続ジョブの定義項目の受信データ格納方式で，ヘッダーとボディを別のファイルに出力すると指定した場合に作成されます。ヘッダーとボディを同じファイルに出力すると指定した場合，受信データのボディ部分は受信ヘッダー格納ファイルに追記され，受信ボディ格納ファイルは作成されません。

補足事項

- HTTP 接続ジョブでは，HTTP ステータスコードによって受信データの HTTP ボディ部分の有無が判断されます。

  HTTP ステータスコードが，100～199，204，または 304 の場合は，受信データに HTTP ボディ部分はないと判断され，受信ボディ格納ファイルは作成されません。

HTTP ステータスコードが上記以外の場合は，RFC2616 に従って HTTP ボディ部分の有無が判断され，HTTP ボディ部分がないと判断された場合は，受信ボディ格納ファイルは作成されません。

- 環境設定パラメーターTRANSFER_DECODING に「Y」を指定している場合，チャンク形式の受信データはデコードして出力されます。環境設定パラメーターTRANSFER_DECODING の詳細については，マニュアル「JP1/Automatic Job Management System 3 構築ガイド　20.14.2(1) TRANSFER_DECODING」を参照してください。

付録 D　各バージョンの変更内容

各バージョンでの変更点については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　付録 B 各バージョンの変更内容」を参照してください。

付録 E　このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　付録 C このマニュアルの参考情報」を参照してください。

付録 F　用語解説

用語解説については，マニュアル「JP1/Automatic Job Management System 3 導入ガイド　付録 D 用語解説」を参照してください。

索引

数字

1 階層に何千個ものルートジョブネットを定義する場合　290
48 時間制に変更した場合のスケジュール見直し　223

H

HTTP 接続ジョブ　193
HTTP 接続ジョブで使用するファイル　399

J

JP1_AJS_Admin　260
JP1_AJS_Editor　260
JP1_AJS_Guest　260
JP1_AJS_Manager　260
JP1_AJS_Operator　260
JP1_JPQ_Admin　263, 265
JP1_JPQ_Operator　263, 265
JP1_JPQ_User　263, 265
JP1/AJS3 が発行するイベントやメッセージを監視する　340
JP1/AJS3 によって終了コードが設定される条件とその終了コード　370
JP1/AJS3 の設計作業の流れ　21
JP1/Base のイベントログトラップ機能　158
JP1/Base のログファイルトラップ機能　156
JP1/Cm2 状態通知ジョブ実行時の戻り値一覧　392
JP1/Cm2 状態通知ジョブの注意事項　349
JP1 イベント受信監視ジョブ　153
JP1 イベント受信監視ジョブ定義時の注意事項　308
JP1 イベント受信監視ジョブの注意事項　304
JP1 イベント送信ジョブ　168
JP1 イベント送信ジョブ実行時の戻り値一覧　388
JP1 イベント送信ジョブの注意事項　347
JP1 権限レベル　259
JP1 権限レベルの検討　259
JP1 資源グループ　259
JP1 資源グループの検討　259
JP1 ユーザーの登録　257

M

MSMQ 送信ジョブ実行時の戻り値一覧　391

O

OR ジョブ　143

Q

QUEUE ジョブ，サブミットジョブ実行環境で使用するコマンドの実行・操作時のアクセス権限　263

S

「SEQ2」を選択した場合に取得メッセージが消失する例としない例　332

W

Windows イベントログ監視ジョブ　158
Windows イベントログ監視ジョブの注意事項　334

あ

アクションジョブを実行したときの戻り値　387
アクセス権限の検討　259
アクセス権を設定する範囲についての検討　255

い

一斉実行　191
一斉配信エージェント　191
イベント情報の引き継ぎ　153
イベントジョブ　151
イベントジョブ実行時の戻り値一覧　387
イベントジョブの打ち切り時間　153
イベントジョブの監視対象　152, 302
イベントジョブの受信情報の引き継ぎ　163
イベントジョブの引き継ぎ情報一覧　394
イベントの検索　154
イベントの到達確認　170

え

エージェント管理情報操作時のアクセス権限　265

か

拡張子による監視対象ファイルの検知の有無　328
各バージョンの変更内容　411
カレンダーの適用例　240
監視開始オプション　316
監視開始オプションを設定した場合のオプションの基本動作例　316
監視状態の引き継ぎ条件　322
監視する事象　310
監視ファイル名の指定例　315

き

起算スケジュール　233
起動条件中の監視対象ファイル名を指定したファイル監視ジョブ　320
起動条件中の監視対象ファイル名を「*（ワイルドカード）」で指定したファイル監視ジョブ　321
起動条件の検討　219
業務設計の概要　20
業務の自動化検討時のヒント　131
業務の自動化のための検討項目　24
業務の自動化を実現する JP1/AJS3 の機能　139

く

グローバルマクロ変数　177

さ

サイクルジョブ定義　232

し

時間経過の監視　160
実行間隔制御ジョブ　160
実行間隔制御ジョブの注意事項　336
実行前のイベント検索　154
実行ユーザー　275
実行ユーザー固定機能　275
自動リトライ　198
従属ユニットの再実行方法　146
受信ヘッダー格納ファイル　408
受信ボディ格納ファイル　409
上位ユニット属性継承機能　267
状態引き継ぎオプションを設定した場合の動作例 323
ジョブネットコネクタ　173
ジョブネット中の監視対象ファイル名を指定したファイル監視ジョブ　319
ジョブネット中の監視対象ファイル名を「*（ワイルドカード）」で指定したファイル監視ジョブ　319
ジョブネット定義・実行時のアクセス権限　260
ジョブネットへのアクセス権の検討　254
ジョブネットへのアクセス権を検討する流れ　254
ジョブの異常終了時に自動でリトライする　198
ジョブの終了コードを確認する　370
所有者権限　266

す

スケジュールに関する検討項目　218
ステータス格納ファイル　407

せ

正規表現　308
セカンダリー認証サーバ　255
設計の流れ　21
接続設定ファイル　399

そ

総称名　315
送信情報ファイル　405
送信情報ファイル（URL パラメーター）　406
送信情報ファイル（メッセージボディ）　407

た

対応づけるユーザーの検討　284

と

登録するユーザーの検討　257

は

判定ジョブ　144
判定ジョブの使用例　144

ひ

引き継ぎ情報設定ジョブ　177
引き継ぎ情報定義時の注意事項　336
標準出力ファイル・標準エラー出力ファイルの注意事項　361
標準ジョブ　142

ふ

ファイル監視ジョブ　154
ファイル監視ジョブ定義時の注意事項　323
ファイル監視ジョブで監視する事象　310
ファイル監視ジョブのオプション　315
ファイル監視ジョブの監視条件　154
ファイル監視ジョブの状態引き継ぎオプション　322
ファイル監視ジョブの注意事項　310
ファイルの有無による判定ジョブの使用例　145
ファイル名の指定方法　315
負荷を分散した処理　131
プライマリー認証サーバ　255
フレキシブルジョブ　188, 191

ほ

保存世代数　292
保存世代数と性能との関係　292
本番運用前の JP1/AJS3 定義内容のチェック　377

ま

マクロ変数　163
待ち合わせ条件　67, 174
待ち合わせ条件付きユニット　68
待ち合わせ状態　72
待ち合わせ対象ユニット　68

め

メール受信監視ジョブの注意事項　330
メール送信ジョブ実行時の戻り値一覧（UNIX）　389
メール送信ジョブ実行時の戻り値一覧（Windows）　388
メール送信ジョブの注意事項　348
メッセージキュー送信ジョブ実行時の戻り値一覧（UNIX）　391
メッセージキュー送信ジョブ実行時の戻り値一覧（Windows）　390

も

戻り値一覧　387

ゆ

ユーザーマッピング　284
ユーザーマッピングを使用したジョブ実行の概要 284
ユニット属性プロファイル　270, 278

り

リカバリージョブ　170
リカバリージョブネット　170
リカバリーユニット定義時の注意事項　300
リターンコードによる判定ジョブの使用例　145
リトライ開始日時　203
リトライ実行　198
リトライ実行回数　202
リトライ状態　201
リトライ情報　201
リトライ設定　199
リトライ中　201
リトライ登録日時　202
リモート電源制御ジョブ実行時の戻り値一覧　392
リモート電源制御ジョブの注意事項　349

る

ルートジョブネットの実行登録数に関する注意事項　290

ろ

ローカル電源制御ジョブ実行時の戻り値一覧　392
ローカル電源制御ジョブの注意事項　349

ログファイル監視ジョブ　156
ログファイル監視ジョブの注意事項　331
ログファイル監視ジョブの動作の概要　157

株式会社 日立製作所

〒 100-8280　東京都千代田区丸の内一丁目 6 番 6 号