

Job Management Partner 1/Automatic Job Management System 3

Overview

3020-3-S02-04(E)

■ Relevant program products

For details about the applicable OS versions, and the service packs and patches required for JPI/Automatic Job Management System 3, see the *Release Notes*.

For Windows Server 2008:

P-2A12-3K97 Job Management Partner 1/Automatic Job Management System 3 - Manager version 09-00

P-2A12-3397 Job Management Partner 1/Automatic Job Management System 3 - Agent version 09-00

P-2A2C-6L97 Job Management Partner 1/Base version 09-00

For Windows 7, Windows Server 2008 and Windows Vista:

P-2A12-3497 Job Management Partner 1/Automatic Job Management System 3 - View version 09-00

For Windows Server 2003 and Windows Server 2003(x64):

P-2412-3K97 Job Management Partner 1/Automatic Job Management System 3 - Manager version 09-00

P-2412-3397 Job Management Partner 1/Automatic Job Management System 3 - Agent version 09-00

P-242C-6L97 Job Management Partner 1/Base version 09-00

For Windows Server 2003, Windows Server 2003(x64), and Windows XP Professional:

P-2412-3497 Job Management Partner 1/Automatic Job Management System 3 - View version 09-00

For HP-UX(IPF):

P-1J12-2792 Job Management Partner 1/Automatic Job Management System 3 - Manager version 09-00

P-1J12-2992 Job Management Partner 1/Automatic Job Management System 3 - Agent version 09-00

P-1J2C-6L92 Job Management Partner 1/Base version 09-00

For Solaris 9(SPARC), and Solaris 10(SPARC):

P-9312-2792 Job Management Partner 1/Automatic Job Management System 3 - Manager version 09-00

P-9312-2992 Job Management Partner 1/Automatic Job Management System 3 - Agent version 09-00

P-9D2C-6L92 Job Management Partner 1/Base version 09-00

For AIX:

P-1M12-2792 Job Management Partner 1/Automatic Job Management System 3 - Manager version 09-00

P-1M12-2992 Job Management Partner 1/Automatic Job Management System 3 - Agent version 09-00

P-1M2C-6L92 Job Management Partner 1/Base version 09-00

■ Trademarks

400 is a trademark of International Business Machines Corporation in the United States, other countries, or both.

Active Directory is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

AIX is a trademark of International Business Machines Corporation in the United States, other countries, or both.

AS/400 is a trademark of International Business Machines Corporation in the United States, other countries, or both.

CONNECT:Queue is a registered trademark of Sterling Commerce, Inc.

HP NonStop is a trademark of Hewlett-Packard Company.

HP-UX is a product name of Hewlett-Packard Company.

Internet Explorer is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Itanium is a trademark of Intel Corporation in the United States and other countries.

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

Lotus Notes is a registered trademark of Lotus Development Corporation.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Excel is a registered trademark of Microsoft Corp. in the U.S. and other countries.

Microsoft Exchange server is a product name of Microsoft Corp. in the U.S. and other countries.

Microsoft Mail is a product name of Microsoft Corp.

Microsoft, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other

countries.

Microsoft SQL Server is a product name of Microsoft Corp.

MQSeries is a trademark of International Business Machines Corporation in the United States, other countries, or both.

MVS is a trademark of International Business Machines Corporation in the United States, other countries, or both.

ORACLE is either a registered trademark or a trademark of Oracle Corporation and/or its affiliates in the United States and/or other countries.

Outlook is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Pentium is a trademark of Intel Corporation in the United States and other countries.

R/3 is a registered trademark or a trademark of SAP AG in Germany and in other countries.

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

SAP is a registered trademark or a trademark of SAP AG in Germany and in other countries.

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

SQL*Plus is either a registered trademark or a trademark of Oracle Corporation and/or its affiliates in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Windows is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Windows Server is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Windows Vista is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

The following program products contain some parts whose copyrights are reserved by Sun Microsystems, Inc.: P-9312-2792, P-9312-2992, and P-9D2C-6L92.

The following program products contain some parts whose copyrights are reserved by UNIX System Laboratories, Inc.: P-9312-2792, P-9312-2992, and P-9D2C-6L92.

Other product and company names mentioned in this document may be the trademarks of their respective owners. Throughout this document Hitachi has attempted to distinguish trademarks from descriptive terms by writing the name with the capitalization used by the manufacturer, or by writing the name with initial capital letters. Hitachi cannot attest to the accuracy of this information. Use of a trademark in this document should not be regarded as affecting the validity of the trademark.

■ Restrictions

Information in this document is subject to change without notice and does not represent a commitment on the part of Hitachi. The software described in this manual is furnished according to a license agreement with Hitachi. The license agreement contains all of the terms and conditions governing your use of the software and documentation, including all warranty rights, limitations of liability, and disclaimers of warranty.

Material contained in this document may describe Hitachi products not available or features not available in your country.

No part of this material may be reproduced in any form or by any means without permission in writing from the publisher.

Printed in Japan.

■ Edition history

Edition 1 (3020-3-S02-03(E)): November 2009

Edition 1.4 (3020-3-S02-04(E)): November 2010

■ Copyright

Copyright (C) 2009, 2010, Hitachi, Ltd.

Copyright (C) 2009, 2010, Hitachi Solutions, Ltd.

Summary of amendments

The following table lists changes in this manual (3020-3-S02-04(E)) and product changes related to this manual.

Changes	Location
Descriptions have been changed. For details, see Appendix C.	<i>Appendix C</i>

In addition to the above changes, minor editorial corrections have been made.

Preface

This manual describes the functions of Job Management Partner 1/Automatic Job Management System 3 (abbreviated hereafter to *JP1/AJS3*).

Intended readers

This manual is intended for:

- Those who wish to operate an automatic job execution system with JP1/AJS3 and those who design automatic job execution systems.
- Those who operate an automatic job execution system with JP1/AJS3.

Organization of this manual

This manual is organized into the following parts. The manual is a common reference for all supported operating systems. Any platform-dependent differences in functionality are noted in the manual.

PART 1: Overview

This part describes the features of JP1/AJS3 and its component programs.

PART 2: Basic Operations

This part describes the basic functions of JP1/AJS3.

PART 3: Functions (Applications)

This part describes the functions for linking JP1/AJS3 with other products, and the JP1/AJS2 functionality that is supported in JP1/AJS3.

Related publications

This manual is part of a related set of manuals. The manuals in the set are listed below (with the manual numbers):

About JP1/AJS:

- *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide* (3020-3-S03(E))
- *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide* (3020-3-S04(E))
- *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1* (3020-3-S05(E))
- *Job Management Partner 1/Automatic Job Management System 3 Configuration*

Guide 2 (3020-3-S06(E))

- *Job Management Partner 1/Automatic Job Management System 3 Administration Guide (3020-3-S07(E))*
- *Job Management Partner 1/Automatic Job Management System 3 Troubleshooting (3020-3-S08(E))*
- *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide (3020-3-S09(E))*
- *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1 (3020-3-S10(E))*
- *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2 (3020-3-S11(E))*
- *Job Management Partner 1/Automatic Job Management System 3 Linkage Guide (3020-3-S12(E))*
- *Job Management Partner 1/Automatic Job Management System 3 Messages 1 (3020-3-S13(E))*
- *Job Management Partner 1/Automatic Job Management System 3 Messages 2 (3020-3-S14(E))*
- *Job Management Partner 1/Automatic Job Management System 3 - Definition Assistant Description, Operator's Guide and Reference (3020-3-S17(E))*
- *Job Management Partner 1/Automatic Job Management System 3 - Web Operation Assistant Description, Operator's Guide and Reference (3020-3-S18(E))*
- *Job Management Partner 1/Automatic Job Management System 3 for Enterprise Applications Description, User's Guide and Reference (3020-3-S29(E))*
- *Job Management Partner 1/Automatic Job Management System 2 for Oracle E-Business Suite Description, User's Guide and Reference (3020-3-F27(E))*

About JP1:

- *Job Management Partner 1/Base User's Guide (3020-3-R71(E))*
- *Job Management Partner 1/Base Messages (3020-3-R72(E))*
- *Job Management Partner 1/Base Function Reference (3020-3-R73(E))*
- *Job Management Partner 1/Integrated Management - Manager Overview and System Design Guide (3020-3-R76(E))*
- *Job Management Partner 1/Integrated Management - Manager Configuration Guide (3020-3-R77(E))*
- *Job Management Partner 1/Integrated Management - Manager Administration*

Guide (3020-3-R78(E))

- *Job Management Partner 1/Integrated Management - Manager GUI Reference (3020-3-R79(E))*
- *Job Management Partner 1/Integrated Management - Manager Command and Definition File Reference (3020-3-R80(E))*
- *Job Management Partner 1/Integrated Management - Manager Messages (3020-3-R81(E))*
- *Job Management Partner 1/Script Description and Reference (3020-3-K55(E)), for Windows systems*
- *Job Management Partner 1/File Transmission Server/FTP Description, Reference, and Operator's Guide (3020-3-S36(E)), for Windows systems*
- *Job Management Partner 1/File Transmission Server/FTP Description, Reference, and Operator's Guide (3020-3-S37(E)), for UNIX systems*
- *Job Management Partner 1/Software Distribution Description and Planning Guide (3020-3-S79(E)), for Windows systems*
- *Job Management Partner 1/Software Distribution Setup Guide (3020-3-S80(E)), for Windows systems*
- *Job Management Partner 1/Software Distribution System Administrator's Guide Volume 1 (3020-3-S81(E)), for Windows systems*
- *Job Management Partner 1/Software Distribution System Administrator's Guide Volume 2 (3020-3-S82(E)), for Windows systems*
- *Job Management Partner 1/Software Distribution Automatic Installation Tool Description and Reference (3020-3-S83(E)), for Windows systems*
- *Job Management Partner 1/Software Distribution Administrator Kit Description and Operator's Guide (3020-3-S84(E))*
- *Job Management Partner 1/Software Distribution Client Description and User's Guide (3020-3-S85(E)), for UNIX systems*
- *Job Management Partner 1/Software Distribution SubManager Description and Administrator's Guide (3020-3-L42(E)), for UNIX systems*
- *Job Management Partner 1/Software Distribution Manager Description and Administrator's Guide (3000-3-841(E))*
- *Job Management Partner 1/NQSEEXEC System Administrator's Guide (3020-3-F30(E))*
- *Job Management Partner 1/Consolidated Management 2/Extensible SNMP Agent Description, Operator's Guide and Reference (3020-3-L04(E)), for UNIX systems*

- *Job Management Partner 1/Open Job Entry Description, User's Guide and Reference (6190-3-365(E)), for VOS3 systems*
- *Job Management Partner 1/Open Job Entry Description, User's Guide and Reference (9000-3-365(E)), for MVS systems*
- *Job Management Partner 1/Open Job Entry Description, User's Guide and Reference (9000-3-366(E)), for OSIV/MSP systems*
- *Job Management Partner 1/Open Job Entry for Midrange Computer Description and User's Guide (9000-3-367(E))*

Conventions: Abbreviations

This manual uses the following abbreviations for product names:

Abbreviation		Full name or meaning
JP1/AJS3	JP1/AJS3 - Manager	Job Management Partner 1/Automatic Job Management System 3 - Manager
	JP1/AJS3 - Agent	Job Management Partner 1/Automatic Job Management System 3 - Agent
	JP1/AJS3 - View	Job Management Partner 1/Automatic Job Management System 3 - View
JP1/AJS2	JP1/AJS2 - Manager	Job Management Partner 1/Automatic Job Management System 2 - Manager
	JP1/AJS2 - Agent	Job Management Partner 1/Automatic Job Management System 2 - Agent
	JP1/AJS2 - View	Job Management Partner 1/Automatic Job Management System 2 - View
JP1/AJS2 - Advanced Manager		Job Management Partner 1/Automatic Job Management System 2 - Advanced Manager [#]
JP1/AJS2 - Client Toolkit		Job Management Partner 1/Automatic Job Management System 2 - Client Toolkit [#]
JP1/AJS3 - Definition Assistant		Job Management Partner 1/Automatic Job Management System 3 - Definition Assistant
JP1/AJS3 - Web Operation Assistant		Job Management Partner 1/Automatic Job Management System 3 - Web Operation Assistant
JP1/AJS3 for Enterprise Applications		Job Management Partner 1/Automatic Job Management System 3 for Enterprise Applications

Abbreviation		Full name or meaning
JP1/AJS2 for Oracle E-Business Suite		Job Management Partner 1/Automatic Job Management System 2 for Oracle E-Business Suite
NNM	HP NNM	HP Network Node Manager Software version 7.5 or earlier
		HP Network Node Manager Software Starter Edition version 7.5 or earlier
JP1/FTP		Job Management Partner 1/File Transmission Server/FTP
JP1/IM	JP1/IM - Manager	Job Management Partner 1/Integrated Management - Manager
	JP1/IM - View	Job Management Partner 1/Integrated Management - View
	JP1/IM - Central Console	Job Management Partner 1/Integrated Manager - Central Console [#]
	JP1/IM - Central Scope	Job Management Partner 1/Integrated Manager - Central Scope [#]
JP1/OJE		Job Management Partner 1/Open Job Entry
JP1/OJE for Midrange Computer		Job Management Partner 1/Open Job Entry for Midrange Computer
JP1/SES		Job Management Partner 1/System Event Service
JP1/OJE for VOS3		VOS3 Job Management Partner 1/Open Job Entry
MSCS		Microsoft(R) Cluster Server
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		Microsoft(R) Internet Explorer(R)
Microsoft Mail		Microsoft(R) Mail

Abbreviation		Full name or meaning
MSMQ		Microsoft(R) Message Queue Server
Outlook	Outlook 2000	Microsoft(R) Outlook(R) 2000
	Outlook 2002	Microsoft(R) Outlook(R) 2002
	Outlook 2003	Microsoft(R) Outlook(R) 2003
	Outlook 2007	Microsoft(R) Outlook(R) 2007
	Outlook Express	Microsoft(R) Outlook(R) Express
Microsoft SQL Server		Microsoft(R) SQL Server
		Microsoft(R) SQL Server Enterprise Edition
Windows 7		Microsoft(R) Windows(R) 7 Enterprise
		Microsoft(R) Windows(R) 7 Professional
		Microsoft(R) Windows(R) 7 Ultimate
Windows Server 2003	Windows Server 2003	Microsoft(R) Windows Server(R) 2003, Enterprise Edition Operating System
		Microsoft(R) Windows Server(R) 2003, Standard Edition Operating System
	Windows Server 2003 (x64)	Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition
		Microsoft(R) Windows Server(R) 2003, Standard x64 Edition
Windows Server 2008		Microsoft(R) Windows Server(R) 2008 Datacenter
		Microsoft(R) Windows Server(R) 2008 Enterprise
		Microsoft(R) Windows Server(R) 2008 Standard
Windows Vista		Microsoft(R) Windows Vista(R) Business
		Microsoft(R) Windows Vista(R) Enterprise
		Microsoft(R) Windows Vista(R) Ultimate
Windows XP Professional		Microsoft(R) Windows(R) XP Professional Operating System

Abbreviation		Full name or meaning
AIX		AIX 5L 5.3
		AIX V6.1
HP-UX	HP-UX (IPF)	HP-UX 11i V2(IPF)
		HP-UX 11i V3(IPF)
Solaris		Solaris 9(SPARC)
		Solaris 10(SPARC)
SAP BW		SAP Business Information Warehouse
SAP R/3		SAP R/3(R)

#: Version 7

- In this manual, *JP1/AJS* is sometimes used generically, referring to *JP1/AJS3* and *JP1/AJS2*.
- *Windows* is sometimes used generically, referring to Windows 7, Windows Server 2008, Windows Vista, Windows Server 2003, and Windows XP Professional.
- *UNIX* is sometimes used generically, referring to HP-UX, Solaris, and AIX.

This manual also uses the following abbreviations:

Abbreviation	Full name or meaning
ACL	Access Control List
DB	Database
DBMS	Database Management System
DNS	Domain Name System
EUC	Extended UNIX Code
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GUI	Graphical User Interface
IPF	Itanium(R) Processor Family
ISAM	Indexed Sequential Access Method
LAN	Local Area Network

Abbreviation	Full name or meaning
MAPI	Messaging Application Programming Interface
MIB	Management Information Base
MIME	Multipurpose Internet Mail Extensions
NAT	Network Address Translator
NFS	Network File System
NIC	Network Interface Card
OS	Operating System
RDB	Relational Database
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SUP	Service Using Program
TCP/IP	Transmission Control Protocol/Internet Protocol
UAC	User Account Control
UNC	Universal Naming Convention
WAN	Wide Area Network
WOW64	Windows On Windows 64
WSDL	Web Services Description Language

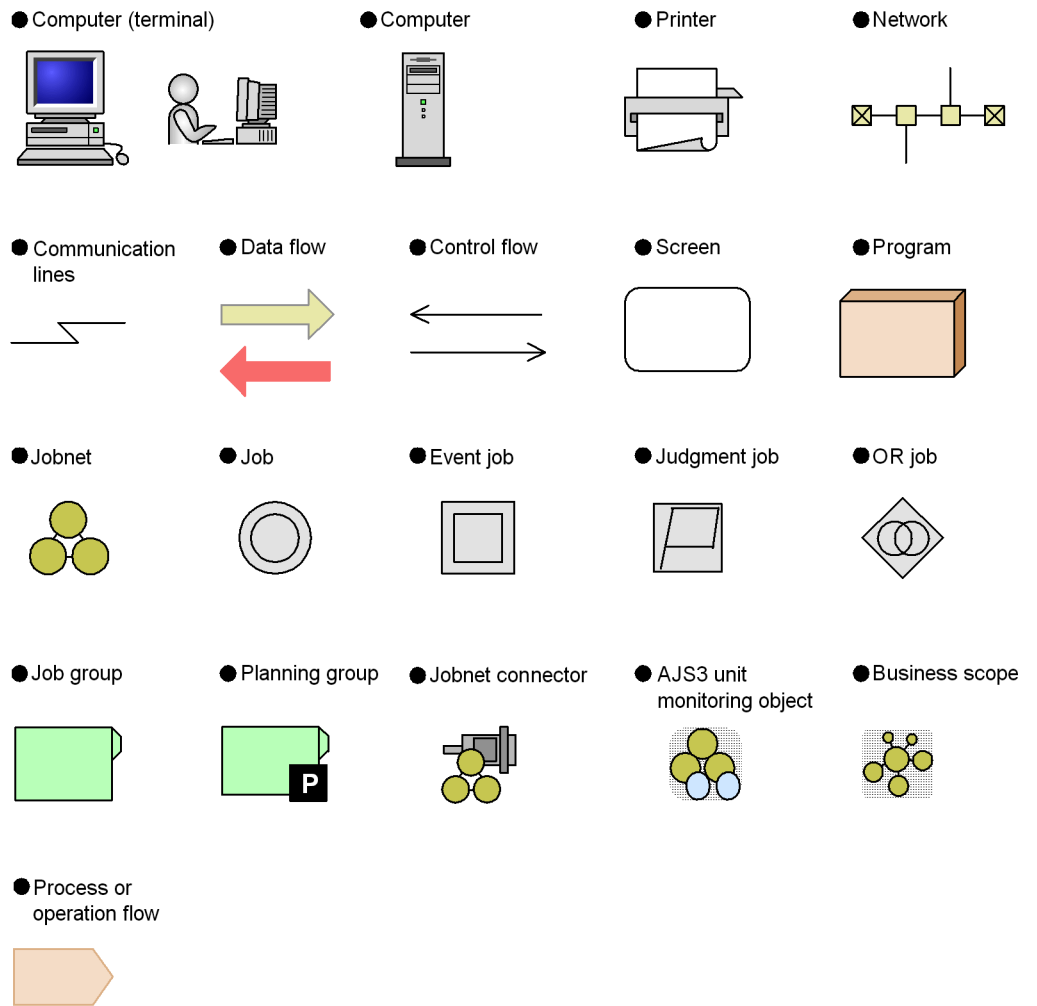
JP1 program reorganization in version 8

The following changes have been made to the JP1 product suite in version 8:

- JP1/AJS2 - Advanced Manager has been eliminated, and the database provided by JP1/AJS2 - Advanced Manager has been integrated into JP1/AJS2 - Manager in JP1 Version 8.
- JP1/AJS2 - Client Toolkit has been eliminated.
- JP1/AJS2 - View is provided only in the Windows version.

Conventions: Diagrams

This manual uses the following conventions in diagrams:



Conventions: Fonts and symbols

Font and symbol conventions are classified as:

- General font conventions
- Conventions in syntax explanations

These conventions are described below.

General font conventions

The following table lists the general font conventions:

Font	Convention
Bold	<p>Bold type indicates text on a window, other than the window title. Such text includes menus, menu options, buttons, radio box options, or explanatory labels. For example, bold is used in sentences such as the following:</p> <ul style="list-style-type: none"> • From the File menu, choose Open. • Click the Cancel button. • In the Enter name entry box, type your name.
<i>Italics</i>	<p>Italics are used to indicate a placeholder for some actual text provided by the user or system. Italics are also used for emphasis. For example:</p> <ul style="list-style-type: none"> • Write the command as follows: <i>copy source-file target-file</i> • Do <i>not</i> delete the configuration file.
Code font	<p>A code font indicates text that the user enters without change, or text (such as messages) output by the system. For example:</p> <ul style="list-style-type: none"> • At the prompt, enter <code>dir</code>. • Use the <code>send</code> command to send mail. • The following message is displayed: <code>The password is incorrect.</code>

Examples of coding and messages appear as follows (although there may be some exceptions, such as when coding is included in a diagram):

```
MakeDatabase
...
StoreDatabase temp DB32
```

In examples of coding, an ellipsis (...) indicates that one or more lines of coding are not shown for purposes of brevity.

Conventions in syntax explanations

Syntax definitions appear as follows:

```
StoreDatabase [temp|perm] (database-name ...)
```

The following table lists the conventions used in syntax explanations:

Example font or symbol	Convention
<code>StoreDatabase</code>	Code-font characters must be entered exactly as shown.
<i>database-name</i>	This font style marks a placeholder that indicates where appropriate characters are to be entered in an actual command.
SD	Bold code-font characters indicate the abbreviation for a command.
<u>Perm</u>	Underlined characters indicate the default value.
[]	Square brackets enclose an item or set of items whose specification is optional.

Example font or symbol	Convention
	Only one of the options separated by a vertical bar can be specified at the same time.
...	An ellipsis (...) indicates that the item or items enclosed in () or [] immediately preceding the ellipsis may be specified as many times as necessary.
()	Parentheses indicate the range of items to which the vertical bar () or ellipsis (...) is applicable.

Conventions for mathematical expressions

This manual uses the following symbols in mathematical expressions:

Symbol	Meaning
x	Multiplication sign
/	Division sign
↑ ↑	The calculation result is rounded up to the next whole number. Example: The result of $\uparrow 34 / 3 \uparrow$ is 12.
~ (tilde)	The item shown before this symbol must be specified in accordance with the conventions shown for angle brackets, double parentheses, and double angle brackets (below).
< > (angle brackets)	Indicates the characters and lexical elements that can be specified. <characters> One or more Kanji characters, katakana characters, upper-case alphabetic characters, lower-case alphabetic characters, or numeric characters <numeric> 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9 <alphabetic character> A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, \, #, or @ <alphanumeric character> Alphabetic or numeric character <symbolic name> No more than eight alphanumeric characters beginning with an alphabetic character <unsigned integer> One or more numeric characters <hexadecimal character> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, or F <file name> A system-determined name assigned to a file <path> The directories contained in the path, with each name separated by a forward slash (/) or backslash (\). The path notation is OS-dependent.

Symbol	Meaning
(()) (double parentheses)	Indicates the range of specifiable values.
<< >> (double angle brackets)	Indicates the default assumed by the system when a value is unspecified. Example: If you do not specify <i>days-to-keep-form</i> ~<numeric> ((0 to 365)) <<365>>, 365 is assumed as the number of days to keep the form.
MAX	Choose the largest of the calculation results. Example: The result of MAX (3 x 6, 4 + 7) is 18.

Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.
- 1 MB (megabyte) is 1,024² bytes.
- 1 GB (gigabyte) is 1,024³ bytes.
- 1 TB (terabyte) is 1,024⁴ bytes.

Conventions: Meaning of "directory" and "folder"

As a general rule, Windows folder names are used in this manual if they are identical to UNIX directory names.

Conventions: Meaning of "member of the Administrators group"

The term *member of the Administrators group* in this manual refers to a user who is a member of the Administrators group on the local PC only. The privileges of local users, domain users, and Active Directory users are no different as long as these users are members of the Administrators group on the local PC.

Conventions: Version numbers

The version numbers of Hitachi program products are usually written as two sets of two digits each, separated by a hyphen. For example:

- Version 1.00 (or 1.0) is written as 01-00.
- Version 2.05 is written as 02-05.
- Version 2.50 (or 2.5) is written as 02-50.
- Version 12.25 is written as 12-25.

The version number might be shown on the spine of a manual as *Ver. 2.00*, but the same version number would be written in the program as *02-00*.

Default installation folders of JP1/AJS3 for Windows

The default installation folders of JP1/AJS3 for Windows are as follows:

Default installation folders of JP1/AJS3 - Manager:

system-drive\Program Files^{#1}\HITACHI\JP1AJS2

and

system-drive\Program Files^{#1}\HITACHI\JP1AJS2CM

Default installation folder of JP1/AJS3 - Agent:

system-drive\Program Files^{#1}\HITACHI\JP1AJS2

Default installation folder of JP1/AJS3 - View:

system-drive\Program Files^{#2}\HITACHI\JP1AJS2V

#1

For 64-bit versions of Windows Server 2008 and Windows Server 2003 (x64), replace Program Files with Program Files (x86).

#2

For 64-bit versions of Windows 7, Windows Server 2008, Windows Vista, and Windows Server 2003 (x64), replace Program Files with Program Files (x86).

Online manual

JP1/AJS3 - View comes with an online manual that you can read in either of the following browsers:

- Microsoft Internet Explorer version 6.0 or later
- Windows Internet Explorer Version 7.0 or later

The online manual has the same contents as the following manuals:

- *Job Management Partner 1/Automatic Job Management System 3 Overview*
- *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*
- *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*
- *Job Management Partner 1/Automatic Job Management System 3 Configuration*

Guide 1

- *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 2*
- *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*
- *Job Management Partner 1/Automatic Job Management System 3 Troubleshooting*
- *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*
- *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*
- *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2*
- *Job Management Partner 1/Automatic Job Management System 3 Linkage Guide*
- *Job Management Partner 1/Automatic Job Management System 3 Messages 1*
- *Job Management Partner 1/Automatic Job Management System 3 Messages 2*

In JP1/AJS3 - View, you can view the manual by choosing **Help** and then **Contents**. You can also press the **F1** key to view the manual contents. Your Web browser must be associated with a file that has the extension `htm`; otherwise, the online manual will not be displayed correctly. If this happens, associate the `htm` file with the Web browser.

Cautionary note

Depending on the OS settings, the online manual might appear in the active window of the browser when you launch the manual from the **Start** menu.

Organization of JP1/AJS3 manuals and choosing the right manuals

There are fourteen JP1/AJS3 manuals. The following table summarizes their contents.

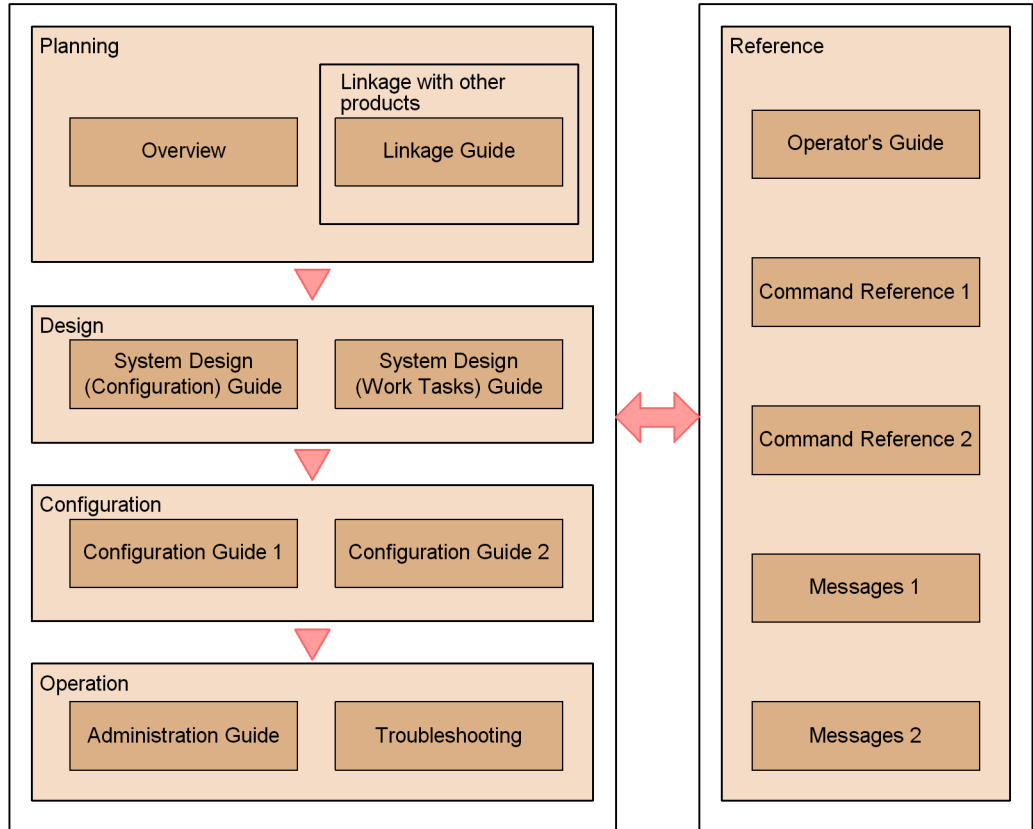
Note that *Job Management Partner 1/Automatic Job Management System 3* is not listed in the table.

No.	Manual	Contents
1	<i>Overview</i> (3020-3-S02(E))	<ul style="list-style-type: none">• JP1/AJS3 features• Description of functions
2	<i>System Design (Configuration) Guide</i> (3020-3-S03(E))	<ul style="list-style-type: none">• Information that must be considered when designing a system• Cautionary notes on designing a system

No.	Manual	Contents
3	<i>System Design (Work Tasks) Guide</i> (3020-3-S04(E))	<ul style="list-style-type: none"> • Information that must be considered when constructing jobs and jobnets • Cautionary notes on designing jobs and jobnets
4	<i>Configuration Guide 1</i> (3020-3-S05(E))	<ul style="list-style-type: none"> • Installation and setup procedures • Environment setup procedure by operation type
5	<i>Configuration Guide 2</i> (3020-3-S06(E))	<ul style="list-style-type: none"> • Description of environment setting parameters
6	<i>Administration Guide</i> (3020-3-S07(E))	<ul style="list-style-type: none"> • Information required to operate a system • Know-how useful for JP1/AJS3 operation
7	<i>Troubleshooting</i> (3020-3-S08(E))	<ul style="list-style-type: none"> • How to troubleshoot errors • Data required when an error occurs
8	<i>Operator's Guide</i> (3020-3-S09(E))	<ul style="list-style-type: none"> • How to operate JP1/AJS3 - View • How to operate JP1/AJS3 Console View • Description of windows and dialog boxes
9	<i>Command Reference 1</i> (3020-3-S10(E))	<ul style="list-style-type: none"> • Command syntax
10	<i>Command Reference 2</i> (3020-3-S11(E))	<ul style="list-style-type: none"> • Syntax of commands used for setup and special operations • Syntax and coding examples of information definition files
11	<i>Linkage Guide</i> (3020-3-S12(E))	<ul style="list-style-type: none"> • Description of functions that can be used when linked with other products and the setup method
12	<i>Messages 1</i> (3020-3-S13(E))	<ul style="list-style-type: none"> • Messages output by JP1/AJS3 (messages beginning with KAVC to KAVT)
13	<i>Messages 2</i> (3020-3-S14(E))	<ul style="list-style-type: none"> • Messages output by JP1/AJS3 (messages beginning with KAVU to KNAD)

Use the following illustration and table as a guide to determine the manuals you need to read.

Organization of JP1/AJS3 manuals



Purpose	Required reading	Read as necessary
To learn about JP1/AJS3's functionalities	<ul style="list-style-type: none"> • <i>Overview</i> (3020-3-S02(E)) 	<ul style="list-style-type: none"> • <i>Linkage Guide</i> (3020-3-S12(E))
To configure a system (including installation and setup) that automatically runs jobs	<ul style="list-style-type: none"> • <i>System Design (Configuration) Guide</i> (3020-3-S03(E)) • <i>Configuration Guide 1</i> (3020-3-S05(E)) 	<ul style="list-style-type: none"> • <i>Configuration Guide 2</i> (3020-3-S06(E)) • <i>Linkage Guide</i> (3020-3-S12(E))
To design work tasks that will be automated (including job definitions and schedule definitions)	<ul style="list-style-type: none"> • <i>System Design (Work Tasks) Guide</i> (3020-3-S04(E)) 	<ul style="list-style-type: none"> • <i>Operator's Guide</i> (3020-3-S09(E))

Purpose	Required reading	Read as necessary
To learn about monitoring and maintaining a running system.	<ul style="list-style-type: none"> • <i>Administration Guide</i> (3020-3-S07(E)) 	<ul style="list-style-type: none"> • <i>Troubleshooting</i> (3020-3-S08(E)) • <i>Messages 1</i> (3020-3-S13(E)) • <i>Messages 2</i> (3020-3-S14(E))
To learn about what action to take for problems that occur during operation.	<ul style="list-style-type: none"> • <i>Troubleshooting</i> (3020-3-S08(E)) 	<ul style="list-style-type: none"> • <i>Messages 1</i> (3020-3-S13(E)) • <i>Messages 2</i> (3020-3-S14(E))
To learn about operating JP1/AJS3	<ul style="list-style-type: none"> • <i>Operator's Guide</i> (3020-3-S09(E)) 	<ul style="list-style-type: none"> • <i>Command Reference 1</i> (3020-3-S10(E)) • <i>Command Reference 2</i> (3020-3-S11(E))

Regular expressions available in JP1/AJS3

Regular expressions can be used in some items in dialog boxes and commands. For details about regular expressions in Windows, see the *Job Management Partner 1/Base User's Guide*. For details about regular expressions in UNIX, see your UNIX documentation.

The regular expressions that you can use when executing an event job on a Windows host depend on the JP1/Base settings. For details on setting regular expressions for event job execution, see the explanation about extending the available regular expressions in the *Job Management Partner 1/Base User's Guide*.

Searching may take a long time if you often use the regular expression `.*` (which means match any character or characters). In long messages, use `.*` only where necessary. In UNIX, you can use `[^]*` (repeat characters other than space characters) instead of `.*` when you want to find a match other than space characters. Using `[^]*` reduces the search time.

About NNM linkage

JP1/AJS3 supports linkage with the following products:

- HP Network Node Manager Software version 6 or earlier
- HP Network Node Manager Starter Edition Software version 7.5 or earlier

In this manual, these products are indicated as *HP NNM*.

Note that linkage with the following products is not supported:

- HP Network Node Manager i Software v8.10

Contents

Preface	i
Intended readers	i
Organization of this manual	i
Related publications	i
Conventions: Abbreviations	iv
JP1 program reorganization in version 8	viii
Conventions: Diagrams	viii
Conventions: Fonts and symbols	ix
Conventions: KB, MB, GB, and TB	xii
Conventions: Meaning of "directory" and "folder"	xii
Conventions: Meaning of "member of the Administrators group"	xii
Conventions: Version numbers	xii
Default installation folders of JP1/AJS3 for Windows	xiii
Online manual	xiii
Organization of JP1/AJS3 manuals and choosing the right manuals	xiv
Regular expressions available in JP1/AJS3	xvii
About NNM linkage	xvii

PART 1: Overview

1. Overview of JP1/AJS3	1
1.1 Features of JP1/AJS3	2
1.1.1 Automating an application	2
1.1.2 Centralized management in a manager/agent configuration	5
1.1.3 GUI-based definition and monitoring	5
1.1.4 Support for a wide range of systems	7
1.1.5 Types of applications that can be automated	9
1.1.6 Benefits of deploying JP1/AJS3	9
1.2 Functions provided in JP1/AJS3	10
1.2.1 Overview of JP1/AJS3 functions	10
1.2.2 Function inter-relationships	11
1.2.3 System management control	13
1.2.4 Scheduler control	13
1.2.5 Agent management control	13
1.2.6 Job execution control	14
1.2.7 Event/action control	14
1.2.8 Embedded database control	14

1.2.9 View control	15
1.2.10 Other functions	15
1.3 JP1/AJS3 program configuration.....	16
1.4 JP1/AJS3 database configurations	18

PART 2: Basic Operations

2. Automating Applications Using JP1/AJS3 23

2.1 Preparing to automate applications.....	24
2.1.1 Application assessment.....	24
2.1.2 Procedures for deploying JP1/AJS3	28

3. Defining Automated Applications 31

3.1 Hierarchical structure of the job network	32
3.1.1 Units.....	34
3.1.2 Building a jobnet	50
3.1.3 Creating a job flow	53
3.2 Defining a calendar for JP1/AJS3 operation.....	59
3.2.1 Defining open days and closed days.....	59
3.2.2 Setting the base day and base time	59
3.3 Defining a schedule	63
3.3.1 Time formats and schedule rules for root jobnets	63
3.3.2 Defining a schedule	65
3.3.3 Concurrent execution and schedule option.....	83
3.4 Defining a start condition	86
3.4.1 Purpose of a start condition	86
3.4.2 Monitoring generation and execution generation.....	89
3.4.3 Valid range of a start condition.....	90
3.4.4 Concurrent execution of execution generations	95
3.4.5 Concurrent execution of monitoring generations	97
3.4.6 Retention of execution generations with satisfied start conditions	100
3.4.7 Suppressing further executions of a jobnet with start conditions after it abnormally terminates	104
3.4.8 Operations on a jobnet registered for execution with a start condition	119
3.4.9 Start condition and number of logs to keep	121

4. Executing an Application 123

4.1 Registering a jobnet for execution.....	124
4.1.1 Methods of registering a jobnet for execution.....	124
4.1.2 Specifying macro variable values during registration for execution.....	134
4.2 Managing jobnet generations.....	142
4.2.1 Managing the scheduled generations of a jobnet.....	142
4.2.2 Jobnet generation number.....	142

4.2.3	Jobnet generation management	145
4.3	Checking execution registration information	155
4.4	Checking execution schedules	158
4.4.1	Checking a jobnet schedule	158
4.4.2	Simulation	160
4.4.3	Statistical information	166
4.5	Operations on registered jobnets	169
4.5.1	Canceling registration of a jobnet	170
4.5.2	Adding an execution schedule to a jobnet	172
4.5.3	Temporarily changing the execution start time of a jobnet	173
4.5.4	Executing a registered jobnet immediately	177
4.5.5	Prohibiting execution of a job or jobnet	182
4.5.6	Temporarily changing the hold attribute of a job or jobnet	184
4.5.7	Temporarily changing the delay monitor settings of a jobnet	184
4.5.8	Temporarily changing the priority of a jobnet	185
4.5.9	Interrupting a jobnet	187
4.5.10	Killing a job or jobnet	187
4.5.11	Rerunning a job or jobnet	188
4.5.12	Changing the status of a job	196
4.5.13	Methods of specifying generations when executing commands	197
4.5.14	Automatically switching a jobnet definition at a specified time	202
4.5.15	Changing job and jobnet definitions without unregistering the jobnet	205

5. Job Execution Environments 231

5.1	How jobs are executed	232
5.1.1	Execution agents	233
5.1.2	Executing jobs concurrently	236
5.1.3	Execution agent group	238
5.1.4	Default execution agent	239
5.2	Restricting job transfer	242
5.3	Job execution environments and system operation	244
5.3.1	User account for job execution	244
5.3.2	OS user environment used for job execution	244
5.3.3	Access permission to executable files and script files	245
5.3.4	Checking file access permission at job execution (Unix jobs)	245
5.3.5	HP-UX OS user groups	247
5.3.6	Login shell at job startup (Unix jobs and action jobs)	247
5.3.7	Value of umask set for the standard output file and the standard error output file (UNIX only)	248
5.3.8	Monitoring the status of registered jobs	249
5.3.9	Distributing jobs among execution hosts (agent hosts)	253
5.3.10	Killing jobs	255
5.3.11	Stopping JP1/AJS3 services and job execution control	256
5.3.12	Group ID for job execution (UNIX only)	257

6. Monitoring Applications	259
6.1 Monitoring execution status and checking execution results	260
6.1.1 Status levels of jobnets, jobs, and jobnet connectors.....	260
6.1.2 Using JP1/AJS3 - View to monitor jobs and jobnets.....	270
7. Monitoring Applications Using JP1/AJS3 Console	301
7.1 Overview of monitoring applications using JP1/AJS3 Console.....	302
7.1.1 Monitoring applications using JP1/AJS3 Console	302
7.1.2 How JP1/AJS3 Console differs from JP1/AJS3 - View	304
7.1.3 Components of JP1/AJS3 Console	308
7.2 Defining the applications to monitor and the monitoring method.....	309
7.2.1 Objects used in JP1/AJS3 Console	309
7.2.2 Modes	310
7.2.3 Creating a monitored object	311
7.2.4 Setting the monitoring properties	315
7.3 Monitoring applications	331
7.3.1 Updating statuses.....	331
7.3.2 Status acquired by JP1/AJS3 Console	331
7.3.3 Manipulating monitored jobnets.....	336
7.3.4 Starting JP1/AJS3 - View	337
8. User Management and Access Control	339
8.1 User management using the functions of JP1/Base	340
8.1.1 User management using the JP1/Base user authentication function	340
8.1.2 User management using the user mapping function of JP1/Base.....	344
8.2 Settings for restricting access to units.....	347
8.2.1 Unit owner permission.....	347
8.2.2 Setting the JP1 resource group	347
8.2.3 Setting the Executed by attribute.....	348
8.2.4 Setting the User name (for PC jobs and Unix jobs)	348
8.3 User management using JP1/AJS3 Console	350
PART 3: Functions (Applications)	
9. Linking JP1/AJS3 with Other Products	353
9.1 Functions implemented by JP1/AJS3 product linkage	354
10. Supported JP1/AJS2 Functionality	361
10.1 Using a planning group to change the plans for root jobnets	362
10.2 Remote jobnet	391
10.3 Manager jobnet	396
10.4 QUEUE jobs and submit jobs	397
10.5 Queueless jobs	405

Appendixes	409
A. Information Passed by Event Jobs	410
B. Version Revisions	416
B.1 Revisions in 09-00	416
B.2 Revisions in 08-00	419
B.3 Revisions in 07-50	420
B.4 Revisions in 07-00	423
C. Changes in 3020-3-S02-04(E)	425
D. Glossary	426
Index	441

Chapter

1. Overview of JP1/AJS3

The JP1/AJS3 suite of programs is for automating and expediting application processing. This chapter describes JP1/AJS3 and its component programs.

- 1.1 Features of JP1/AJS3
- 1.2 Functions provided in JP1/AJS3
- 1.3 JP1/AJS3 program configuration
- 1.4 JP1/AJS3 database configurations

1.1 Features of JP1/AJS3

Computer-based operations performed on a daily basis typically include a large number of regular and routine tasks. For example, an enterprise might need to calculate daily sales data, prepare daily reports, produce monthly accounting summaries, or issue order entry forms. Many operations are not completed in a single step but entail a number of processes carried out in a set order, or may require various conditions to be fulfilled.

By automating operations of this nature, costs can be reduced and valuable human resources can be freed up for more creative work. Automation allows the system to be run reliably with just a few operators.

JP1/AJS3 is designed to automate routine applications. It can also be used to automate complex applications.

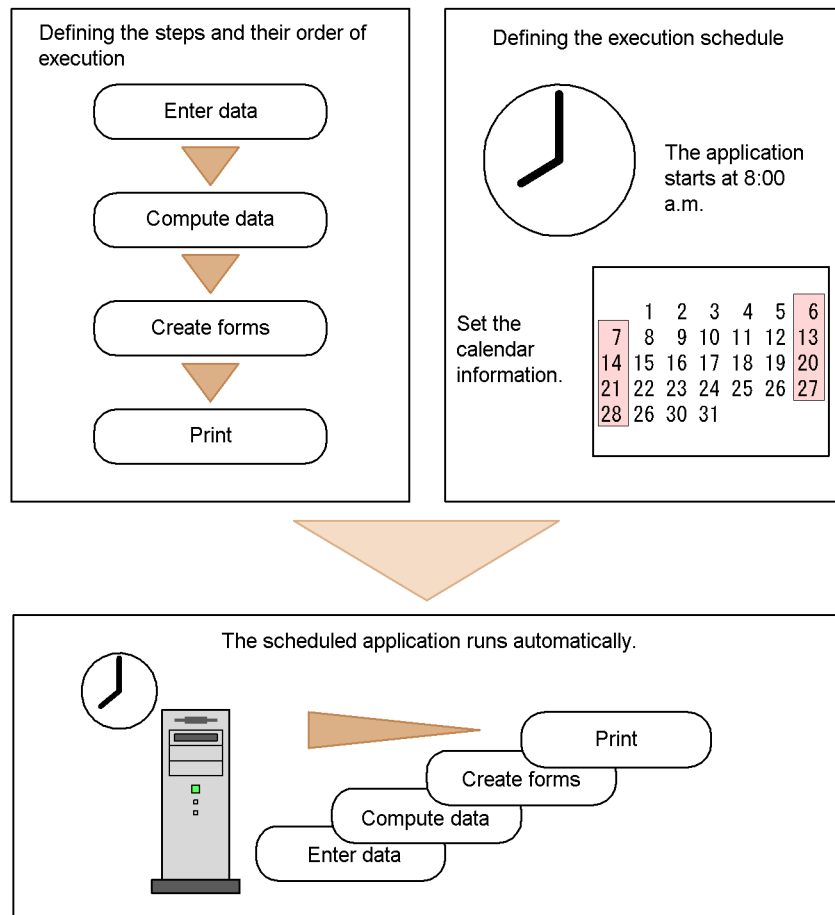
1.1.1 Automating an application

To run an application automatically using JP1/AJS3, you must first define the following information:

- The steps involved and their order of execution
- The execution schedule or an event that initiates the application

The following figure provides an overview of automating an application.

Figure 1-1: Overview of automating an application

**(1) Defining the steps in an application and their order of execution**

Many applications involve a set sequence of steps. For example, creating sales forms might involve the following steps:

1. Data entry
2. Computation
3. Form creation
4. Printing

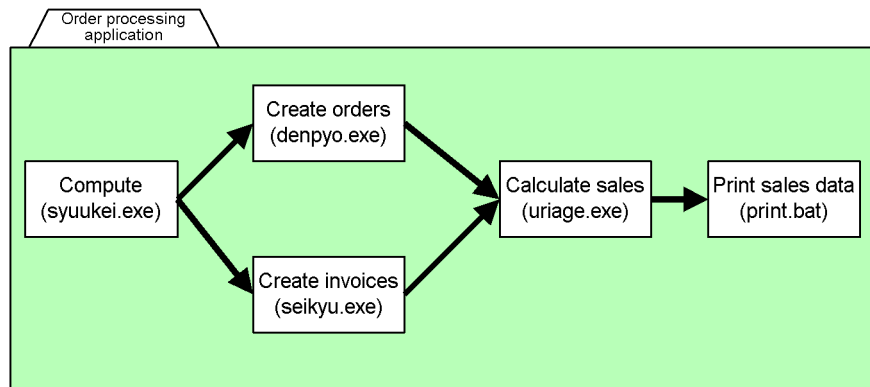
To run an application under JP1/AJS, you divide it into separate steps and define their execution sequence. Each command, program, or shell script is regarded as a separate step. In JP1/AJS3, this smallest element of processing is called a *job*. Multiple jobs

defined in a set sequence are known as a *jobnet*.

The steps in an application do not always form a simple linear sequence. Their order of execution is determined by a variety of conditions. For example, multiple steps may be executed in parallel, or a subsequent process may differ depending on whether the computed data is greater or less than a base value. You create a jobnet by refining these conditions and working out the application flow.

The following figure shows an example of an application flow.

Figure 1-2: Application flow



(2) Defining an execution schedule

To run an application automatically, you must first set a schedule that defines when the application is to start.

Day-to-day business applications run according to various schedules. Some run on a daily basis, some only at the end of the week or on a specific day of the month, depending on the nature of the application. Some applications need to be performed in the morning, some at the end of the business day or at night. Using JP1/AJS3, you can define schedules like these, so that the date and time of execution are set automatically.

In defining a schedule, you set up a calendar that shows the company's working days and non-working days. You then define the date and time for running the application, or the interval at which it is to be executed. Based on these definitions, JP1/AJS3 sets the execution schedule and starts the application automatically when the specified time arrives.

JP1/AJS3 provides ways of refining scheduling to match actual operations in the workplace. For example, you can set the following rules:

- If the scheduled execution date falls on a holiday, execute the job on the preceding day.
- If a job could not begin at the scheduled time because of a delay with the previous

job, cancel that particular run.

(3) Registering conditions to start execution

As well as starting an application at a specific time, you can start an application when a specific situation occurs - for example, when a file is created or a particular event takes place. When you register file creation or some other event as the start condition, the application will start running when that situation occurs.

The condition that you define to start an application may occur several times a day, not necessarily once a day in an invariable pattern. If a condition is likely to apply more than once a day, you must work out in advance whether to execute the processes in parallel or in succession.

1.1.2 Centralized management in a manager/agent configuration

JP1/AJS3 enables centralized management on a single host, even when applications are linked and executed on multiple hosts. The central host is called the *manager host*. The hosts that actually execute the applications are called *agent hosts*.

The manager host manages application sequencing and scheduling. When the set run-time arrives, or when an event that triggers an application takes place, the manager host requests the agent host to execute the application. The manager host also receives and manages the execution result.

In situations where a process runs on agent host A, followed by a process that runs on agent host B, the manager host manages the execution sequence.

Because multiple agent hosts can be centrally managed by a manager host, it is easy to manage large-scale systems.

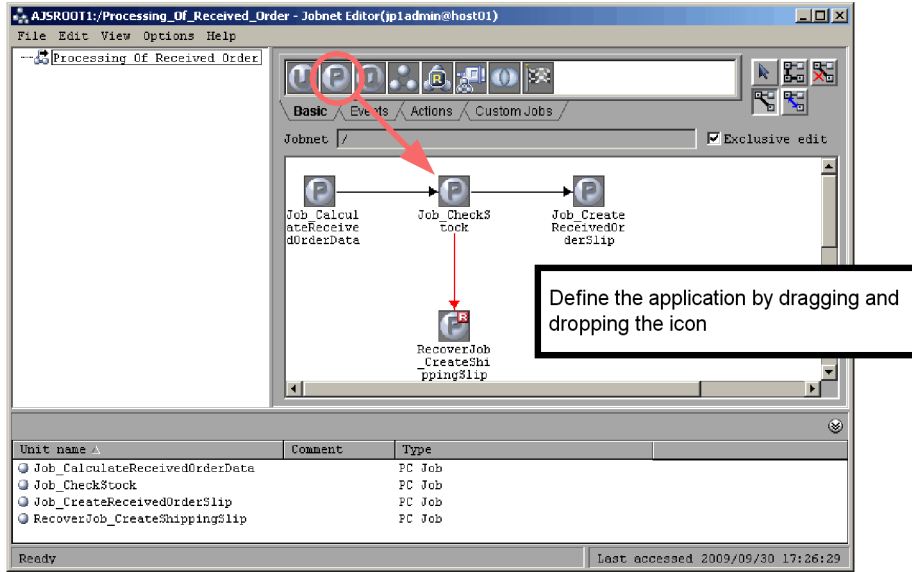
1.1.3 GUI-based definition and monitoring

In JP1/AJS3, you can define and monitor applications visually using the GUI. Application definition and execution, status monitoring and checking of execution results can all be performed from the windows on a single host, even if the manager host and agent hosts are in different locations.

You can define applications by dragging and dropping icons, using the intuitive interface.

The following figure shows an example of defining an application using the GUI.

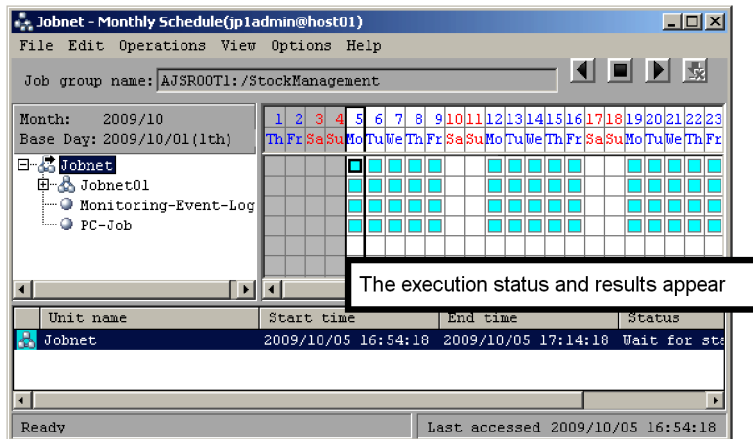
Figure 1-3: Defining an application using the GUI



When you register an application, the execution schedule appears in the GUI. Once the application starts, its execution status and results appear. Color-coding is used to represent the application's changing status, so you immediately notice any errors or problems.

The following figure shows an example of monitoring execution status using the GUI.

Figure 1-4: Monitoring execution status using the GUI

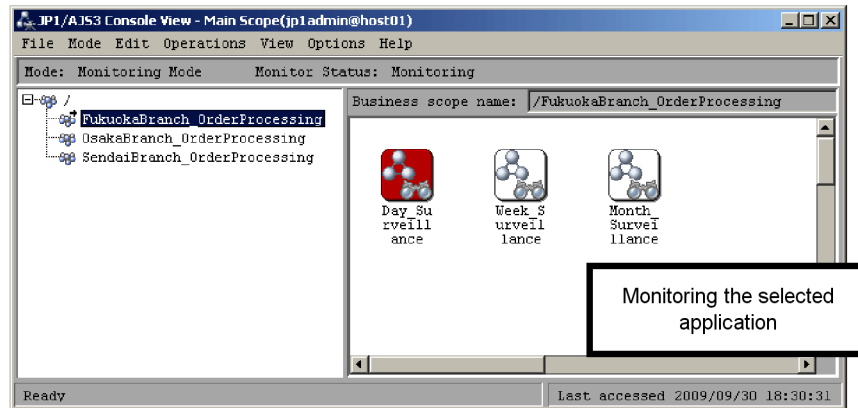


You can monitor all the applications running on a host. Alternatively, you can monitor

a selected application running on multiple hosts. When an application is being executed at branch offices in other locations, you can view and monitor that application on a single machine.

The following figure shows an example of monitoring an application.

Figure 1-5: Monitoring an application

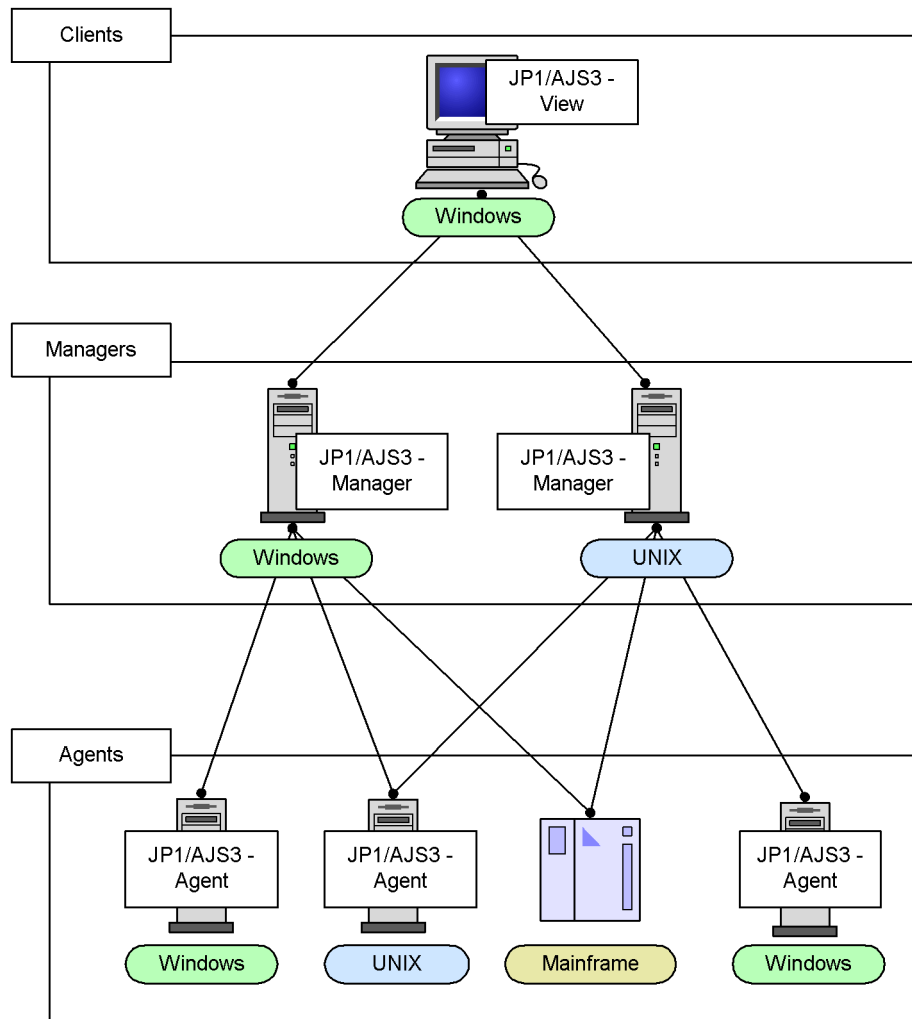


1.1.4 Support for a wide range of systems

JP1/AJS3 runs under Windows and under all UNIX operating systems. The integrated user interface enables transparent operation whatever system you are using. By linking JP1/AJS3 with other products in the JP1 series, you can manage jobs in a multi-platform environment that may include a mainframe.

The following figure shows an example of using JP1/AJS3 in a multi-platform environment.

Figure 1-6: Using JP1/AJS3 in a multi-platform environment



You can build your system in a flexible way, geared to the size of the applications. JP1/AJS3 supports everything from a small system in which all settings and job runs are performed on a single host, right up to large-scale systems in which execution of applications is distributed over dozens of hosts.

JP1/AJS3 can be linked with the JP1 series and with other programs. JP1/AJS3's flexible scheduling and integrated monitoring functions can be utilized in running other programs by defining the external processing as part of the JP1/AJS3 processing.

Execution status can also be managed from a system management program that is linked to JP1/AJS3, such as JP1/IM or HP NNM.

1.1.5 Types of applications that can be automated

Using JP1/AJS3, you can automate the following types of applications:

- Sales application: Calculate daily sales at each branch office at the end of the day, and count up the sales total for all the branch offices at head office at night. Update the database when the computations are completed. Do not update the database if the computations fail.
- Order processing application: During the day, automatically update the database and print an order entry form only when the operator enters an order and price. At night, register the data entered during the day into separate databases created for each of the items on the order entry forms.
- Accounting application: Create and print account forms automatically on the 5th, 10th, 15th, 20th, and 25th of every month. If the date falls on a non-working day, hold the processing over until the following day.

1.1.6 Benefits of deploying JP1/AJS3

The following benefits can be gained by deploying JP1/AJS3:

- Fewer personnel needed to run applications
Work can be automated, and applications can be executed and managed from a single host, even in a distributed system with hosts in different locations. Since fewer people are needed to run applications, costs are reduced and human resources can be utilized effectively.
- Reliable operation with fewer mishaps
Routine tasks can be automated, reducing the risk of operator error.
- Prompt error response even in an unattended system
JP1/AJS3 provides means of responding to errors automatically. For example, the administrator can be informed by email if an error occurs, or a process can be defined for execution only in the event of an error. These procedures enable prompt error response even when the system is unattended.

1.2 Functions provided in JP1/AJS3

JP1/AJS3 controls job execution using a number of functions that work together. This section summarizes the range of functions provided in JP1/AJS3.

1.2.1 Overview of JP1/AJS3 functions

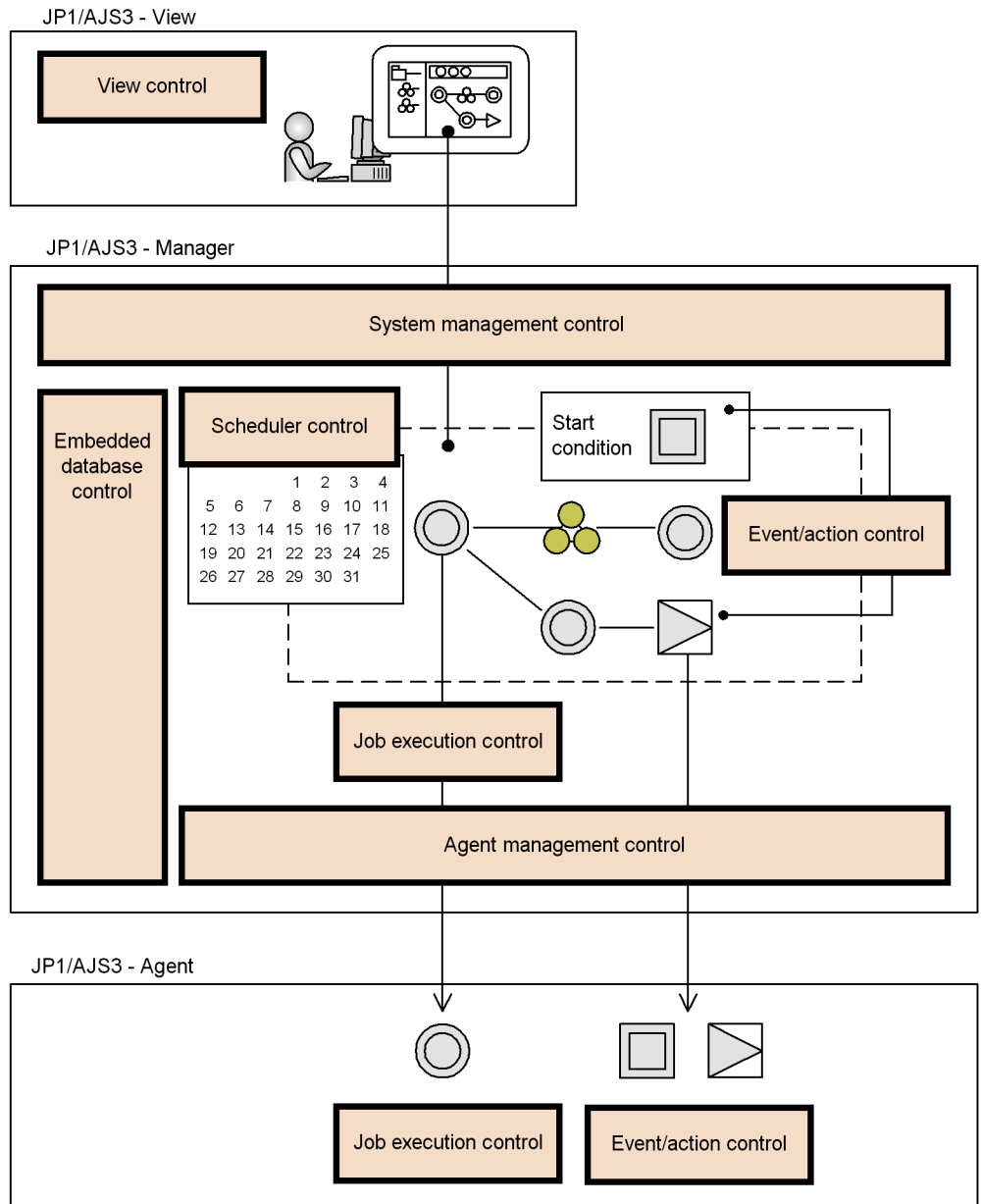
The following table summarizes the programs and functions that constitute JP1/AJS3. JP1/AJS3 uses these functions in combination to control job execution.

Table 1-1: JP1/AJS3 functions

Program	Function	Summary
JP1/AJS3 - Manager	System management control	<ul style="list-style-type: none"> Management of services, processes, and environment settings
	Scheduler control	<ul style="list-style-type: none"> Execution schedule management Execution sequence control using a jobnet
	Agent management control	<ul style="list-style-type: none"> Execution agent management
	Job execution control	<ul style="list-style-type: none"> Job transfer and status management (Manager) Job execution (Agent)
	Submit job execution control	<ul style="list-style-type: none"> Job management using queuing (Manager)
	Event/action control	<ul style="list-style-type: none"> Management of event monitoring status (Manager) Event monitoring (Agent)
	Embedded database control	<ul style="list-style-type: none"> The relational database that stores jobnet definitions, execution agent definitions, and other information
JP1/AJS3 - Agent	Job execution control	<ul style="list-style-type: none"> Job execution (Agent)
	Event/action control	<ul style="list-style-type: none"> Event monitoring (Agent)
JP1/AJS3 - View	View control	<ul style="list-style-type: none"> Job and jobnet definitions Job and jobnet scheduling and execution status display

The following figure shows the functions provided by JP1/AJS3 and their inter-relationships.

Figure 1-7: JP1/AJS3 functions



1.2.2 Function inter-relationships

JP1/AJS3 enables planned job execution. You can run a job at a scheduled date and

time, or run a number of jobs in a set sequence. Jobs can also be executed automatically, driven by a specific event such as a file update or JP1 event.

To control job execution, the JP1/AJS3 functions work together as follows.

To control job execution:

1. Determine the job execution time (scheduler control).
 - Issue an instruction for job execution at a scheduled time.
 - Execute jobs in a predetermined sequence (jobnet).
2. Specify job execution on receipt of a specific event (event/action control).
 - Issue an instruction for job execution in response to a file update or JP1 event.
3. Execute jobs (job execution control).
 - Actually execute the job, on instruction from the scheduler and the event/action control.

The following example illustrates the way in which these functions work together.

Example: Execute a jobnet with a set start condition, already submitted for planned execution.

To execute a jobnet with a set start condition:

1. Determine the execution date and time according to the execution schedule calculated from the calendar and processing cycle.
Start executing the jobnet when the execution time arrives.
2. Wait to receive an event set as a start condition.
Start executing the jobnet when the event is received and the start condition is satisfied.
3. Execute each of the jobs defined in the jobnet according to their execution sequence.

The job status changes as follows: *Waiting to execute* → *Now running* → *Ended normally*. Each job is executed in turn, in accordance with this status transition.

4. As the jobs are executed, the process including the number of jobs being executed concurrently is controlled by the execution agent.

Each job is executed on the agent host mapped to an execution agent.

Working together in this way, the JP1/AJS3 functions control the timing for initiating each job according to the set execution time and sequence (1, 2, and 3). The execution

agents provide an efficient job execution environment (4).

Each of the JP1/AJS3 functions is responsible for a specific type of processing. These processes work in concert to execute jobs.

The JP1/AJS3 functions are described individually below.

1.2.3 System management control

System management control is a core function for managing JP1/AJS3 services, processes, and environment settings. It controls the job management system as a whole, including starting and stopping processes gracefully in accordance with the environment settings. It performs process monitoring, and restarts processes if a problem occurs.

1.2.4 Scheduler control

(1) *Execution order control*

JP1/AJS3 executes applications according to a defined sequence of jobs and jobnets. As each job or jobnet finishes, the next process starts running automatically, according to the jobnet definitions.

Using this feature, you can change subsequent processing, depending on the result of the preceding job. You can also define a job to be executed only if the preceding job terminates abnormally.

(2) *Execution schedule management*

JP1/AJS3 executes processing according to the schedule defined for a jobnet.

A schedule consists of a calendar that defines working days and non-working days for running the application, together with schedule rules that define the execution start time and processing cycle. JP1/AJS3 determines the jobnet's execution schedule from these two sets of information.

You can also define an execution trigger (event) for an application. The application will start running whenever that event occurs.

1.2.5 Agent management control

Agent management control manages logical entities called *execution agents*. Each execution agent has a name assigned by JP1/AJS that corresponds to the name of the agent host that is to execute the job. Keeping the name of the agent host separate from the job definition provides greater portability of job definitions. By using execution agent, you can set restrictions on concurrent execution and job transfer, enabling the number of jobs handled by an agent host to be controlled according to its processing load or the time of day.

Two or more execution agents can be grouped in an *execution agent group*. This allows job execution to be distributed among multiple agents.

1.2.6 Job execution control

Job execution control executes jobs using execution agents. When a job is executed, job execution control references information about the maximum number of concurrently executable jobs, the job transfer restriction status, and the agent host name set for the execution agent.

Job execution control places jobs temporarily in memory in the order in which they were registered for execution. It passes the jobs one by one to the agent host, keeping watch on the number of jobs being executed concurrently. The agent host executes the jobs as each is received.

1.2.7 Event/action control

Processing can be executed automatically at a specific occurrence, such as receipt of an email message or file creation or update. An occurrence used to trigger execution of a job is called an *event*.

In JP1/AJS3, you can execute a specific process, such as sending an email message or JP1 event, as a status or termination notification when the preceding job completes. This is called an *action*.

The main types of events and actions available in JP1/AJS3 are as follows:

- Events
 - Register a JP1 event
 - Receive an email message
 - Create or update a file
 - Write a message to a log file
 - Write an event message to the Windows event log
- Actions
 - Send a JP1 event
 - Send an email message
 - Power on or power off the local host or a remote host

1.2.8 Embedded database control

Embedded database control uses a relational database that stores various JP1/AJS3 information, such as definition information about jobnets and execution agents. It operates transparently while an application is running, and is a core function of the job management system. Using transaction management, embedded database control maintains the integrity of the stored data even if an error occurs.

1.2.9 View control

Using the JP1/AJS3 - View functions, you can execute jobs and jobnets, and view job and jobnet schedules and execution results, via the GUI. Colored icons are used to show the execution status and results.

1.2.10 Other functions

(1) *Submit job control*

Submit job control manages the following jobs that were managed under the job execution control in version 8 and earlier versions:

- Submit jobs executed using the `jpqjobs` command
- QUEUE jobs registered and executed from a queue
- Jobs executed from another system such as JP1/OJE

Submit job control uses queues and agents defined separately from those governed by the agent management control. Their configuration definition information, and the job information registered for execution with the submit job control, is managed in an ISAM database.

For the setup required to use submit job control, see *7.1.2 Setup for using QUEUE jobs and submit jobs* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

(2) *Queueless job execution control*

This feature executes queueless jobs (jobs that are not submitted to a queue). When registered for execution, a queueless job is passed directly to an agent, bypassing the manager host's job execution control.

(3) *JP1/AJS3 Console control*

Using the JP1/AJS3 Console functions, you can centrally monitor the status of multiple applications being executed in JP1/AJS3. Applications controlled by a number of managers can be monitored from a single machine.

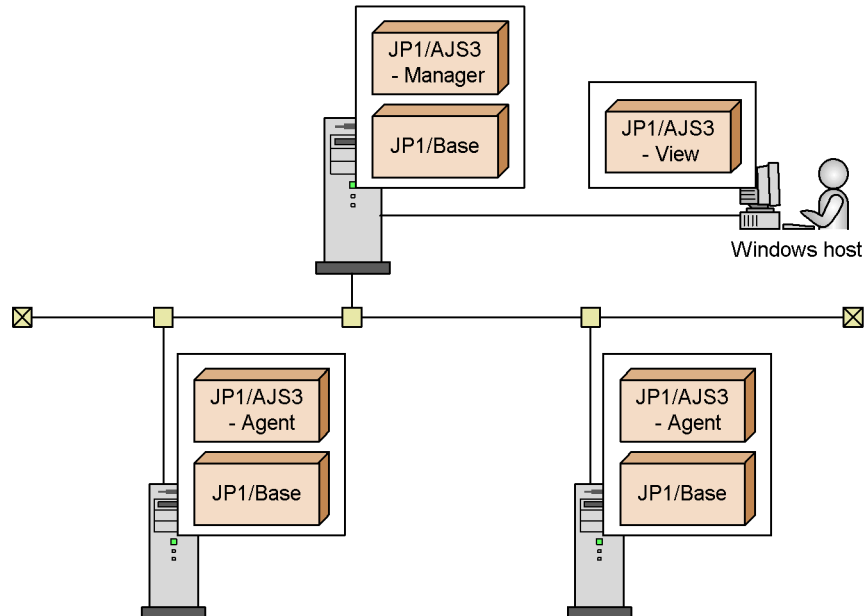
The status of the applications is obtained by the manager components, and is displayed via the JP1/AJS3 - View component.

1.3 JP1/AJS3 program configuration

This section describes the programs that make up a JP1/AJS3 system.

The following figure shows how a JP1/AJS3 system is configured.

Figure 1-8: JP1/AJS3 program configuration



JP1/AJS3 - Manager

JP1/AJS3 - Manager manages jobnet and schedule definitions, and controls the sequence in which jobnets are executed. JP1/AJS3 - Manager requests the JP1/AJS3 - Agents to execute jobs, and manages the information received from the JP1/AJS3 - Agents about each job's execution status and execution result.

Because JP1/AJS3 - Manager incorporates both agent functionality and job execution control functions, it can also execute jobs as an agent itself.

JP1/AJS3 - Agent

JP1/AJS3 - Agent executes jobs on request from JP1/AJS3 - Manager, and sends information about the execution status and execution result back to JP1/AJS3 - Manager.

JP1/AJS3 - View

JP1/AJS3 - View provides a GUI for working with jobnets and schedules. It can be linked with JP1/AJS3 - Manager for operations such as editing jobnet and schedule definitions and executing jobnets. The user can also view the status and results of executed jobs using JP1/AJS3 - View.

JP1/Base

JP1/Base is a JP1/AJS3 prerequisite program, used for managing user permissions and for sending and receiving JP1 events.

1.4 JP1/AJS3 database configurations

JP1/AJS3 - Manager uses an embedded database as standard for managing information about jobs and jobnets. However, an ISAM database can be used for reasons of compatibility with a previous version or if required by the operating environment.

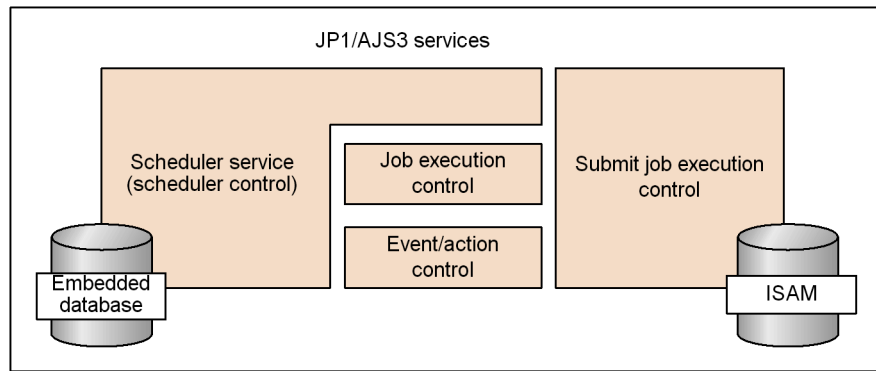
JP1/AJS3 - Manager supports three types of database configurations:

- Standard configuration
- Standard configuration without ISAM
- Compatible ISAM configuration

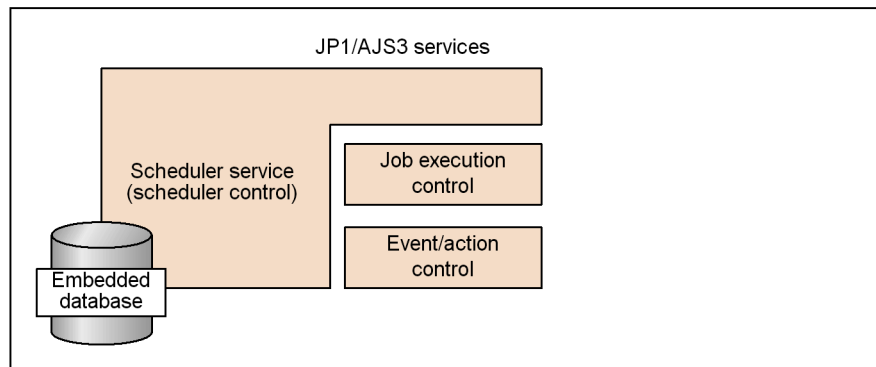
The following figure shows the database configurations supported by JP1/AJS3 - Manager.

Figure 1-9: JP1/AJS3 - Manager database configurations

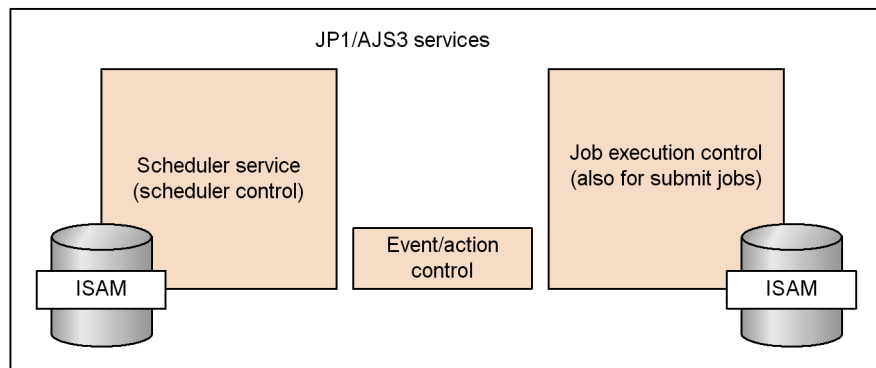
■ Standard configuration



■ Standard configuration without ISAM



■ Compatible ISAM configuration



Each database configuration is described next.

(1) Standard configuration

In the standard configuration, all JP1/AJS3 - Manager functions are available.

The standard configuration applies when you perform:

- A new installation of JP1/AJS3 - Manager
- Setup after an upgrade installation from JP1/AJS2 - Manager to JP1/AJS3 - Manager

For the setup procedure after upgrading to JP1/AJS3 - Manager, see *8.5 Setup after upgrading from JP1/AJS2 - Manager to JP1/AJS3 - Manager* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

(2) Standard configuration without ISAM

The standard configuration without ISAM is the same as a standard configuration, but without the submit job control function. Because there are no ISAM files, periodic re-organization of the database is unnecessary.

You cannot perform the following operations using a standard configuration without ISAM:

- Execute QUEUE jobs and submit jobs on the local host from JP1/AJS3 - Manager on the local host or from JP1/AJS2 - Manager or JP1/AJS3 - Manager on a remote host.
- Set up the execution environment for QUEUE jobs and submit jobs.
- Execute commands used in the execution environment for QUEUE jobs and submit jobs (jpcqxxx commands).

For details about the commands used in the execution environment for QUEUE jobs and submit jobs, see *1.5.9(2) Commands available in an execution environment for queue or submit jobs* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2*.

- Execute jobs on the local host from JP1/AJS2 - Client ToolKit or P1/AJS2 - SDK on a remote host.
- Execute jobs on the local host from another system such as JP1/NQSEXEC or JP1/OJE for VOS3.

For the setup required to change from a standard configuration to a standard configuration without ISAM, see *6.6.3 Changing the JP1/AJS3 database configuration to the standard configuration without ISAM* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or see *14.7.2 Changing the JP1/AJS3 database configuration to the standard configuration without ISAM* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

As a variant of a standard configuration, the standard configuration without ISAM is included in the term *standard configuration* unless otherwise stated.

(3) Compatible ISAM configuration

The compatible ISAM configuration supports the previous JP1/AJS2 - Manager.

You cannot perform the following operations using a compatible ISAM configuration:

- Manage execution agents and execution agent groups.
- Switch a jobnet definition at a specified time while the jobnet is running (jobnet release function).
- Display the Summary Monitor window in JP1/AJS3 - View.
- Specify search conditions, other than the unit name or unit execution status, when searching for units in JP1/AJS3 - View.
- Use commands that operate on the embedded database, other than the command for setting up a standard configuration.

If you were using ISAM files in the scheduler database with JP1/AJS2 - Manager, after you upgrade to JP1/AJS3 - Manager, the database configuration will be the compatible ISAM configuration. Although this will work, we recommend changing to the standard configuration. For the procedure to change the database configuration, see 8.5 *Setup after upgrading from JP1/AJS2 - Manager to JP1/AJS3 - Manager* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

The descriptions in this manual assume a standard database configuration unless otherwise stated. If you are using the compatible ISAM configuration, see your JP1/AJS2 manual.

Chapter

2. Automating Applications Using JP1/AJS3

This chapter describes the preparations required for automating applications using JP1/AJS3.

2.1 Preparing to automate applications

2.1 Preparing to automate applications

This section describes the prior considerations, and the definitions and settings required for automating applications using JP1/AJS3.

2.1.1 Application assessment

To automate operations using JP1/AJS3, you must first determine which applications to automate. You must then consider the types of operational schedules under which they will run.

You need to consider the following points:

- Appropriate applications for automation
- Processes required for automation
- Conditions for starting execution
- Working and non-working days
- Execution time and execution cycle
- JP1/AJS3 users and their access permissions

(1) *Selecting applications for automation*

Decide which applications to automate. Using JP1/AJS3, you can automate not only routine applications that run on a set daily or monthly cycle, but also applications in which the processing changes dynamically depending on the results of the preceding process. JP1/AJS3 also supports automation of irregular or ad hoc applications executed when some event occurs.

Examples

- Create and print account forms automatically on the 5th, 10th, 15th, 20th, and 25th of every month. If the date falls on a non-working day, hold the processing over until the following day.
- Calculate daily sales at each branch office at the end of the day, and count up the sales total for all the branch offices at head office at night. Update the database when the computations are completed. Do not update the database if the computations fail.
- During the day, automatically update the database and print an order form only when the operator enters an order and price. At night, register the data entered during the day into separate databases created for each of the items on the order forms.

(2) Determining the processes involved in the application

Work out the processes needed to run a selected application automatically. Consider the program in which each process runs and their execution sequence. Also consider the access permissions needed at the processing side: Determine which users are allowed access, and whether they can execute, or also define and edit, each process.

Example

Preparation of a daily report: At 6 p.m. every day, calculate sales data from the total number of orders and purchases that day, and collate the results in a daily report.

Required processes

- Process to calculate the total number of orders received (program: `juchu.exe`)
Calculate total orders and output the result to a file.
- Process to calculate the total number of purchases (program: `shiire.exe`)
Calculate total purchases and output the result to a file.
- Process to create a daily report (program: `nippou.exe`)
Calculate sales data from the total orders and total purchases, and collate in a daily report.

Process execution sequence

It does not matter whether the orders calculation or the purchases calculation is performed first. Therefore, run the two processes in parallel. To prepare the daily report, the sales data has to be computed from the results of the first two calculations. Therefore, run the report creation process after both the orders calculation process and purchases calculation process have completed.

Process permissions

Permit only the system administrator and the manager and operators in the Sales Department to access the automated application. The Sales Department manager can define and edit processes; operators are permitted only to run processes.

Other requirements and considerations

- End judgments
Consider the threshold for determining the end status of each process. For example, normal terminations return a return value is 0, warning terminations return return values 1 to 4, and abnormal terminations return return value 5 or above.

- Error recovery
Consider what processing to execute in the event of an error.
- Delay monitoring
Monitor for any delays in starting or ending a process that would cause difficulties.
- Process timeout and termination
Consider the sorts of situations in which a process should be forcibly terminated, and what action to take after termination.
- Job rerun
Consider the sorts of situations in which to rerun the job (following an error or other problem, for example). Also consider which process the rerun should be started from.

(3) Setting an operations calendar

Define a calendar for JP1/AJS3 operations. Like an ordinary calendar that shows weekdays, Sundays, and public holidays, in JP1/AJS3 you can set enterprise-specific or application-specific working days (*open days*) and non-working days (*closed days*). You can also specify various information appropriate to the processing being carried out. For example, you can specify a month by start and end dates, and you can set the time at which the working day begins. You must therefore consider how the processes in the application are to be carried out against these operational requirements.

Examples

Closed days and open days

- No work is carried out on Saturdays, Sundays, or public holidays. Set these as *closed days*.
- Saturdays, Sundays, and public holidays are closed days at the company, but you want to execute a particular process on a Saturday. So, make Saturday an *open day*.
- Saturdays, Sundays, and public holidays are closed days as a general rule, but the second Saturday in each month should be an *open day*.

Monthly start day and daily start time

- Accounts are settled on the 25th of each month. Therefore, set the 26th as the first day of the month. For example, regard the period from August 26 to September 25 as the month of July.
- A process will actually start at 6:00 a.m. the next day. But you want the system to handle that process as completed on the current day. So, consider a day as running from 7:00 a.m. until 6:59 a.m. the following day.

(4) Determining the execution time and execution cycle

Consider the date and time at which the application is to start, and the cycle at which it is to be repeated. For an ad hoc process without a set start time, consider the criteria governing when the process is to start.

Examples

Execution time and execution cycle

- Start execution on a specific day of the year (August 10, 2009).
- Start the process at 9:00 a.m. every day.
- Start the process at 5:00 p.m. every Friday.
- Start the process at 8:00 a.m. on the second Monday of every month.
- If the date of execution falls on a closed day, execute the process on the following day instead.

Start conditions

- Execute the process whenever a data file is updated.
- Execute the process 10 minutes after the preceding process has completed.
- Start monitoring the file at 17:00 every Friday and execute the process whenever the file is updated.

(5) Selecting JP1/AJS3 users and their access permissions

Select the users who are able to use JP1/AJS3 (JP1 users). Consider the operating privileges (access permissions) to be granted to each user for the application in question.

To set access permissions, select them from the permissions already set up in JP1/AJS3. The following access permissions are used:

- Administrator permission, and permission to define, execute, and edit a *jobnet*
- Permission to define, execute, and edit a jobnet
- Permission to define and edit a jobnet
- Permission to execute and reference a jobnet
- Permission to reference a jobnet

Examples

- The work flow controller in the Accounting Department is allowed to define and modify departmental jobs, but not to execute them. The controller has no authority to operate on jobs related to the Personnel Department.
- Operators in the Accounting Department are allowed to execute

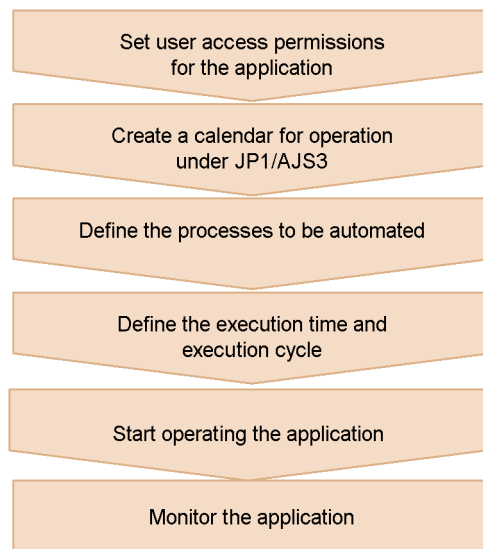
departmental jobs, but not to define or modify them. Operators are permitted only to reference jobs related to the Personnel Department.

- The work flow controller in the Personnel Department is allowed to define and modify departmental jobs, but not to execute them. The controller has no authority to operate on jobs related to the Accounting Department.
- Operators in the Personnel Department are allowed to execute departmental jobs, but not to define or modify them. Operators are permitted only to reference jobs related to the Accounting Department.
- Employees in the Sales Department are permitted only to reference departmental jobs.
- The system administrator has all permissions for all jobs regardless of which department they belong to.

2.1.2 Procedures for deploying JP1/AJS3

Having considered what processes to automate in an application, complete the job definitions and settings. You can then start using JP1/AJS3. The following figure shows the procedures for automating applications using JP1/AJS3.

Figure 2-1: Procedures for automating applications using JP1/AJS3



(1) Set user access permissions for the application

On the authentication server (JP1/Base), register the users who will use JP1/AJS3. For each user, define his or her access permissions to the application.

(2) Create a calendar for operation under JP1/AJS3

Create a JP1/AJS3 calendar (open days and closed days). Complete other settings, including the monthly start date (base day) and the daily start time (base time).

(3) Define the processes to be automated

Define each process in the automated application as a separate job. Set their order of execution, and build the processes into a jobnet.

(4) Define the execution time and execution cycle

Define the schedule rules for the jobnet, including the execution start time and execution cycle, according to the application schedule. If the application has no predetermined start time, define a start condition.

(5) Start operating the application

Register the created jobnet for execution and start operation. Once the jobnet has been registered, it is scheduled according to the settings and runs automatically.

(6) Monitor the application

Monitor the execution status of the registered jobnet and check the execution result and execution schedule.

Chapter

3. Defining Automated Applications

This chapter describes how to define jobnets, calendar information, schedules, and other information for automating applications under JP1/AJS3.

- 3.1 Hierarchical structure of the job network
- 3.2 Defining a calendar for JP1/AJS3 operation
- 3.3 Defining a schedule
- 3.4 Defining a start condition

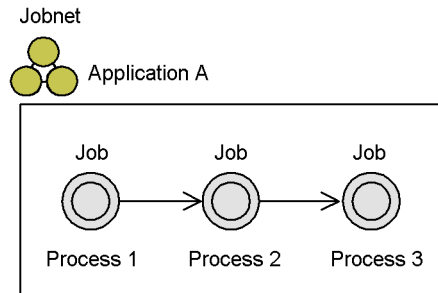
3.1 Hierarchical structure of the job network

In JP1/AJS3, the elements in an application to be automated are known as *units*.

Each of the individual processes involved in an application is defined as a unit called a *job*. A job is the smallest unit. You then arrange the defined jobs in order of execution, creating a network of jobs grouped together to form a unified application. This collection of jobs is called a *jobnet*.

The following figure shows the relationship between a jobnet and jobs.

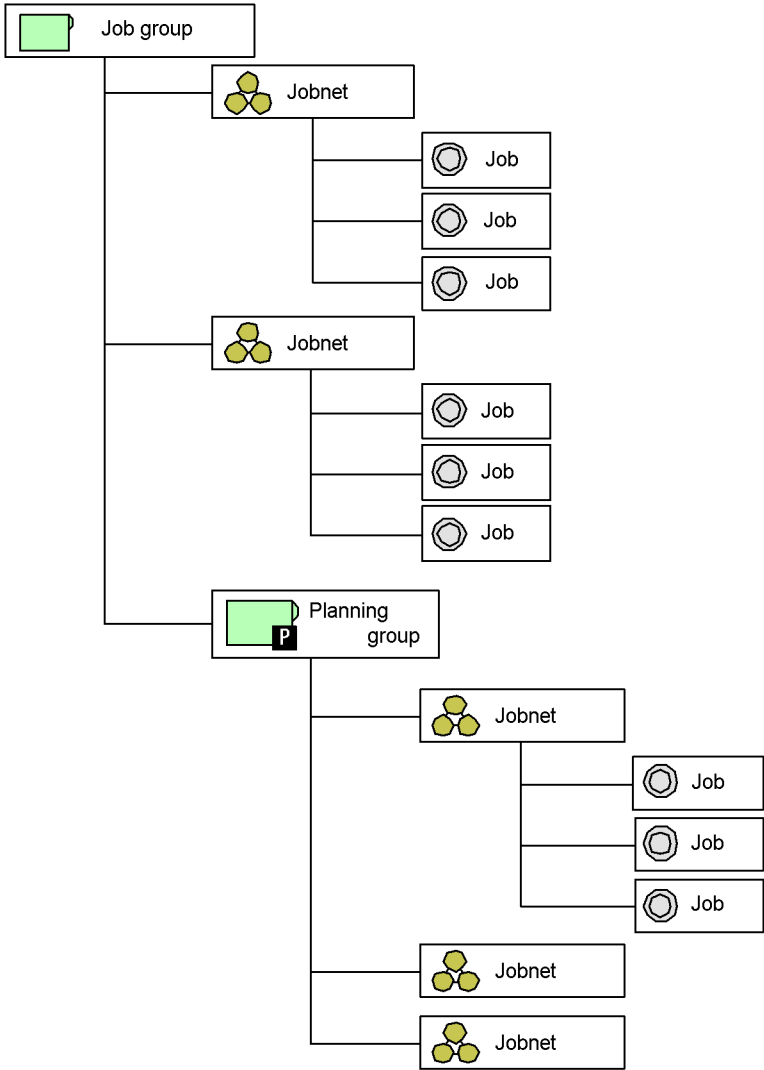
Figure 3-1: Jobnet and jobs



Jobnets themselves can be grouped into management units called *job groups* and *planning groups*.

This approach allows all the processing in an automated application, right down to most detailed individual processes, to be managed systematically. The management levels of a job network are illustrated in the following figure.

Figure 3-2: Job network management hierarchy



The entire set of units is managed as a control unit called a *scheduler service*. By default, the scheduler service is defined as the root job group (top-level job group) or `AJSROOT1`, which manages all units defined below it.

Each scheduler service can have its own environment. This means that you can set up a test environment for a new application. By launching multiple scheduler services concurrently, you can run multiple independent applications (jobnets) in parallel, each under a different root job group.

3.1.1 Units

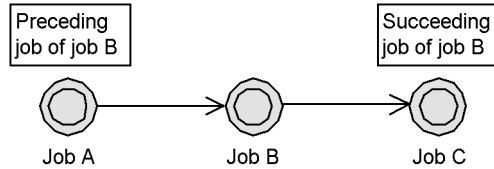
This section describes the various units.

(1) Jobs

A *job* is the smallest of the units in an automated application. In JP1/AJS3, you define an application by arranging a number of processes in order of their execution. Each of these processes corresponds to a job.

Each job is arranged in execution sequence as shown in the following figure.

Figure 3-3: Preceding job and succeeding job



In this example, job A is the *preceding job*, and job C is the *succeeding job*, of job B. For details on sequencing jobs, see 3.1.3 *Creating a job flow*.

You can set a hold attribute, a type, an execution service, a timeout period, information about the end delay monitoring of a job, an owner, a JP1 resource group, and an execution user type. Hold attributes, types, execution services, timeout periods, and end delay monitoring are described below. For details about owners, JP1 resource groups, and execution user types, see 8.2 *Settings for restricting access to units*.

- Hold

By setting this attribute, you can hold a job so that it will not be executed until its held status is released.

- Type

You can specify whether to use a unit as a recovery unit. Select either **Normal** or **Recovery** as the unit type. The default is **Normal**.

A job whose type is set to **Recovery** is called a recovery job and is executed if a preceding unit has terminated abnormally. A jobnet whose type is set to **Recovery** is called a recovery jobnet. Recovery jobs and recovery jobnets are collectively called recovery units.

- Execution service

You can specify a service on which the job is to be executed. Specify either **Standard** or **Queueless Agent**. The default is **Standard**. For a queueless job, usually use **Standard** for the execution service. If **Queueless Agent** is specified, an execution agent or execution agent group cannot be used.

For details about execution agents, see *5. Job Execution Environments*. For details about queueless jobs, see *10.5 Queueless jobs*.

■ **Timeout period**

By setting a timeout period, you can cancel execution based on the elapsed time since the job began execution. For example, if you set a timeout period of 10 minutes, the job will be canceled if it has not ended 10 minutes after it started. Set the timeout period in the range 1 to 1,440 (minutes).

■ **End delay monitoring**

By setting the time required to run a job, you can monitor for end delays based on the elapsed time since the job began execution. Set the time required for execution in the range 1 to 1,440 (minutes). For example, if you set a time required for execution of 10 minutes, an end delay will be detected if the job has not ended 10 minutes after it started. On detection of an end delay, the job is placed in *Running (end-delay)* status. When the job terminates, its status becomes *Ended normally (end-delay)* or other status as appropriate. Message KAVS0248-I is output on detection of a delay. For the message text, see *2.2 Messages beginning with KAVS (Messages about the scheduler)* in the manual *Job Management Partner 1/Automatic Job Management System 3 Messages 1*. For the output destination and the output conditions, see *1.3(2) Output destinations of messages beginning with KAVS* in the manual *Job Management Partner 1/Automatic Job Management System 3 Messages 1*.

End delay monitoring cannot be specified for OR jobs or judgment jobs.

For details about monitoring end delays in job execution, see *5.1(3) End delay monitoring based on time-required-for-execution* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

Because there are many different types of jobs, you must sequence the jobs that you are defining according to the type of process required. JP1/AJS3 provides the following job types:

- Standard job
- OR job
- Judgment job
- Event job
- Action job
- Custom job

The characteristics of each type of job are described below.

(a) Standard job

For a *standard job*, you define an executable file and specify the host on which the job is to run. There are three types of standard jobs:

- Unix job
- PC job
- QUEUE job

The following table outlines each standard job type and the processes (specifiable files) that you can define.

Table 3-1: Standard jobs and definable processes

No.	Job type	Description	Definable processes
1	Unix job	Process executed on a UNIX host.	<ul style="list-style-type: none"> • Executable file • Shell script
2	PC job	Process executed on a Windows host.	<ul style="list-style-type: none"> • .exe file • .com file • .cmd file • .pif file • .bat file • .spt file[#] (script file created with JP1/Script) • Data file that has a file type (extension) associated with the application
3	QUEUE job	Process executed by submitting a job to a specified queue. Used when linking with another system such as JP1/NQSEXEC or JP1/OJE.	<ul style="list-style-type: none"> • Executable file • Shell script • .exe file • .com file • .cmd file • .pif file • .bat file • .spt file[#] (script file created with JP1/Script) • Data file that has a file type (extension) associated with the application

#

To execute an .spt file, JP1/Script must also be installed on the host running the job.

For details on defining a jobnet that uses standard jobs, see *2.4.1 Executing a process*

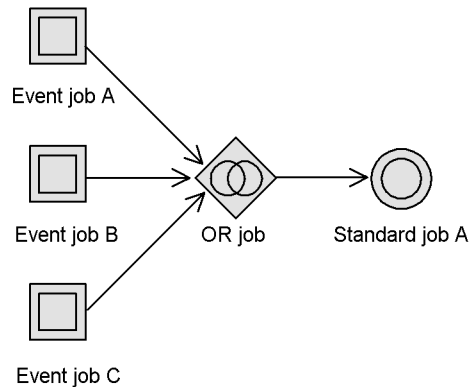
in a specified file (example of defining a jobnet consisting of standard jobs) in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

(b) OR job

For an *OR job*, you define a number of jobs (*event jobs*) which will monitor the system for a specific event. You define these event jobs as the preceding jobs of the OR job. If any one of the monitored events occurs, the OR job executes its succeeding job. Only event jobs can be defined as preceding jobs of an OR job.

The following figure shows an example of a jobnet containing an OR job.

Figure 3-4: Example of a jobnet containing an OR job



When one of the defined event jobs detects an event, that job terminates and the other event jobs are set to *Bypassed* status. Event monitoring ends at this point.

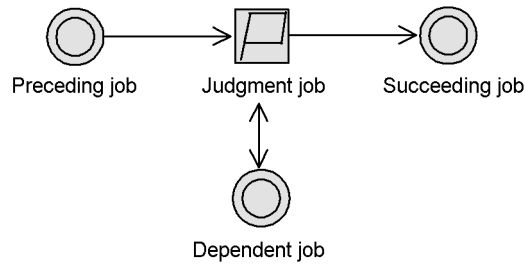
For details on defining a jobnet that uses an OR job, see 2.4.2 *Executing a process when one of multiple conditions is satisfied (example of defining a jobnet that uses an OR job)* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

(c) Judgment job

A judgment job checks whether a given condition is satisfied. A dependent job of a judgment job is called a *dependent job*, and the results of the judgment determine whether the dependent job is executed. In a judgment job, you can set a condition for executing a dependent job. If the condition is true, the dependent job is executed, followed by the succeeding job. If the condition is false, the succeeding job runs without execution of the dependent job. However, if the dependent job ends abnormally, the succeeding job is not executed.

The following figure shows an example of a jobnet that uses a judgment job.

Figure 3-5: Example of a jobnet that uses a judgment job



Three judgment conditions can be used in a judgment job:

- Judgment based on the return code of the preceding job

Determine whether to execute the dependent job by comparing the return code of the preceding job with a set judgment value.

The following conditions can be set:

- Return code $>$ judgment value
- Return code \geq judgment value
- Return code $<$ judgment value
- Return code \leq judgment value
- Return code = judgment value
- Return code \neq judgment value

- Judgment based on the presence of a file

Determine whether to execute the dependent job according to whether or not a specified file exists on the manager host.

The following conditions can be set:

- File exists
Execute the dependent job if the file exists.
- File does not exist
Execute the dependent job if the file does not exist.

When you specify a network file, the file is assumed to be non-existent if access to the network is not possible.

- Judgment based on variable comparison

Determine whether to execute the dependent job according to the information

inherited from the root jobnet or preceding job.

The passing information usable for judgments is as follows:

- The passing information set in event jobs
- The passing information set with macro variable values specified during registration for execution

You can select whether the value is to be treated as a numeric value or character string, for the variable value inherited from the preceding job, and for the condition value to be evaluated by the judgment job.

The conditions that can be set are as follows.

When *numeric* is selected:

- Is the variable value greater than the condition value?
- Is the variable value equal to or greater than the condition value?
- Is the variable value smaller than the condition value?
- Is the variable value equal to or smaller than the condition value?
- Is the variable value equal to the condition value?
- Does the variable value differ from the condition value?

When *string* is selected:

- Is the variable value equal to the condition value?
- Does the variable value include the condition value?
- Does the variable value differ from the condition value?
- Is the variable value present?
- Is the variable value absent?

When judgment is based on the return code of the preceding job, the judgment job requires a single preceding job and a single subordinate unit.

When a jobnet is defined as the preceding unit, the judgment result is always the same because the jobnet's return value is treated as zero. This is why, for a judgment based on the return code, the preceding unit must be a job.

When the judgment is based on the presence of a file or on variable comparison, the preceding unit does not have to be a job. You can define a judgment job at the beginning of a jobnet, or a nested jobnet as the preceding unit. However, when a judgment job that uses a variable comparison is defined at the beginning of a jobnet, the judgment result will always be false in the following cases because there is no value to be judged:

- When no macro variable value was specified during registration for execution
- When no macro variable has been defined in an event job that is used as a start condition for a jobnet

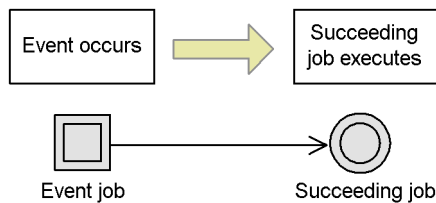
For details on defining a jobnet that uses a judgment job, see 2.4.3 *Dynamically changing a process depending on the result of a preceding job (example of defining a jobnet that uses a judgment job)* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

(d) Event job

An *event job* monitors an event occurring in the system. You can define an event job as the start condition for a job flow or jobnet, so that the particular job or jobnet will be executed only when the monitored event occurs. For details on jobnet start conditions, see 3.4 *Defining a start condition*.

The following figure shows an example of a jobnet that uses an event job.

Figure 3-6: Example of a jobnet that uses an event job



There are eight types of event jobs:

- Receive JP1 event job
- Monitoring files job
- Receive mail job
- Receive MQ message job
- Receive MSMQ message job
- Monitoring log files job
- Monitoring event log job
- Interval control job

The following table outlines each of these event jobs.

Table 3-2: Types of event jobs

No.	Event job name	Description
1	Receive JP1 event job	This job terminates when it receives a specific event from JP1/Base.
2	Monitoring files job	This job terminates when a specific file is created, deleted, or updated.
3	Receive mail job	This job terminates when it receives a specific email.
4	Receive MQ message job	This job terminates when it receives a specific message from TP1/Message Queue or MQSeries.
5	Receive MSMQ message job	This job terminates when it receives a specific message from MSMQ.
6	Monitoring log files job	This job is linked with the log file trapping of JP1/Base, and terminates when specific information is written to the specified log file.
7	Monitoring event log job	This job is linked with the log file trapping of JP1/Base, and terminates when specific information is written to the Windows event log file.
8	Interval control job	This job terminates when the specified time period elapses.

Note

JP1/AJS3 must be linked with the appropriate program to use a Receive mail job. For details, see the *Job Management Partner 1/Automatic Job Management System 3 Linkage Guide*.

Operation of event jobs is independent of JP1 user permissions and the authority level defined for the job (by owner, JP1 resource group, or by the user executing the job). In Windows, because event job operation is governed by the account rights to the JP1/AJS3 service, JP1/AJS3 service rights must be set in advance.

For details on defining a jobnet that uses an event job, see 2.4.4 *Executing an event-driven process (example of defining a jobnet that uses an event job)* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

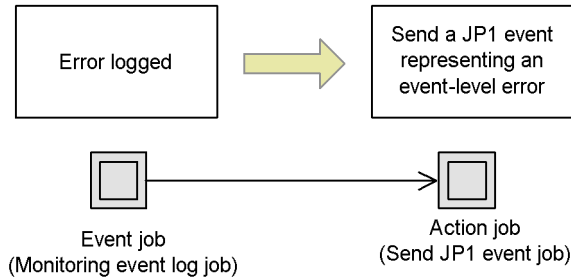
The event information received by an event job can be defined as a variable (macro variable) inherited by the succeeding job. For details on passing event information, see 2.4.4(6) *Passing information received by an event job* in the *Job Management Partner*

(e) Action job

An *action job* executes a specific process. You can combine an action job with an event job to execute a process (action) when an event occurs. Typical actions might be to send a JP1 event, an email message, or a status notification.

The following figure shows an example of a jobnet that uses an action job.

Figure 3-7: Example of a jobnet that uses an action job



There are seven types of action jobs:

- Send JP1 event job
- Send mail job
- Send MQ message job
- Send MSMQ message job
- OpenView Status Report job
- Local power control job
- Remote power control job

The following table outlines each of these action jobs.

Table 3-3: Types of action jobs

No.	Action job name	Description
1	Send JP1 event job	Sends a JP1 event to the event service of JP1/Base.
2	Send mail job	Sends an email message.
3	Send MQ message job	Sends a message to TP1/Message Queue or MQSeries.

No.	Action job name	Description
4	Send MSMQ message job	Sends a message to MSMQ.
5	OpenView Status Report job	Sends a status to HP NNM.
6	Local power control job	Powers on/off or restarts the host on which the local power control job was executed, working in conjunction with JP1/Power Monitor.
7	Remote power control job	Powers on a remote host or shuts down a system on the network, working in conjunction with JP1/Power Monitor.

Note

JP1/AJS3 must be linked with the appropriate program to use a Send mail job, OpenView Status Report job, Local power control job, or Remote power control job. For details on program linkage, see the *Job Management Partner 1/Automatic Job Management System 3 Linkage Guide*.

For details about defining a jobnet that uses an action job, see 2.4.5 *Sending a JP1 event at completion of the preceding job or when an event occurs (example of defining a jobnet that uses a Send JP1 event job)* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

(f) Custom job

A custom job defines a process that is executed in conjunction with JP1/AJS3 by a program other than JP1/AJS3. By using the custom job registration facility in JP1/AJS3 - View, you can define the non-JP1/AJS3 process as a job in JP1/AJS3, represented as a custom job icon.

Several types of standard custom jobs are provided as standard in JP1/AJS3, to enable linkage with the Windows version of the JP1 series.

Standard custom jobs are listed below.

Table 3-4: Standard custom jobs

No.	Custom job name	Works with	Custom job functions
1	JP1FTP	JP1/FTP	Transfers files in linkage with JP1/FTP.

No.	Custom job name	Works with	Custom job functions
2	JP1AMR3	JP1/AJS3 for Enterprise Applications	Works in conjunction with JP1/AJS3 for Enterprise Applications to automatically execute the background jobs of an R/3 (R) system.
3	JP1AMR3BW	JP1/AJS3 for Enterprise Applications	Works in conjunction with JP1/AJS3 for Enterprise Applications to execute the Info Package of a BW system.
4	JP1AMOAP	JP1/AJS2 for Oracle E-Business Suite	Works in conjunction with JP1/AJS2 for Oracle E-Business Suite to automatically execute the requests of an Oracle E-Business Suite system.

In addition to these standard custom jobs, you can create a *custom PC job* or *custom Unix job* by redefining the icon of a PC job or Unix job.

Setup is required to use custom jobs in JP1/AJS3. For details, see 3. *Adding Custom Jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Linkage Guide*.

(2) Jobnets

A *jobnet* is a collection of ordered jobs. The top-level jobnet is called the *root jobnet* and the jobnets defined below the root jobnet are known as a *nested jobnet*.

In JP1/AJS3, automated applications are executed by root jobnet. Thus, in the root jobnet, you must define the information required to schedule the jobnets for execution in JP1/AJS3. For details about schedule information, see 3.3 *Defining a schedule*.

When schedule information has been defined for a root jobnet, all lower units are scheduled according to that information. However, you can set schedule information separately for a specific nested jobnet. For example, if some processes require a different execution schedule from the other processes defined in a jobnet, you can group those processes into a nested jobnet and define a separate schedule for the nested jobnet. For details on scheduling a nested jobnet, see 3.5.3 *Defining a different schedule for some jobs in a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

For a root jobnet, you can also set the following scheduling-related information:

- Concurrent executions
- Number of logs to keep

- Priority[#]
- Timeout period
- Schedule option
- Jobnet monitoring
- Execution sequence control

#

Can also be set for a nested jobnet.

The number of logs to keep, priority, timeout period, and jobnet monitoring are explained below. For details about concurrent execution and the schedule option, see *3.3.3 Concurrent execution and schedule option* in this manual. For details about execution order control, see *2.2.4 Using jobnet connectors to control the order of root jobnet execution* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

■ Number of logs to keep

When a jobnet is defined and automated (registered for execution) in JP1/AJS3, it is managed as an object with a specific execution schedule, based on its schedule information. This object is called a jobnet *generation*. For details about generations, see *4.2 Managing jobnet generations*.

Number of logs to keep is a setting that specifies how many generations of execution results to save. The setting applies only to a root jobnet. Once you set the number of logs to keep, you can view the execution results for the set number of generations (root jobnet executions) in the Daily Schedule window or Monthly Schedule window. Alternatively, you can view past execution results by running the `ajsshows` command. For the command syntax, see `ajsshows` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

As the number of logs to keep, you can set any number from 1 to 99. However, the extended feature for this setting supports a maximum of 999 generations, depending on the environment settings for the Manager's scheduler service. For details about setting the Manager's scheduler service environment, see the `MAXSAVEGEN` environment variable in *2.2 Setting up the scheduler service environment* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 2*.

Past execution results are automatically deleted when they exceed the number of generations set in the **number of logs to keep**. For example, if you set 1, you will only be able to check the results of the latest run, even if the jobnet executes two or more times a day. Therefore, for a jobnet that is executed several times a day depending on the start conditions, you need to set a sufficient number of

generations. For details on the number of logs to keep, see *4.2.3 Jobnet generation management*.

■ Priority

You can set a process priority for jobs executed in a jobnet. The priority set for the root jobnet is handled as the default for all lower jobs and nested jobnets. Alternatively, you can set a separate priority for a specific standard job or for a specific nested jobnet.

The following table shows the correspondences between priority settings in JP1/AJS3 and their meaning in Windows and UNIX.

Table 3-5: Priority settings and their meaning in Windows and UNIX

Priority	Execution priority in Windows ^{#1}	Execution priority in UNIX ^{#2}
1	Lower than interactive processing	JP1/AJS3 service's nice value +20
2		JP1/AJS3 service's nice value +10
3	Same as interactive processing	JP1/AJS3 service's nice value
4	Higher than interactive processing	JP1/AJS3 service's nice value -10
5		JP1/AJS3 service's nice value -20

#1

There are three levels of priority in Windows. The job processes are started according to one of the following three priority classes.

- With priority setting **1** or **2**, the processes are executed when the system is idle.
(The Windows setting `IDLE_PRIORITY_CLASS` applies.)
- With priority setting **3**, the processes are executed as normal processes.
(The Windows setting `NORMAL_PRIORITY_CLASS` applies.)
- With priority setting **4** or **5**, the processes are executed before any process threads that were assigned priority **1**, **2**, or **3**.
(The Windows setting `HIGH_PRIORITY_CLASS` applies.)

#2

In UNIX, JP1/AJS3 service's nice value used when `ja js_spm` command is executed is used as the default (base) nice value. When the JP1/AJS3 service's nice value has not been set, 20 is assumed as the nice value.

For example, when the priority setting is 1 and the nice value is 20, the priority value is calculated as shown below.

$$39 \approx 20 \text{ (default)} + 20 \text{ (increment)}$$

If the resulting value is not within the possible nice value range (0-39), the maximum value is 39 and the minimum value is 0.

If **Queueless Agent** is selected as the execution service, you can set only 39, 30, 20, 10, or 0 as a nice value, according to the job execution priority; 0 corresponds to the highest execution priority. When you change the nice value, you must specify the job execution priority corresponding to the nice value.

For the `jajs_spmc` command syntax, see `jajs_spmc` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

■ Timeout period

Once registered, a root jobnet is placed in *Waiting for start time* status until execution begins. The timeout period specifies how long after the base time on the scheduled start day to cancel the root jobnet's *Waiting for start time* status if it fails to start at the scheduled start time because the scheduler service is inactive or a hold setting is in effect. When the *Waiting for start time* status is cancelled, the jobnet is placed in *Skipped so not executed* status. The following timeout settings are supported:

- **Use system settings**
- **1 day**
- **2 days**
- **Unlimited**

If you set a timeout period of 1 day or 2 days, the *Waiting to execute* or *Being held* status will be canceled when 1 or 2 days have elapsed from the base time on the scheduled start day. The jobnet will then be placed in *Skipped so not executed* status.

When the root jobnet is defined in a 48-hour time format in the schedule definition, and its scheduled start time is the next day, the timeout period will be taken as 2 days even is specified as 1 day.

The default is **Use system settings**. This means that the value set in the environment variable `EXECDEFER` (by default, `oneday`) will apply.

■ Jobnet monitoring

By setting the time required for execution for a jobnet, you can monitor for end delays based on how much time has passed since the jobnet started execution. For

example, if you set 30 minutes as the time required for execution, an end delay will be detected if the jobnet has not terminated 30 minutes after it started. Set the time required for execution in the range 1 to 2,879 (minutes).

Cautionary note

Jobnet monitoring is not supported if you are using version 06-71 or earlier of JP1/AJS2 - View or JP1/AJS2 - Manager.

(3) Jobnet connector

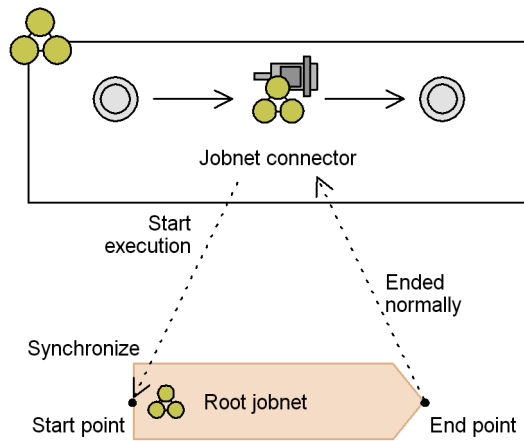
A jobnet connector is a unit that controls the execution sequence of root jobnets. It has the following two functions:

- Waits for the preceding root jobnet to end
 The jobnet connector waits for the root jobnet to terminate normally. If it terminates abnormally, the root jobnet enters *Running + Abend* status and waits without ending.
- Synchronizes execution of the associated root jobnet with its own startup process (optional function)

When the jobnet connector starts, it activates the associated root jobnet.

The following figure illustrates how a jobnet connector can be used.

Figure 3-8: Functions of a jobnet connector



A jobnet connector can be defined as a unit in a jobnet. It is used to control the execution sequence of a root jobnet that is defined directly under a root jobnet or under a planning group.

For details about jobnet connectors, see 2.2.4 *Using jobnet connectors to control the order of root jobnet execution* in the *Job Management Partner 1/Automatic Job*

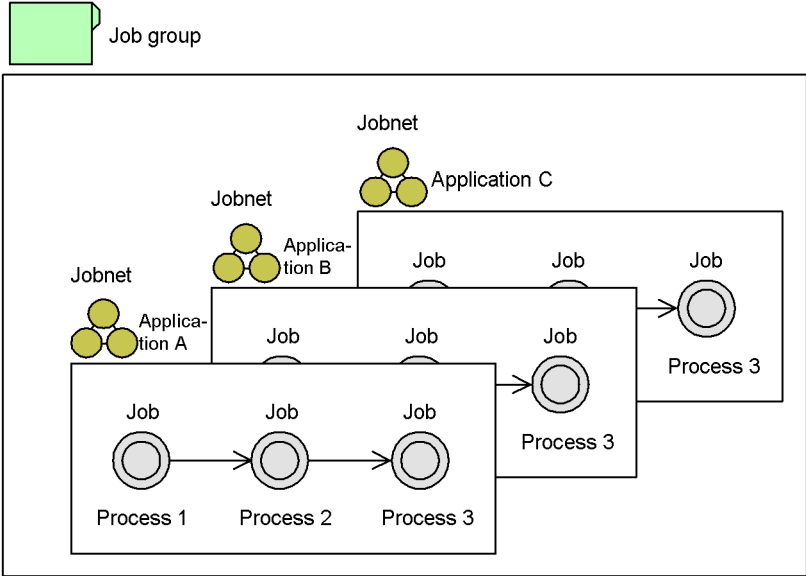
Management System 3 System Design (Work Tasks) Guide.

(4) Job group

A job group is a unit for sorting and grouping jobnets so that they can be managed systematically. A job group can be created under another job group.

The following figure illustrates the job group concept.

Figure 3-9: Job group



In a job group, you can define calendar information for JP1/AJS3 operation (open/closed days, base day/time, and so on). For details on calendar information, see 3.2 Defining a calendar for JP1/AJS3 operation.

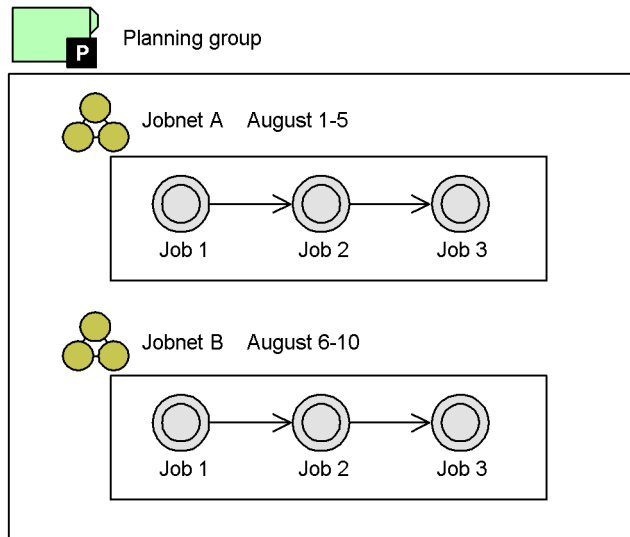
Note that job groups are jobnet management units and cannot be executed in their own right.

(5) Planning group

In JP1/AJS3, you can automatically switch execution from one jobnet (root jobnet) to another by specifying an execution period for each jobnet. A planning group is a unit that provides for this type of system operation.

The following figure shows an example of using a planning group.

Figure 3-10: Example of using a planning group



Suppose that you want to execute jobnet A from August 1 to August 5, and jobnet B from August 6 to August 10. First, you would create a planning group, and then create jobnets A and B directly under it. Execute the two jobnets, specifying the execution periods as August 1-5 for jobnet A, and August 6-10 for jobnet B. Execution will automatically switch from one jobnet to the other without interruption.

For details on planned jobnet switching using a planning group, see *10.1 Using a planning group to change the plans for root jobnets*.

You can create a planning group immediately below the scheduler service (AJSROOT) or under a job group. Only root jobnets and remote jobnets can be created in a planning group.

You can also define JP1/AJS3 calendar information for a planning group. For details on calendar information, see *3.2 Defining a calendar for JP1/AJS3 operation*.

3.1.2 Building a jobnet

The jobnet definition flow for automating applications in JP1/AJS3 is as follows:

1. Create a job group as required.
2. Create a jobnet.
3. Create the jobs.
4. Sequence the created jobs as required.
5. Set a start condition for the jobnet as required.

For details on setting a start condition, see *3.4 Defining a start condition*.

In JP1/AJS3, operations such as scheduling, execution, and management are performed in application units, that is, by root jobnet. For this reason, the jobnets will be easier to manage as a whole if you have some criteria in mind when grouping processes into jobnets.

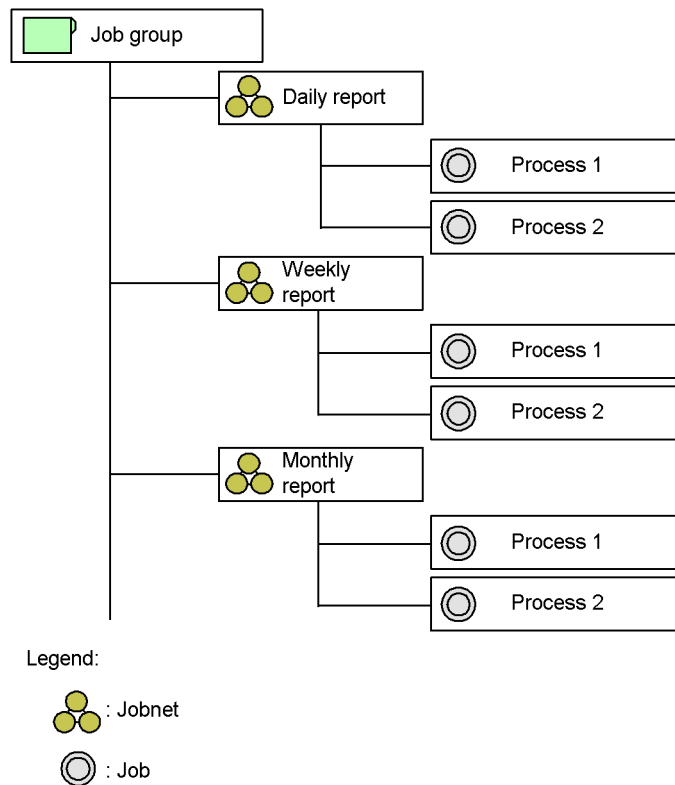
Below, we provide some examples of jobnet creation to illustrate how to group processes into a jobnet.

■ Example of management by application category

This example builds a jobnet of related applications. An application-based jobnet allows jobs to be easily added, changed, or deleted whenever required.

The following figure shows an example of management by application category.

Figure 3-11: Example of management by application category

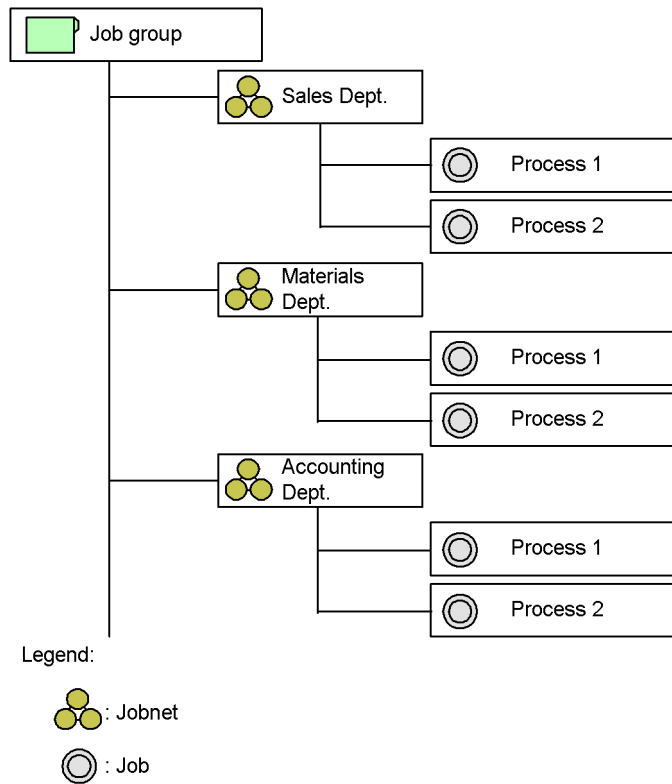


■ Example of management by department

This example builds a jobnet categorized by department. A departmental jobnet facilitates overall jobnet management and prevents operators in other departments from accessing the wrong applications.

The following figure shows an example of management by department.

Figure 3-12: Example of management by department

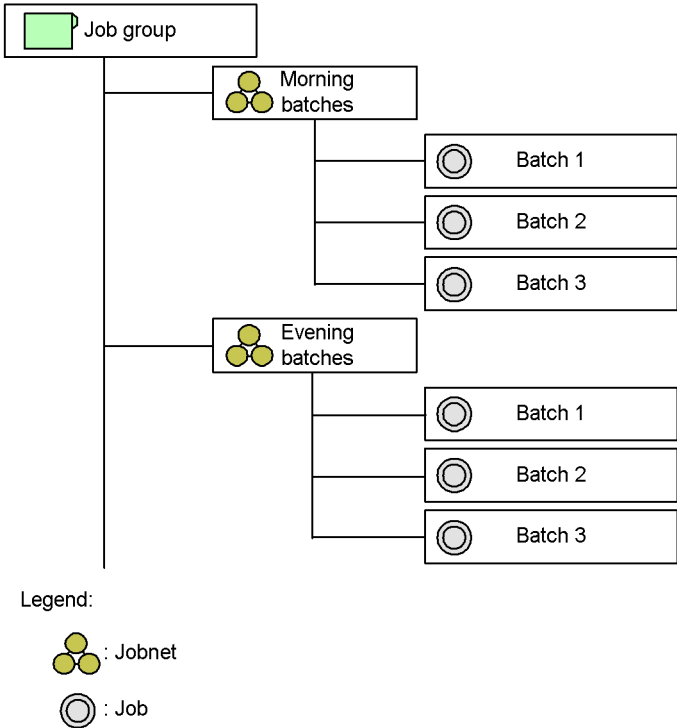


■ Example of management by processing category

This example builds a jobnet by processing unit. With a processing-based jobnet, there is no need for complex scheduling or structuring into hierarchical levels.

The following figure shows an example of management by processing category.

Figure 3-13: Example of management by processing category



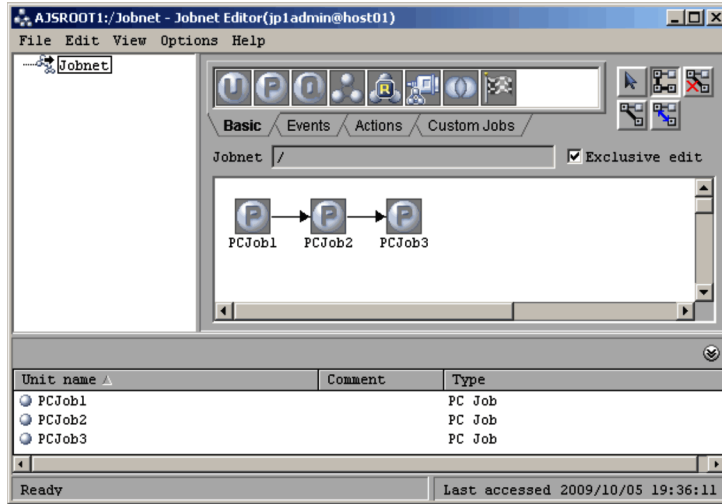
If you are using commands to operate on jobnets, it would be best to use lower-case alphanumeric characters when setting names to allow for the use of regular expressions.

3.1.3 Creating a job flow

A *job flow* is easy to create in the Jobnet Editor window of JP1/AJS3 - View.

In the Jobnet Editor window, you can easily create a job flow simply by linking jobs with arrows called *relations*. For details on JP1/AJS3 - View operations, see 5. *Defining Jobnets in the Job Management Partner 1/Automatic Job Management System 3 Operator's Guide.*

Figure 3-14: Example of creating a job flow in the Jobnet Editor window



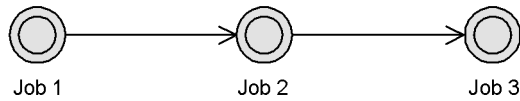
A variety of job flows can be created, depending on how the jobs are sequenced and inter-related. Some examples of job flow creation are provided below.

■ **Single processing path**

Job 1, Job 2, and Job 3 are executed sequentially in a single processing path. The job flow is created as follows:

Figure 3-15: Job flow in a single path

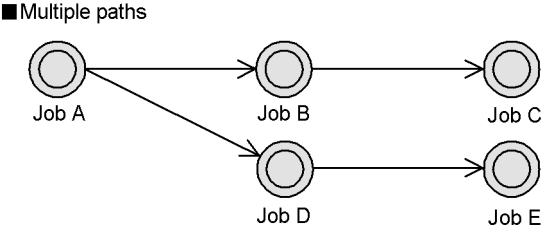
■ One path only



■ **Multiple processing paths**

The following figure shows an example of creating a job flow in multiple processing paths.

Figure 3-16: Job flow in multiple paths



When Job A is executed, the processing branches into two paths: Job A -- Job B -- Job C, and Job A -- Job D -- Job E.

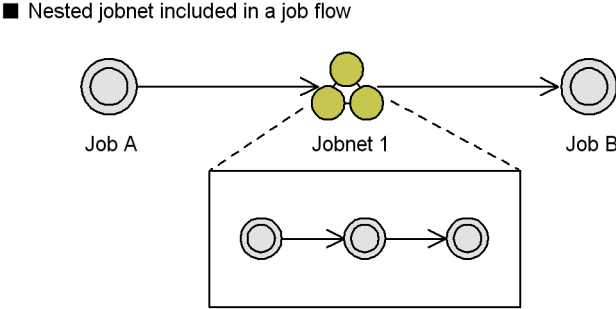
■ **Nested jobnet**

You can use nested jobnets in a job flow. Examples of using nested jobnets are shown below.

Nesting a jobnet

The following figure shows an example of nesting a jobnet within the job flow.

Figure 3-17: Example of using nesting a jobnet



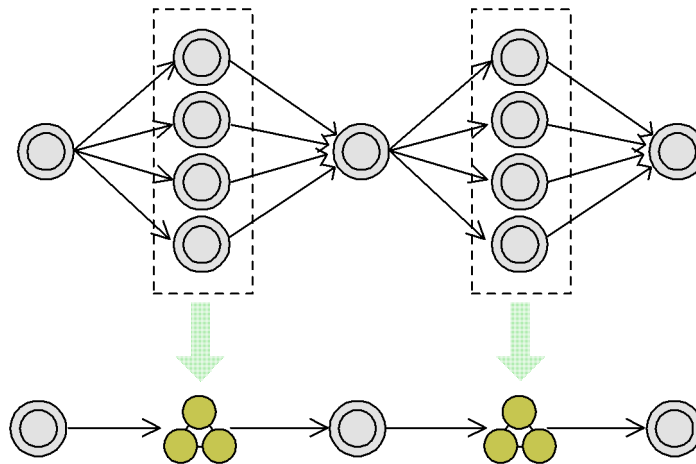
When Job A completes execution, the jobs defined in Jobnet 1 are executed. When processing of Jobnet 1 is finished, Job B is executed.

Grouping multiple jobs

The following figure shows an example of grouping multiple jobs into nested jobnets.

Figure 3-18: Example of grouping multiple jobs into nested jobnets

■ Collections of jobs grouped into two nested jobnets

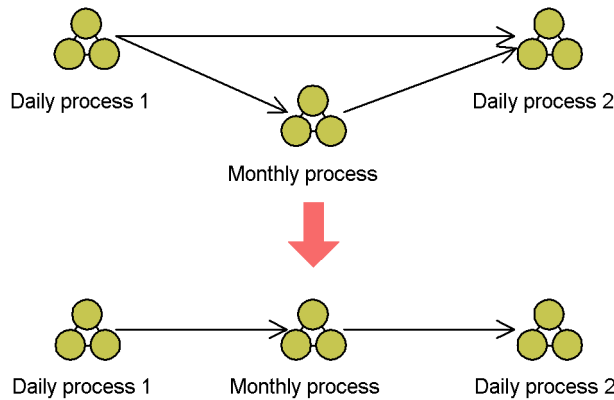


Joining two paths

Using a nested jobnet, you can join two processing paths into a single path.

The following figure shows an example of joining the paths *Daily process 1 -- Daily process 2*, and *Daily process 1 -- Monthly process -- Daily process 2*, into one flow.

Figure 3-19: Joining jobnets into one flow

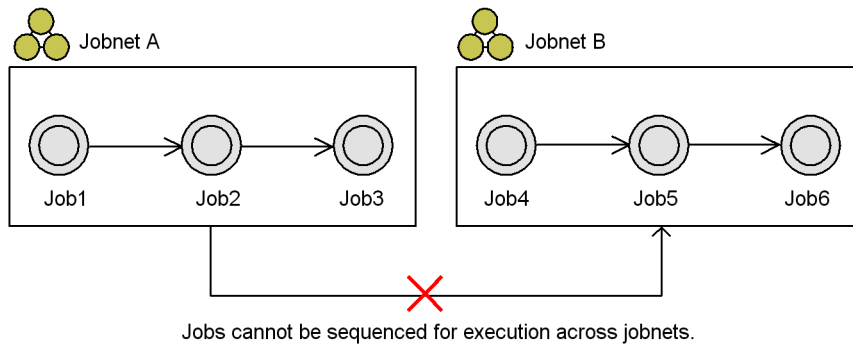


Because the two daily processes are executed every day and the monthly process is executed only once a month, JP1/AJS3 will skip any jobnet not scheduled for a particular day. This means that the two paths can be joined into one path.

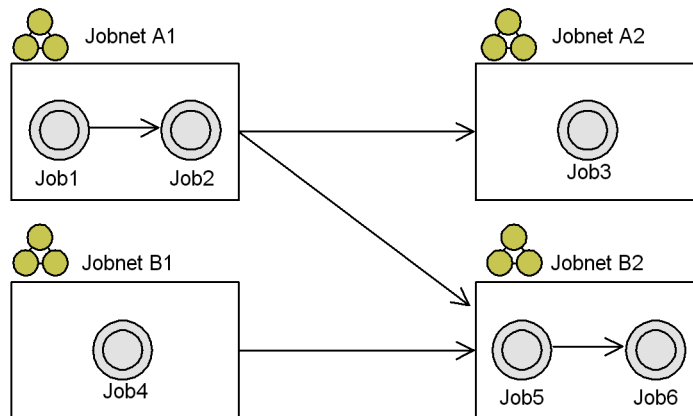
■ **Sequencing jobs in different jobnets**

In principle, JP1/AJS3 does not support job sequencing across jobnets. If you wish to sequence jobs from different jobnets, you must either split the jobnets or integrate the jobs from the two jobnets into a single jobnet, in either of the ways shown below:

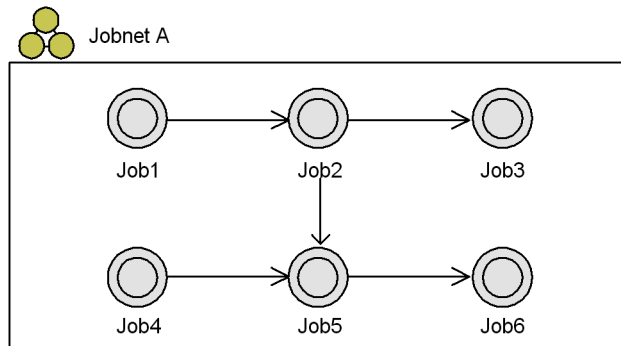
Figure 3-20: Ways of sequencing jobs in different jobnets



■ Split the jobnets



■ Integrate the jobnets



3.2 Defining a calendar for JP1/AJS3 operation

You can create an operational calendar that defines JP1/AJS3 open days and closed days, just like the Sundays and holidays in an ordinary calendar. You can also set the date on which each month begins (monthly start day), and the base time for calculating daily schedules, according to the work system and processing schedules in the workplace.

To create a calendar for JP1/AJS3 operation, you must define the following:

- Open days and closed days
- Base day and base time

You can define these items separately for each scheduler service (AJSROOT), job group, and planning group. Calendar settings for a scheduler service apply to all units defined under that scheduler service. Calendar settings for a job group or planning group apply to the jobnets belonging to that group. Therefore, to use a single operational calendar for an entire scheduler service, set calendar information for that service. To use application-specific calendars, set calendar information separately for each job group or planning group.

For the settings to define an application-specific calendar, see *3.5.7 Defining a different calendar for each application* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

3.2.1 Defining open days and closed days

Defining open days and closed days in JP1/AJS3 allows greater flexibility. For example, you can schedule jobnets on open days only, or move a scheduled execution time to an open day if it falls on a closed day.

The closed day and open day settings are reflected in the Monthly Schedule window, used for checking jobnet schedules and monitoring jobnet status.

3.2.2 Setting the base day and base time

You can set a monthly start day and daily start time to coincide with work systems and processing run times in your company. The date on which each month begins is known as the *base day*. The time from which daily schedules are calculated is known as the *base time*.

(1) Setting the base day

When no base day is set, the JP1/AJS3 system calculates each monthly period from the 1st day of the month. However, you can set a preferred monthly start day, according to operational and company requirements.

The system determines the base day from the following settings.

■ **Base day**

Set the base day in any of the following three ways:

• **Day**

Specify a particular day of the month as the monthly start day.

• **Week**

Specify a particular day of the week, and week of the month, as the monthly start day.

• **None**

Do not specify a base day. In this case, the first of the month is set as the monthly start day.

■ **Treat as**

Specify whether the month starting on the specified base day is to be treated as this month or next month.

Specify how to treat the month starting on the base day as either of the following:

• Treat as this month.

• Treat as next month.

For example, if the 26th is the base day and you want the period August 26 to September 25 to be treated as August, select **This month**. But if you want August 26 to September 25 to be treated as September, select **Next month**.

In this way, if you set **Base day** to 16 and then select **This month**, the period from August 16 to September 15 will be treated as August. This is useful when you are considering scheduling of a jobnet for an application that has a particular cut-off date.

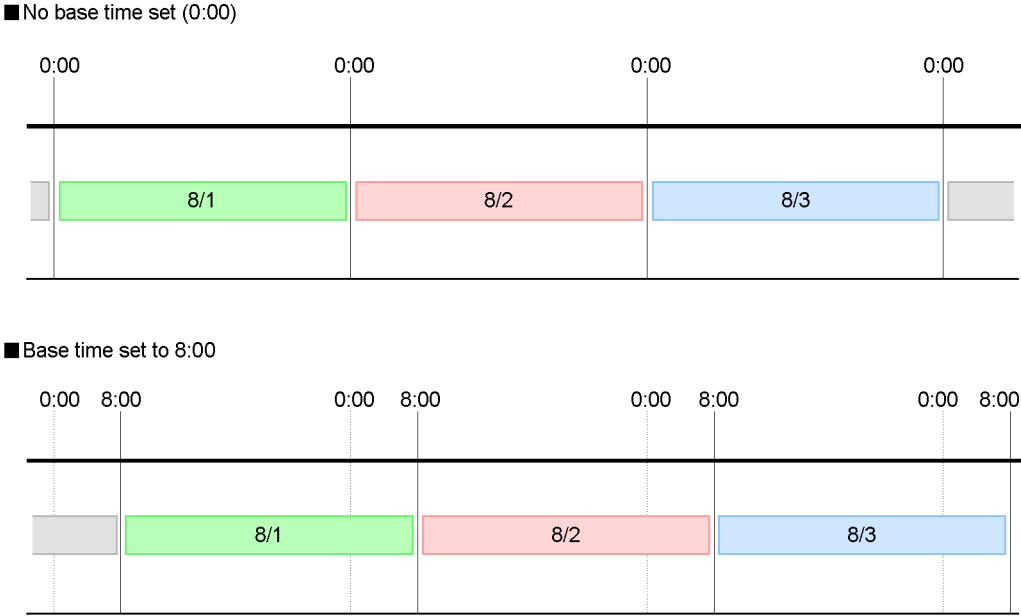
When you set the base day, the month starting from that base day is shown in the Monthly Schedule window, used for checking jobnet schedules and monitoring jobnet status.

(2) Setting the base time

When no base time is set, the JP1/AJS3 system calculates each day from 0:00. However, you can set a preferred time at which to start each day, according to the jobnets' mode of execution or other requirements.

The following figure illustrates how dates are calculated from the base time.

Figure 3-21: Base time handling



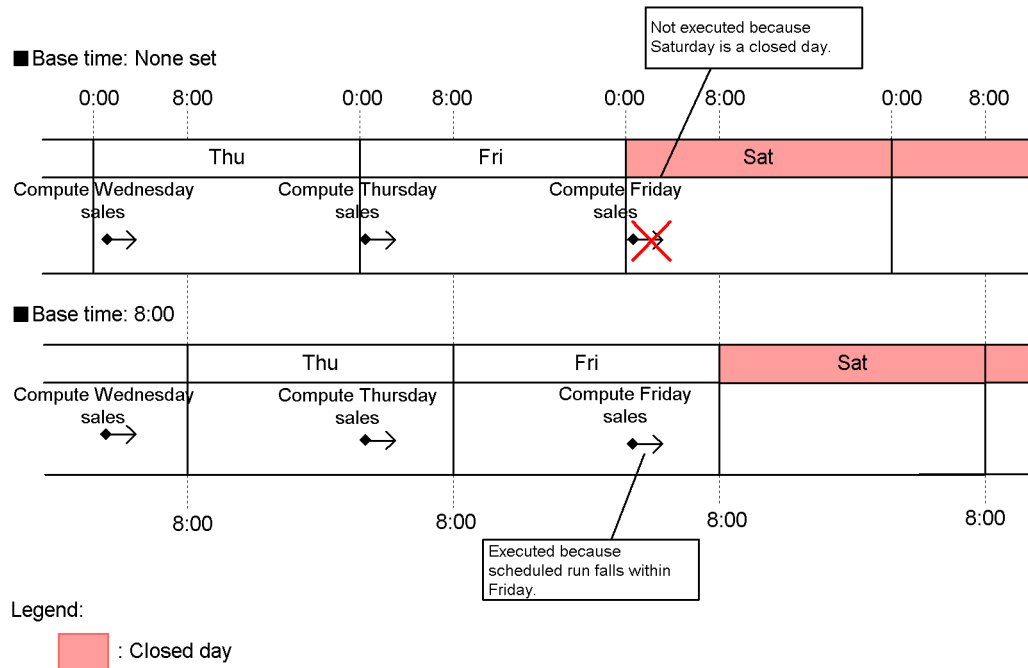
In this example, the base time is set to 8:00. This means that each day starts at 8:00, and lasts until 7:59 the following day.

When you set the base time, a 24-hour period from that base time is shown in the Daily Schedule window, used for checking jobnet schedules and monitoring jobnet status.

Setting a specific base time is useful when, from an operational point of view, you want to treat a jobnet as executed on the current day, although it will actually be executed on the following day.

An example of setting a base time is shown below.

Figure 3-22: Example of setting a base time



In this example, the sales data for each day from Monday to Friday is computed at 1:00 on the following day. Saturdays are closed days. Thus, although Friday's sales data should be computed at 1:00 on Saturday, the calculations are not performed because Saturday is a closed day. However, with the base time set to 8:00, each day ends a little later, so the computing run actually performed on Saturday can be handled as Friday's schedule.

Cautionary note

In the example in *Figure 3-22*, by setting the scheduled start time as 25:00 on the current day, the sales calculation can be performed at 1:00 on Saturday. However, determining the time settings will be complicated if the root jobnet has a 24-hour schedule and you are creating an application flow that extends into the next day or starts at 24:00 or later. For these sorts of applications, we recommend that you use a 48-hour schedule without setting a base time.

For details about using a 48-hour schedule, see *3.3.1(2) 48-hour schedule*. For a description of time formats and base times for root jobnets, see *3.3.1 Time formats and schedule rules for root jobnets*.

3.3 Defining a schedule

In JP1/AJS3, execution schedules are calculated from calendar settings and defined schedule information. This section describes how schedules are determined in JP1/AJS3, and the settings and functions used for schedule calculation.

An execution schedule derived from schedule information becomes a JP1/AJS3 operational schedule when you register it for execution as described in the next chapter.

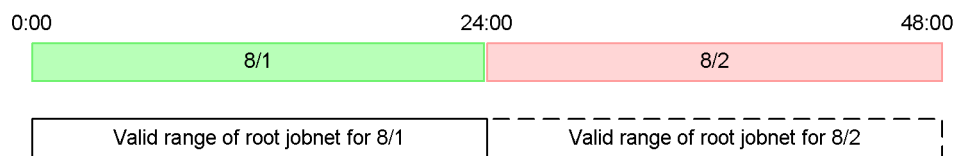
3.3.1 Time formats and schedule rules for root jobnets

A root jobnet can be run on a 24-hour schedule or 48-hour schedule.

(1) 24-hour schedule

The following figure shows the valid range of a root jobnet that uses a 24-hour schedule.

Figure 3-23: Root jobnet using a 24-hour schedule (base time 0:00)



When the root jobnet is based on a 24-hour time format, a jobnet that is set to start at 24:00 or later (between 24:00 and 47:59) will be treated as scheduled for the following day. For example, suppose that the scheduled start time is 25:00 on August 1. Because this setting exceeds the valid range of the root jobnet, the jobnet is treated as starting at 1:00 on August 2. The jobnet will also be shown as scheduled for August 2 when you check it in the Monthly Schedule window.

Cautionary note

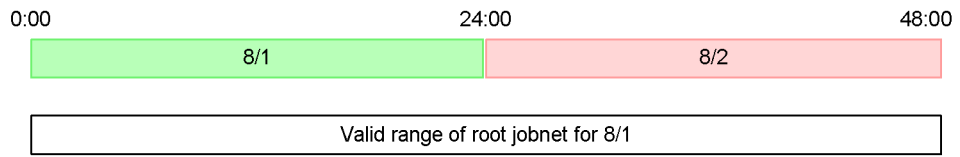
Although a jobnet due to start at 24:00 or later will be treated as scheduled for the next day when the root jobnet is based on a 24-hour time format, the calendar is executed according to the settings for the current day. For example, if the jobnet's scheduled start time is 25:00 on August 1, it will be treated as starting at 1:00 on August 2, but will be shown on the calendar as set for August 1. If August 1 is an open day, the jobnet will be executed on August 2 even if it is a closed day. If August 1 is a closed day, the jobnet will not be executed on August 2 even if it is an open day.

(2) 48-hour schedule

The following figure shows the valid range of a root jobnet that uses a 48-hour

schedule.

Figure 3-24: Root jobnet using a 48-hour schedule (base time 0:00)



When the root jobnet is based on a 48-hour time format, the period from 0:00 to 47:59 (with base time 0:00) is handled as one day. For example, a jobnet that is set to start at 25:00 on August 1 will be handled as scheduled for August 1. The jobnet will also be shown as scheduled for August 1 when you check it in the Monthly Schedule window.

Supplementary note

We recommend that you use the 48-hour schedule for jobnets that extend over two days, or are executed at or after 24:00. Time management in a 48-hour schedule whose base time is not 0:00 is complex, and the time to be specified might be different from the JP1/AJS3 operation time. Accordingly, we recommend that you do not change the base time from 0:00 when you use a 48-hour schedule. For details, see the explanation about setting an execution start time in 3.3.2(1) *Defining a schedule rule*, and 3.5.1(1) *Defining an application that extends over two days using a 48-hour schedule* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

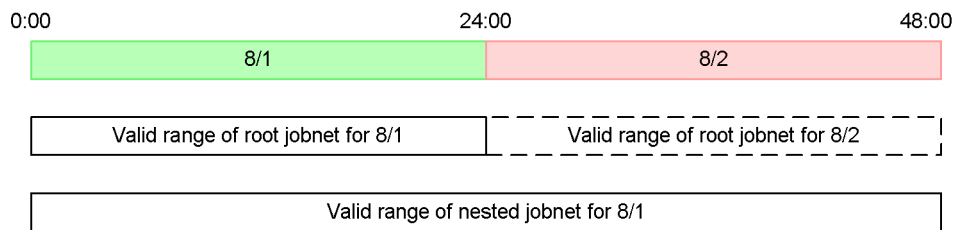
(3) Time format of nested jobnets

Nested jobnets run on a 48-hour schedule regardless of whether the root jobnet is based on a 24-hour or 48-hour schedule.

(a) Root jobnet based on a 24-hour schedule

The following figure shows the valid range of a nested jobnet when the root jobnet uses a 24-hour schedule.

Figure 3-25: Valid range of a nested jobnet based on a 24-hour schedule (base time 0:00)



Although the root jobnet has a valid range of 24 hours, the nested jobnet has a valid

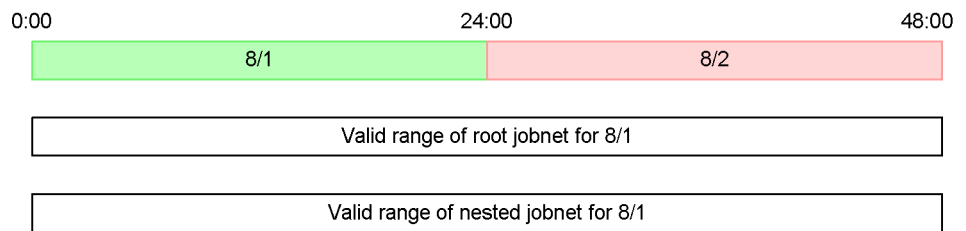
range of 48 hours. Therefore, if the start time for both jobnets is set to 24:00 or later (between 24:00 and 47:59), their execution dates will differ. For example, if both jobnets are set to start at 25:00 on August 1, the root jobnet will be scheduled for August 2, whereas the nested jobnet will be handled as executing on August 1.

Furthermore, because the two execution dates are different, the nested jobnet will not appear in the Daily Schedule window or Monthly Schedule window when you check its execution schedule.

(b) Root jobnet based on a 48-hour schedule

The following figure shows the valid range of a nested jobnet when the root jobnet uses a 48-hour schedule.

Figure 3-26: Valid range of a nested jobnet based on a 48-hour schedule (base time 0:00)



When the root jobnet uses a 48-hour schedule, the nested jobnet and root jobnet will have the same valid range. Therefore, even if the start time for both jobnets is set to 24:00 or later (between 24:00 and 47:59), their execution dates will be the same.

In this case, the nested jobnet schedule will appear in both the Daily Schedule and Monthly Schedule windows.

From the above discussion, we recommend that you use the 48-hour time format for the root jobnet when setting an execution schedule that starts at or after 24:00.

3.3.2 Defining a schedule

To calculate a jobnet execution schedule, you need to define schedule information including the start date/time, execution cycle, and how to reschedule an execution date that falls on a holiday.

(1) Defining a schedule rule

A *schedule rule* prescribes a particular requirement for calculating an execution schedule, such as the execution date/time and processing cycle (execution cycle) of a jobnet, or how to reschedule an execution date that falls on a holiday. The jobnet's execution schedule is determined from such schedule rules.

You can set up to 144 schedule rules for a jobnet. When multiple schedules have been defined, they are applied in order of the time settings in the schedule rules. When the

same start time is defined in two or more schedule rules, the schedules are applied in order of the schedule rule numbers.

For details on defining multiple schedule rules, see *3.5.2 Setting multiple execution start times* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

By default, JP1/AJS3 calculates the execution schedule of a nested jobnet according to the schedule rule for the upper-level jobnet. However, you can set schedule rules specifically for a nested jobnet. To set a schedule rule for a nested jobnet, you assign a rule number to the schedule rule set for the upper-level jobnet and link the schedule rule for the nested jobnet to that schedule. For details on defining a schedule for a nested jobnet, see *3.5.3 Defining a different schedule for some jobs in a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

The details that you can set in a schedule rule are described next.

(a) Setting the execution start time

There are a number of ways of setting an execution start time, depending on various factors such as whether a base day and base time have been set, and whether the root jobnet is based on a 24-hour or 48-hour time format. To define a schedule that spans days or starts at or after 24:00, we recommend that you use a 48-hour schedule without setting a base time (i.e., base time is 0:00). For details, see *3.5.1(1) Defining an application that extends over two days using a 48-hour schedule* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

■ Setting the start date

JP1/AJS3 determines the date for executing a jobnet from the following three settings.

Type

Method of setting the start date. Set one of the following five options:

- **Registered day**

Start the jobnet on the date it was registered for execution. With this setting, you do not need to set the start date.

- **Absolute day**

Start the jobnet on the date set in the calendar (where each month begins on the 1st).

- **Relative day**

Calculate the jobnet start date from the base day set in the calendar information.

- **Open day**

Calculate the jobnet start date from the base day set in the calendar information, counting only open days.

- **Closed day**

Calculate the jobnet start date from the base day set in the calendar information, counting only closed days.

Start year and month

Calendar year and month for starting the jobnet.

Start day

Set one of the following three options:

- **Specify day**

Specify which day of the month to start the jobnet.

- **Specify end of month**

Specify how many days before the last day of the month to start the jobnet.

- **Specify day of the week**

Specify which week and which day of the week to start the jobnet.

The following tables shows how the **Start day** options relate to the selected **Type** setting.

Table 3-6: Start day options by Type setting

Type	Start day option	Explanation
Registered day	--	--
Absolute day	Specify day	Specify a calendar date in the form <i>[]th day</i> .
	Specify end of month	Specify the number of days before the end of the calendar month in the form <i>[] days before last day</i> .
	Specify day of the week	Specify the day of the week, counting from the start of the calendar month, in the form <i>ordinal number</i> and <i>[]th week and day of the week</i> .
Relative day	Specify day	Specify a date, counting from the date set as the base day, in the form <i>[]th day</i> .
	Specify end of month	Specify the number of days before the end of the month, where a month is the period calculated from the set base day, in the form <i>[] days before last day</i> .

Type	Start day option	Explanation
	Specify day of the week	Specify the day of the week, counting from the date set as the base day, in the form [] <i>th week and day of the week</i> .
Open day	Specify day	Specify an open day, counting only the open days from the date set as the base day, in the form <i>ordinal number</i> .
	Specify end of month	Specify the number of open days before the end of the month, where a month is the period calculated from the set base day, in the form [] <i>open days before last day</i> .
Closed day	Specify day	Specify a closed day, counting only the closed days from the date set as the base day, in the form <i>ordinal number</i> .
	Specify end of month	Specify the number of closed days before the end of the month, where a month is the period calculated from the set base day, in the form [] <i>closed days before last day</i> .

■ **Setting the start time**

You can set the execution start time from 0:00 to 47:59 (minutes).

The time can be set in either of two ways:

Absolute time

Absolute time is the real time by the system clock.

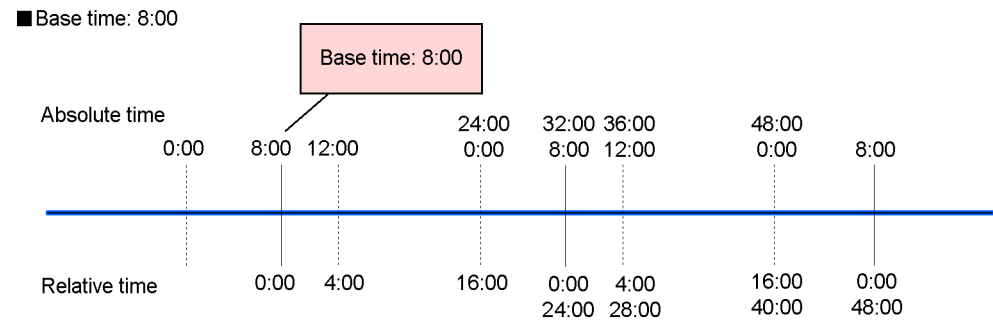
Using absolute time, you can set a schedule by clock time, regardless of whether a base time is set.

Relative time

Relative time is the time relative to a base time that is treated as 0:00.

The following figure illustrates the difference between relative time and absolute time.

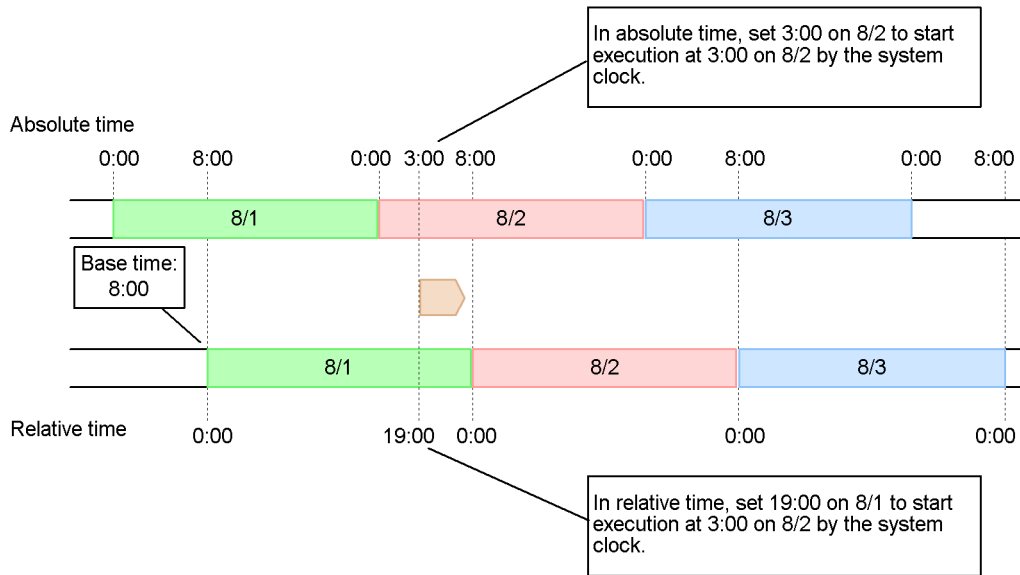
Figure 3-27: Absolute time and relative time



In this example, the base time is set to 8:00. If relative times are specified, the base time of 8:00 in absolute time will be 0:00 in relative time. Thus, 12:00 in absolute time will be 4:00 in relative time, and 24:00 in absolute time will be 16:00 in relative time.

The following figure shows how time settings differ in absolute time and relative time when a base time is set.

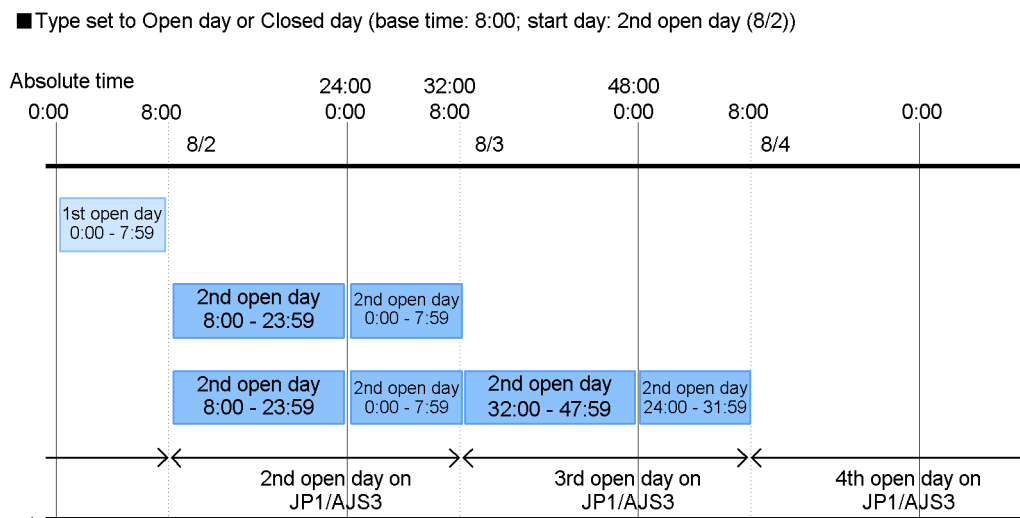
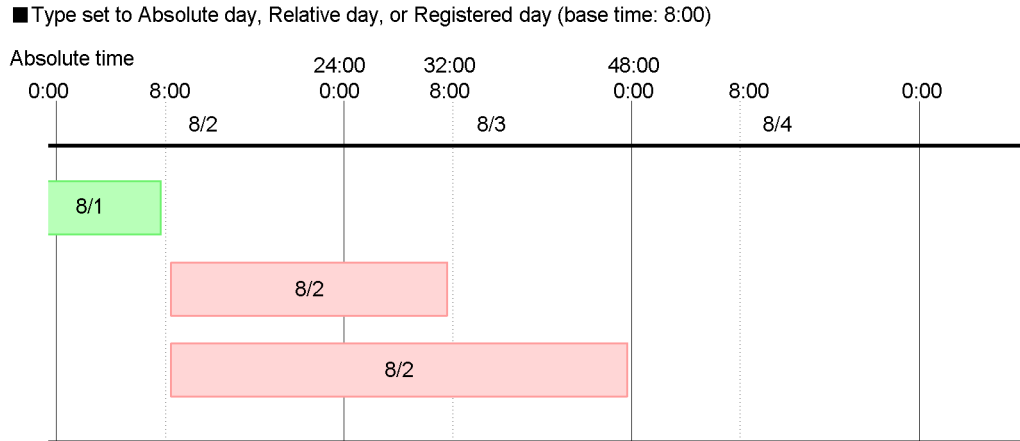
Figure 3-28: Difference between settings in absolute time and relative time



When you use relative time, the specified date matches the date in the JP1/AJS3 calendar. When you use absolute time, however, the date in JP1/AJS3 may differ from the specified date and time because the boundary between one day and the next shifts according to the base time.

The **Type** option can also affect the behavior of date and time settings. The following figure shows how dates and times are handled in JP1/AJS3 when the time is set in absolute time and a base time is set.

Figure 3-29: Handling of absolute time settings



■ When **Type** is set to **Absolute day**, **Relative day**, or **Registered day**

The period from 0:00 to the base time is within the previous day in absolute time.

The times in this example are handled as follows:

Using a 24-hour time format

The period from 0:00 to 7:59 is handled as within 8/1. The period from 8:00 to 23:59 on 8/2 and the period from 0:00 to 7:59 on 8/3 together constitute 8/2 in JP1/AJS3.

Using a 48-hour time format

The period from 8:00 to 47:59 on 8/2 is handled as within 8/2. The period from 0:00 to 7:59 on 8/4 when using a 48-hour time format will extend beyond the 48 hours in absolute time. Because it is not possible to specify a time on or after 48:00 under the absolute time specification, the period from 0:00 to 7:59 on 8/4 is handled as belonging not to 8/2 but to 8/3.

■ When Type is set to Open day or Closed day

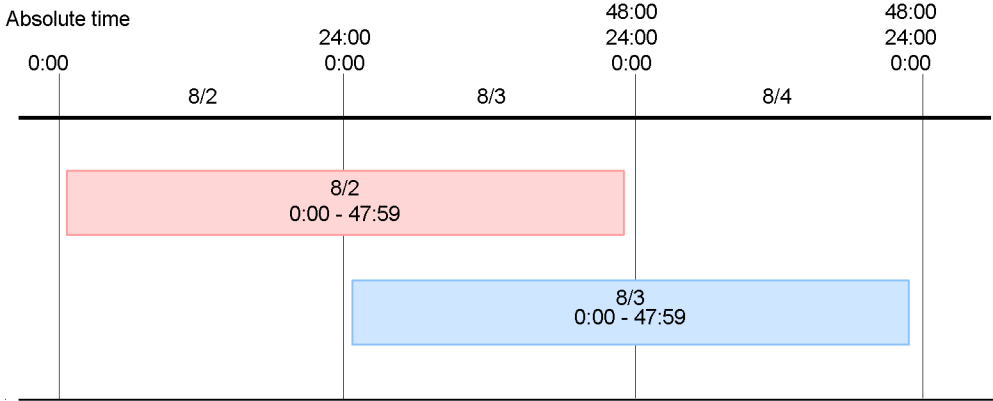
Because dates are determined relative to the base time, the date in JP1/AJS3 will be the same as the specified date, but a break occurs in the time continuum.

From these considerations, when you are creating a schedule that spans days or begins at or after 24:00, we recommend that you use a 48-hour schedule without setting a base time. For details about using a 48-hour schedule, see 3.3.1(2) 48-hour schedule.

When you use a 48-hour format without setting a base time (i.e., base time is 0:00), the date on JP1/AJS3 matches the specified date, and times are handled in a time series. This makes time specification clearer and simpler.

The following figure shows how time settings are handled with a 48-hour schedule.

Figure 3-30: Handling of time settings based on a 48-hour schedule (base time 0:00)



(b) Processing cycle

A processing cycle is a cycle for executing a jobnet. By setting a processing cycle, you can run a jobnet weekly, every three days, or at other regular intervals.

The execution dates in a processing cycle may differ according to the option set in Start day. An example of how the Start day option affects the execution date is shown below. The processing cycle here is monthly.

Figure 3-31: Differences between execution dates according to start day option

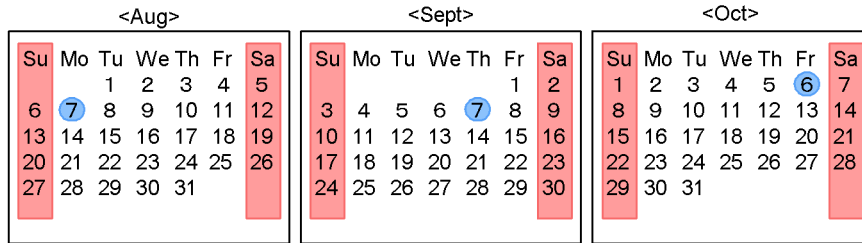
Execution example 1 Start day: August 1, 20XX



Execution example 2 Start day: August 20XX, 1st Thursday



Execution example 3 Start day: August 20XX, 5th open day (Saturday & Sunday are closed days)



Legend:

- : Jobnet execution date
- : closed day

In example 1, the execution date is always the 1st of every month, beginning from the start year and month (**Execute without shift** specified). In example 2, the execution date is the first Thursday of every month. In example 3, the execution date is the fifth open day every month.

(c) Substitute schedule of closed day job

The **Substitute schedule of closed day job** option specifies how to handle a scheduled jobnet when its execution date falls on a closed day.

Substitute schedule methods

You can set any of the following four options for rescheduling a jobnet when the execution date falls on a closed day:

- **Do not execute**

Do not reschedule the jobnet even if its scheduled execution date falls on a closed day. Do not execute the jobnet.

Do not select this option when a closed day is set as the start date. Otherwise, the execution schedule will not be calculated, even if the jobnet was registered for planned execution or fixed execution.

- **Execute on previous open day**

Execute the jobnet on the nearest open day before the closed day.

- **Execute on next open day**

Execute the jobnet on the nearest open day after the closed day.

- **Execute without shift**

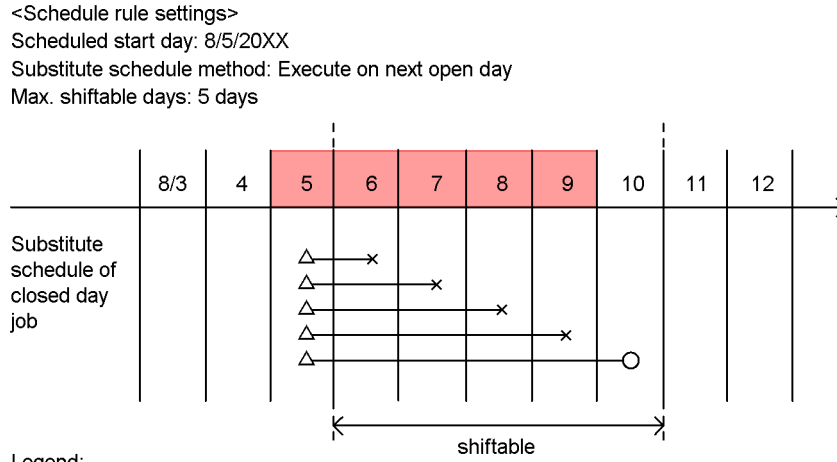
Do not reschedule the jobnet. If the JP1/AJS3 - Manager service is active, execute the jobnet on the closed day. If the JP1/AJS3 - Manager service is inactive, set the jobnet to *Skipped so not executed* status.

Maximum shiftable days

Max. shiftable days means the maximum number of days that a jobnet's execution date can be rescheduled (shifted) when it falls on a closed day. You can set **Max. shiftable days** when you set **Execute on previous open day** or **Execute on next open day** as the substitute schedule method.

In the following example, **Execute on next open day** is set as the substitute schedule method and **Max. shiftable days** is set to 5 days.

Figure 3-32: Example of a substitute schedule of closed day job



- Legend:
- : Closed day
 - Δ : Start day before shift
 - : Start day after shift
 - x : Closed day shift

Because **Execute on next open day** is set as the substitute schedule method, JP1/AJS3 looks for the nearest open day within the set maximum shiftable days (5 days).

In this example, however, the next four days (8/6, 8/7, 8/8, and 8/9) are also closed days. Therefore, the start day is rescheduled to the fifth open day.

If an open day is available within the set maximum shiftable days, the jobnet will be rescheduled even if its start date is outside the valid term of the schedule rules. For details about the valid term of schedule rules, see (2) *Valid term of schedule settings*.

■ **Substitute schedule of closed day job** and calendar information

- When a jobnet starts at or after 24:00 (between 24:00 and 47:59), the substitute schedule is calculated according to the JP1/AJS3 calendar date. For example, suppose that a jobnet is originally scheduled to run at 25:00 on August 6 (actually 1:00 on August 7 by the ordinary calendar). Because July 6 is a closed day, the jobnet is shifted to the next day. The rescheduled execution date will therefore be 25:00 on July 7 (actually 1:00 on July 8).
- When a jobnet has been registered for planned execution, and you alter the calendar information after JP1/AJS3 has set a substitute schedule for a closed day, the jobnet will be rescheduled according to the new calendar

information. For example, suppose that the jobnet has been shifted to the 6th because the 5th is a closed day. If you then alter the calendar information, making the 5th an open day, the jobnet will be rescheduled to run on the 5th. However, a jobnet registered for fixed execution will not be rescheduled if you alter the calendar information. For details about registering a jobnet for execution, see 4. *Executing an Application*.

■ **Substitute schedule of closed day job** and the processing cycle

When a processing cycle is specified, if there are no open days within the set **Max. shiftable days**, the system looks for a shift day within the set maximum period from the next scheduled execution date (processing cycle).

When multiple schedules are defined on the same day and at the same time, the jobnet actually runs once. Therefore, if the shift day falls on the date scheduled for the next processing cycle, the jobnet will be executed once only. Thus, if a daily processing cycle is set (**Execution every 1 Day**), the system cannot reschedule the jobnet. When using a processing cycle, make sure that you set **Max. shiftable days** to less than the number of days to the next run (the period set for the processing cycle).

■ **Substitute schedule of closed day job** and **Change plan** settings

The **Substitute schedule of closed day job** settings do not apply when you temporarily change the next scheduled start day and the new execution date falls on a closed day. The jobnet will be executed on that closed day, provided the JP1/AJS3 - Manager service is running. For details about temporarily changing a schedule, see 4.5.3 *Temporarily changing the execution start time of a jobnet*.

(d) **Start condition**

A *start condition* is a condition for executing a jobnet at a time other than the specified execution start time. As a start condition, you can define an event job that monitors the system for a specific event. When you set a start condition, you must also define in a schedule rule whether to use that condition. For details on start conditions, see 3.4 *Defining a start condition*.

(e) **Valid range of start condition**

When a start condition is set for a jobnet, you must specify in advance the valid range of that start condition. You can set the valid range in any of three ways:

- Specify how many times to run the jobnet when its start condition is satisfied.
- Specify the valid range in absolute time.
- Specify the valid range in relative time from the scheduled monitoring start time.

When you register the jobnet for planned execution or fixed execution, the JP1/AJS3 system ceases monitoring the start condition in accordance with the valid range definition that you set.

For details on the valid range of a start condition, see *3.4 Defining a start condition*.

(f) Monitor delay

The **Monitor delay** option determines whether a jobnet started and ended at the expected times, as set in its execution schedule.

Jobnets can be monitored for end delays based on the elapsed time since the jobnet started execution. Alternatively, start delays and end delays can be monitored using a time threshold set as a delay judgment criterion.

End delay monitoring based on the elapsed time since the jobnet started

Using this method, you set the time required for execution for the jobnet. An end delay is detected if the jobnet's execution time exceeds the set duration. For details about jobnet monitoring based on time required for execution, see the description of jobnet monitoring in *3.1.1(2) Jobnets*.

End delay and start delay monitoring based on delay judgment criteria

Using this method, you set a time threshold as the delay judgment criterion. A start delay or end delay is detected if the set time is exceeded. You can set the criterion in absolute time (in the range 0:00 to 47:59) or relative to the jobnet's start time (in the range 1 to 2,879 minutes). If the time set by either of these methods is exceeded, the delay is reported (an event is issued and the status change is shown in JP1/AJS3 - View). Note that a delay does not mean that execution of the jobnet is canceled.

You can also set delay monitoring for a nested jobnet, using any of the following four methods:

- Set by absolute time
- Set by relative time from the start time of the root jobnet
- Set by relative time from the start time of the upper-level jobnet (jobnet at one level higher)
- Set by relative time from the start time of the nested jobnet

To monitor delays in a jobnet whose execution schedule extends into the next day or starts at 24:00 or later, we recommend that you use a 48-hour schedule without setting a base time (that is, the base time is 0:00). For details, see *(a) Setting the execution start time*. See also *3.5.1(1) Defining an application that extends over two days using a 48-hour schedule* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

Cautionary notes

- Delay monitoring is not performed if the jobnet has already entered *Ended* status. This occurs, for example, when a preceding unit terminates abnormally and the succeeding jobnet is not executed and enters *Not*

executed + *Ended* status.

- Delay monitoring restarts if the jobnet is re-executed and once again enters *Waiting for start time* or *Waiting for prev. to end* status. However, if a start delay was detected at the first run, the rerun jobnet is not monitored for a start delay, even if it enters *Waiting for start time* or *Waiting for prev. to end* status. This is also true for an end delay.
- When monitoring for a start delay or end delay is set for a dependent jobnet, if the preceding judgment job exceeds the set monitoring time without terminating, a delay will be detected even if the dependent jobnet is not executed. You can avoid detection of an end delay in such cases by monitoring the jobnet based on its time required for execution.

(g) Schedule by days from start

Schedule by days from start means rescheduling a jobnet by specifying the number of open days to count forward or backward from the execution date already determined from the start date/time, processing cycle, substitute schedule for closed days, or other schedule settings.

Rescheduling methods

There are two methods of rescheduling a jobnet by counting days from the scheduled start time:

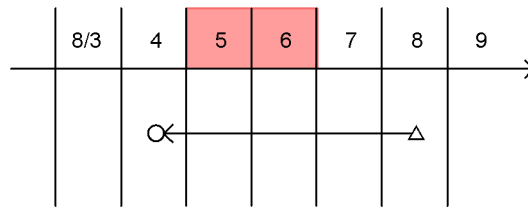
- Execute n open days before the scheduled start time
- Execute n open days after the scheduled start time

In n , set the number of days to count.

In the following example, a jobnet is rescheduled by setting **Execution: 2 open days before start time** in **Schedule by days from start**.

Figure 3-33: Example of rescheduling a jobnet

<Settings for schedule by days from start>
 Scheduled start day: 8/8/20XX
 Reschedule: 2 open days before start time



Legend:

- : Closed day
- Δ : Existing execution schedule
- : New execution schedule

The start time is counted back two open days from the scheduled start time on August 8. Therefore, the jobnet will now execute on August 4.

Max. shiftable days

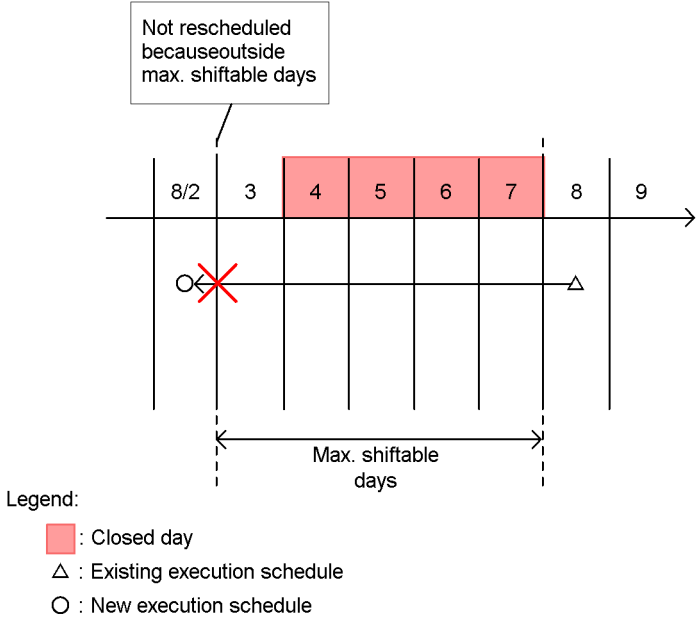
Max. shiftable days means the maximum number of days that you can count forward or backward from the scheduled start time.

Only open days are counted in **Schedule by days from start**, but both open and closed days are included in **Max. shiftable days**. Therefore, if there are no open days in the range set in **Max. shiftable days**, the system skips that execution schedule without rescheduling the jobnet.

In the following example, **Execution: 2 open days before start time** is set in **Schedule by days from start**, and **Max. shiftable days** is set to 5 days.

Figure 3-34: Example of max. shiftable days

<Settings for schedule by days from start>
Scheduled start day: 8/8/20XX
Reschedule: 2 open days before start time
Max. shiftable days: 5 days



Two open days before the start time falls outside the set maximum shiftable days (5 days). Therefore, the jobnet is not rescheduled and the run is skipped.

For details on defining a jobnet using the **Schedule by days from start** function, see 3.5.5 *Shifting the scheduled execution date forward or back based on a calculated schedule (Schedule by days from start)* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

(2) Valid term of schedule settings

You can set the period for which a schedule rule defined for a jobnet is applicable.

The valid term of a schedule rule (date on which it expires) can be set separately for each jobnet. When the specified date arrives, the jobnet automatically stops being executed.

(3) Schedule setting options

JP1/AJS3 provides the following two options for defining schedule information:

- Refer to a calendar of another job group

- **Exclusive schedule**

These options are described below.

(a) Refer to a calendar of another job group

When defining a jobnet, if you need to consider the operational information set for another job group, you can refer to that job group's calendar information as the operational calendar for the jobnet you are defining.

Note that when you schedule a jobnet based on the calendar information of another job group, the Monthly Schedule and Daily Schedule windows show the execution schedule using the calendar information of the job group to which that jobnet belongs.

For details on calendar information, see *3.2 Defining a calendar for JPI/AJS3 operation*.

Cautionary note

If any of the following operations is performed on a jobnet registered for planned execution, schedule calculation might fail, with the jobnet being placed in *Shutdown* status. Cancel planned registration before you execute any of these operations:

- Change the job group specified in the schedule definition as the job group whose calendar is to be referenced.
- Delete the job group whose calendar is to be referenced.
- Rename the job group whose calendar is to be referenced.

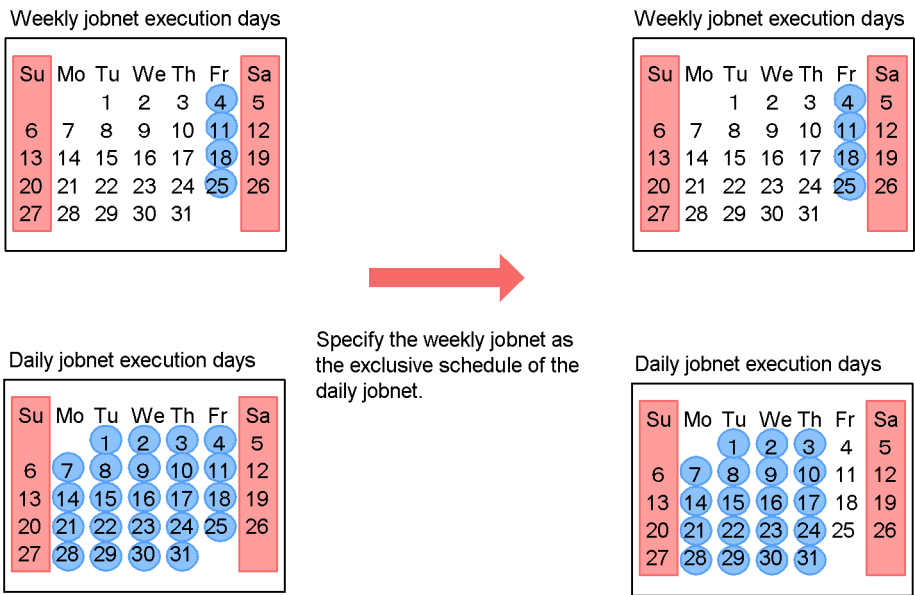
(b) Exclusive schedule

You can set an exclusive schedule so that the jobnet you are defining will not be executed if its execution schedule coincides with that of another jobnet that you do not want to run on the same day.

The following figure shows an example of using an exclusive schedule.

Figure 3-35: Example of using an exclusive schedule

The daily and weekly jobnets are both executed on a Friday. Because the weekly jobnet incorporates the daily jobnet process, you do not want to run the daily jobnet on the same day as the weekly jobnet.



The daily jobnet's Friday schedules, which coincided with execution of the weekly jobnet, disappear from the calendar.

Legend:
■ : Closed day
● : Jobnet execution day

In this example, the daily jobnet runs every day and the weekly jobnet runs on Fridays. When these two jobnets are executed, their execution dates will necessarily coincide on Fridays. The weekly jobnet incorporates the daily jobnet process, so you do not want to run the daily jobnet on any day scheduled for the weekly jobnet. Therefore, set the weekly jobnet as the exclusive schedule of the daily jobnet. When you set an exclusive schedule in this way, execution of the daily jobnet is canceled for any execution date that falls on the same execution date as the weekly jobnet.

The following rule applies to exclusive schedule settings.

Figure 3-36: Rule governing exclusive schedule settings

<Rule for setting an exclusive schedule>

Set jobnet B as an exclusive schedule of jobnet A.
 Set jobnet C as an exclusive schedule of jobnet B.
 ↓
 Jobnet C will be an exclusive schedule of jobnet A.

Jobnet A

Execution schedule: Daily
 Exclusive schedule: Jobnet B

Jobnet B

Execution schedule: 8/2 to 8/6
 Exclusive schedule: Jobnet C

Jobnet C

Execution schedule: Every Thursday
 Exclusive schedule: None

<Jobnet exclusive schedules>

	8/1 Tu	2 We	3 Th	4 Fr	5 Sa	6 Su	7 Mo	8 Tu
Jobnet A	●	--	--	--	--	--	●	●
Jobnet B	--	●	--	●	●	●	--	--
Jobnet C	--	--	●	--	--	--	--	--

Legend:

- : Scheduled
- : Not scheduled

Suppose that jobnet B is set as an exclusive schedule of jobnet A, and jobnet C is set as an exclusive schedule of jobnet B. In this case, jobnet C becomes an exclusive schedule of jobnet A, even though it was not explicitly set as an exclusive schedule of jobnet A.

You can specify a planning group as an exclusive schedule. For details, see *10.1(1)(b) Exclusive schedules for planning groups*.

Cautionary note

If any of the following operations is performed on a jobnet registered for planned execution, schedule calculation might fail, with the jobnet placed in *Shutdown* status. Cancel planned registration before you execute any of these operations:

- Change the jobnet specified as an exclusive schedule in the jobnet's schedule definition.
- Delete the jobnet specified as an exclusive schedule.
- Rename the jobnet specified as an exclusive schedule.

Supplementary note

- To determine an exclusive schedule, the system references the schedule definition information of the jobnet specified as the exclusive schedule. This

means that the schedule can be simulated if the specified jobnet has not been registered for execution.

- If the schedule of the jobnet specified as an exclusive schedule is temporarily changed by **Change plan**, the change is applied, but does not affect any schedule for which the exclusive schedule is specified.

3.3.3 Concurrent execution and schedule option

The **Concurrent exe.** and **Schedule option** settings determine how the system handles execution schedules when a jobnet with a set processing cycle fails to complete before the start time of the next run. You can enable or disable concurrent execution, and set a schedule option, separately for each root jobnet.

(1) Concurrent execution

Concurrent execution means running multiple instances of the same jobnet at the same time. You can enable or disable concurrent execution separately for each root jobnet. By enabling concurrent execution, you can start the next run of a jobnet at the scheduled time in the processing cycle, even if the previous run has not yet completed. Concurrent execution applies only when you select **Multi-schedule** as the **Schedule option**, as described below.

By enabling concurrent execution for a jobnet that has a start condition, you can run the jobnet every time the start condition is satisfied, even if earlier executions are still running. For details on start conditions, see *3.4 Defining a start condition*.

(2) Schedule option

The **Schedule option** sets the method for determining the next execution schedule when a jobnet with a set processing cycle fails to complete before the start time of the next run.

JP1/AJS3 provides two schedule options:

- **Schedule skip**
- **Multi-schedule**
- Schedule skip

The **Schedule skip** option skips execution of any jobnet whose previous run is still being processed at the start time of the next run, and instead schedules a jobnet whose previous run is already finished by the start time of the next run. A jobnet whose execution was skipped, because the previous run did not finish before the start time of the jobnet, is placed in the *Skipped so not exe.* status.

Note that skipping of a jobnet occurs only when the previous run was started by an execution schedule. If the previous run was started by a re-execution operation, the jobnet is executed after the re-executed run ends.

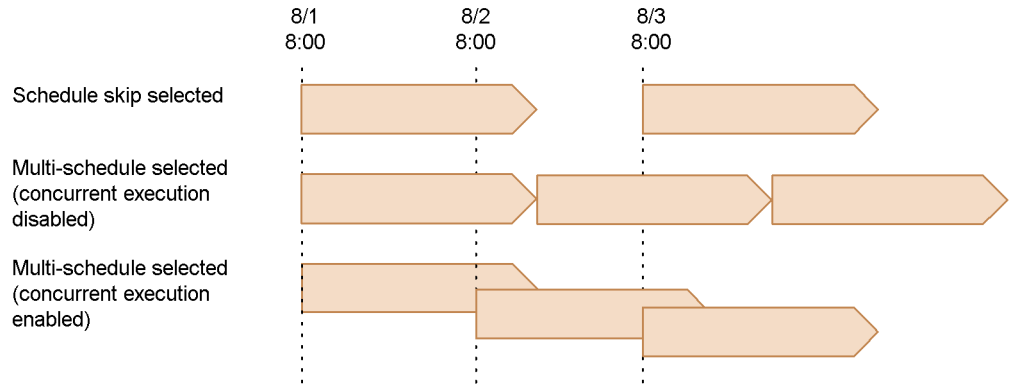
■ Multi-schedule

When **Multi-schedule** is selected, jobnet scheduling differs depending on whether concurrent execution is enabled or disabled. If you enable concurrent execution, the next run of a jobnet will be executed at the scheduled time without waiting for completion of the previous run. If you disable concurrent execution, the next run will not be executed until the previous run finishes processing.

The following figure shows how the system handles the next run of a jobnet when **Schedule skip** and **Multi-schedule** are selected.

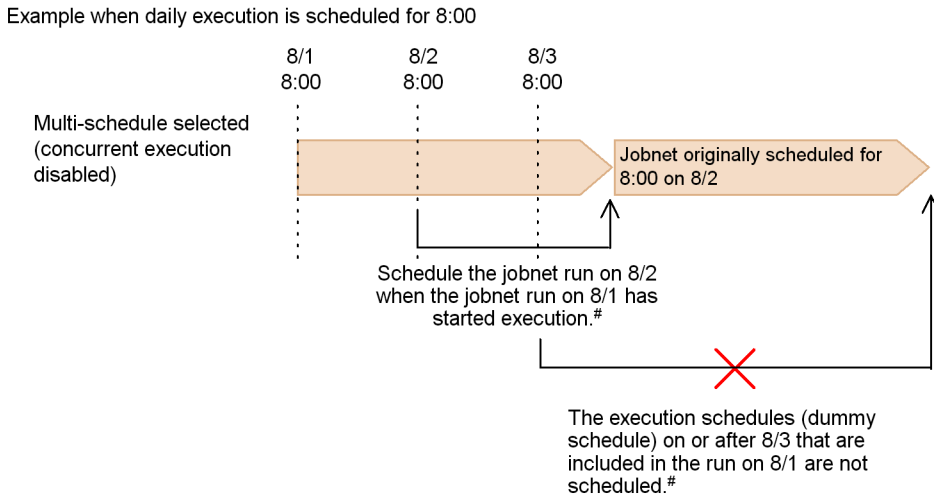
Figure 3-37: Method of execution with 'Schedule skip' and 'Multi-schedule' selected

<Jobnet executed daily at 8:00>



When a jobnet is registered for planned execution, the time of the next execution is finalized at the start time of the previous execution. The second and subsequent executions are treated as dummy runs and are not actually scheduled. Therefore, if concurrent execution is disabled and **Multi-schedule** is selected, a jobnet registered for planned execution will be handled as follows when two or more runs of the jobnet are due to start during execution of the previous run. For details about registration for planned execution and dummy schedules, see 4. *Executing an Application*.

Figure 3-38: Two scheduled jobnet runs held over



This figure shows an example of a jobnet timeout period that has been set to **2 days** or **Unlimited**. If the jobnet timeout period has been set to **1 day**, the schedule for August 2 is placed in *Skipped so not exe.* status on the base time. Therefore, the schedule for August 3 is executed after the schedule for August 1 is executed.

The system schedules the jobnet for 8/2 when the jobnet scheduled for 8:00 on 8/1 has started execution. Because **Multi-schedule** is set and concurrent execution is disabled, the 8/2 jobnet does not start until the 8/1 jobnet has completed. Similarly, the execution schedule (dummy run) set for 8:00 on 8/3 should be scheduled when the 8/2 jobnet starts execution, but because the 8/1 jobnet does not complete until after 8:00 on 8/3, this dummy run is effectively skipped.

Supplementary note

You can also set **Concurrent exec.** and **Schedule option** for a root jobnet in a planning group.

3.4 Defining a start condition

In addition to setting a start time, you can set a condition for initiating a jobnet. This is called a *start condition*.

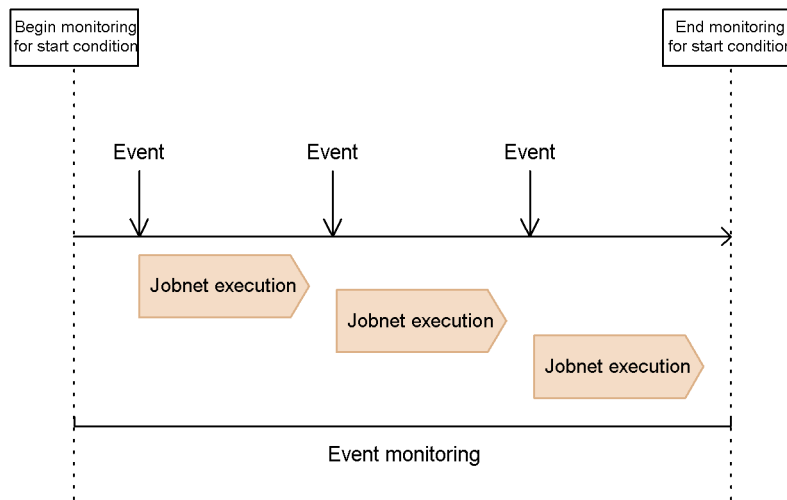
This section describes how to define start conditions and their purpose.

3.4.1 Purpose of a start condition

By setting a start condition, you can execute a jobnet when an event occurs, just like defining a preceding event job to trigger a jobnet.

The following figure shows how the system executes a jobnet that has a preset start condition.

Figure 3-39: Execution of a jobnet with a start condition



As a start condition, you can only define an event that can be monitored by an event job. For example, you can define a JP1 event, incoming email, file update, or time interval.

You can define a start condition only for the root jobnet. Only one start condition can be set. When you set a start condition, the system creates a jobnet called `.CONDITION`, which monitors for the set condition to be satisfied. In the `.CONDITION` jobnet, you define an event job for monitoring the event that will act as the condition. For how to set a start condition, see *5.4 Setting a start condition* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

A start condition can consist of multiple event jobs. When you define multiple event jobs for the start condition, whether the condition is satisfied depends on which of the

following you use:

- Execute the jobnet when all conditions are satisfied (AND condition)
- Execute the jobnet when one of the conditions is satisfied (OR condition)

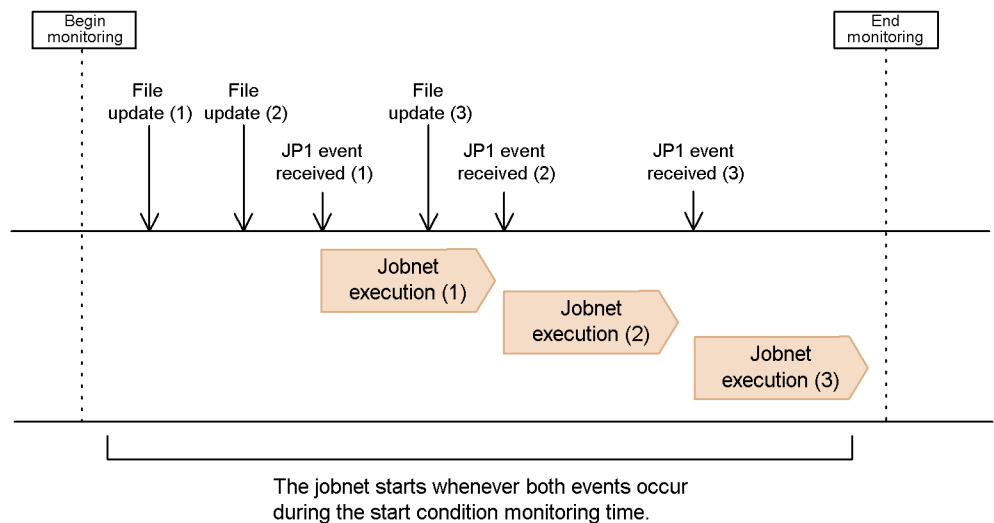
AND and OR conditions are explained next.

AND condition

An *AND condition* is satisfied when all of the event jobs defined in the condition have occurred.

The following example shows how a jobnet is executed when a start condition, defined as an AND condition, is made up of a Monitoring files job and a Receive JP1 event job.

Figure 3-40: Example of an AND start condition with multiple events



Cautionary note

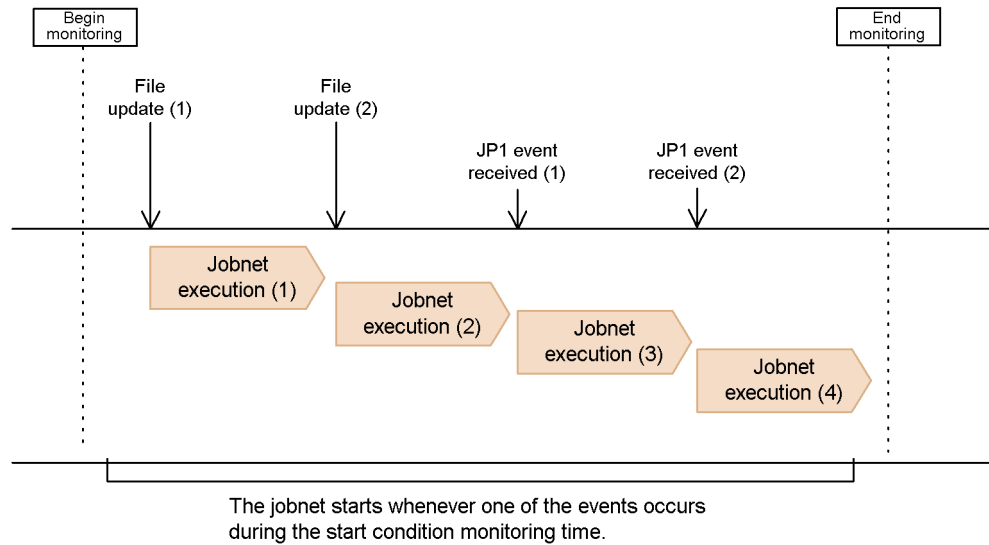
When you use AND to monitor multiple conditions, create definitions for which the same event does not occur in succession. (For example, in the above figure, file updates occur in succession.) If an event occurs in succession, multiple generations will be created, which will wait for the occurrence of the other events defined in the AND condition. This might affect system performance.

OR condition

An *OR condition* is satisfied when one of the event jobs defined in the condition occurs.

The following example shows how a jobnet is executed when a start condition, defined as an OR condition, is made up of a Monitoring files job and a Receive JP1 event job.

Figure 3-41: Example of an OR start condition with multiple events



When you use an OR condition, the condition is satisfied when either one of the defined events occurs.

When concurrent execution is enabled, multiple instances of the jobnet may run at the same time, executed each time the start condition is satisfied.

Supplementary notes

- The event information received by an event job defined in a start condition can be passed to the succeeding unit. By setting a macro variable in an event job, you can pass the received information to all the jobs under the root jobnet governed by the start condition. For details about passing information from an event job, see 2.4.4(6) *Passing information received by an event job in the Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.
- When one of the event jobs defined for a start condition detects the same event in succession, the order in which those events are detected may differ from their actual order of occurrence, due to the communication status or other factors. There may be situations, however, when the order in which the start condition is satisfied by each event needs to match the order in which the events actually occurred, because the event information is to be passed to the succeeding job, for example. In such cases, use the event order option.

However, when this option is used, the number of events that can be processed in a specific length of time is fewer than when this option is not used.

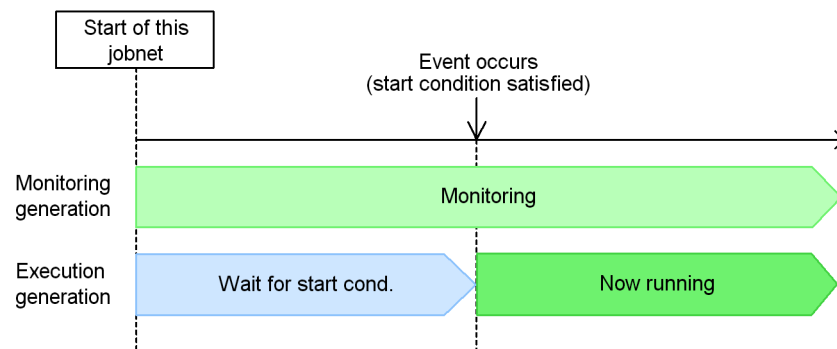
For details on setting the event order option, see 6.3.2 *Setting the event order option* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1* for a Windows host, or 14.3.2 *Setting the event order option* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1* for a UNIX host. For details on estimating the number of events that can be processed with the event order option, see 3.1.4 *Event monitoring performance* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

3.4.2 Monitoring generation and execution generation

When a jobnet with a start condition is executed, two types of generations are generated: a *monitoring generation*, which controls monitoring for the occurrence of an event, and an *execution generation*, which waits for the occurrence of the event. When the event being monitored occurs, the start condition is satisfied and the execution generation is executed.

The following figure shows the monitoring generation and the execution generation.

Figure 3-42: Monitoring generation and execution generation



When a jobnet with a start condition that has been registered for execution starts at the start time specified for a schedule rule, a monitoring generation in the *Now monitoring* status and one or more execution generations in the *Wait for start condition* status are generated.

If an event occurs and the start condition is satisfied, the execution generation in the *Wait for start condition* status is placed in the *Now running* status, and a new execution generation with the *Wait for start condition* status is generated.

Note that when monitoring terminates, the monitoring generation in the *Now*

monitoring status is placed in the *Monitor-end normal* status. If the condition has not been satisfied before the monitoring terminates, the generation is placed in the *Unmonitored + Ended* status. If an error occurs in a monitoring generation in the *Now monitoring* status, the execution generation is not executed even when the start condition is satisfied, but is instead placed in the *Ended abnormally* status.

You can check these statuses from the Daily Schedule window or the Monthly Schedule window.

3.4.3 Valid range of a start condition

The *valid range of a start condition* refers to the applicable range in which the occurrence of an event defined as a start condition will count as satisfying that condition.

How a valid range is set for a start condition depends on the method of jobnet execution registration.

Planned execution registration or fixed execution registration

Select whether to use a start condition when defining a schedule rule for a jobnet. If you use a start condition, also set a valid range. The settings take effect when you register the jobnet for planned or fixed execution.

Immediate execution registration

When you register the job for execution, select whether to use a start condition. If you use a start condition, also set a valid range.

For details about registering a jobnet for planned, fixed, or immediate execution, see *4. Executing an Application*.

You can specify the valid range of a start condition by setting an execution count and/or by setting a period (in absolute time or relative time).

Number of executions

Specify the number of times that the execution generation can be executed from the time that monitoring of the start condition begins.

Monitoring of the start condition continues from the time set in the jobnet's schedule definition until the number of executions initiated by the start condition reaches this value.

Period

- **Absolute time**

Specify in absolute time when to stop monitoring.

Monitoring of the start condition continues until the specified absolute time arrives.

- **Relative time**

Specify relative to the monitoring start time when to stop monitoring.

Monitoring of the start condition continues for the specified period (in minutes) from the time that monitoring of the start condition begins.

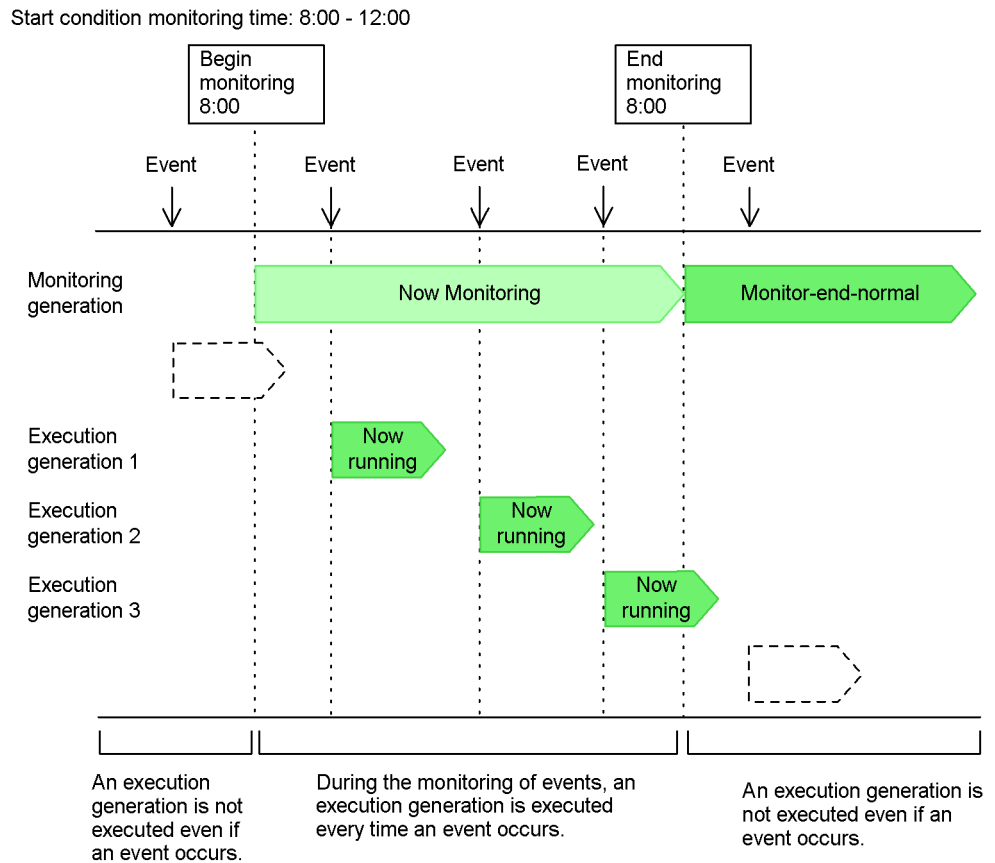
If the event defined in a start condition occurs outside the valid range set for the jobnet, the start condition is not satisfied and the execution generation is not executed. For a jobnet registered for planned execution or fixed execution, the next run is scheduled according to the schedule rule definition when monitoring of the start condition ends.

- **Valid range specified in absolute time**

When you specify a valid range in absolute time, events are monitored from the time that monitoring of the start condition begins until the specified absolute time. The execution generations are executed for the number of events that occur within this period.

The following figure shows an example of setting an absolute time.

Figure 3-43: Example of setting the valid range of a start condition (absolute time)



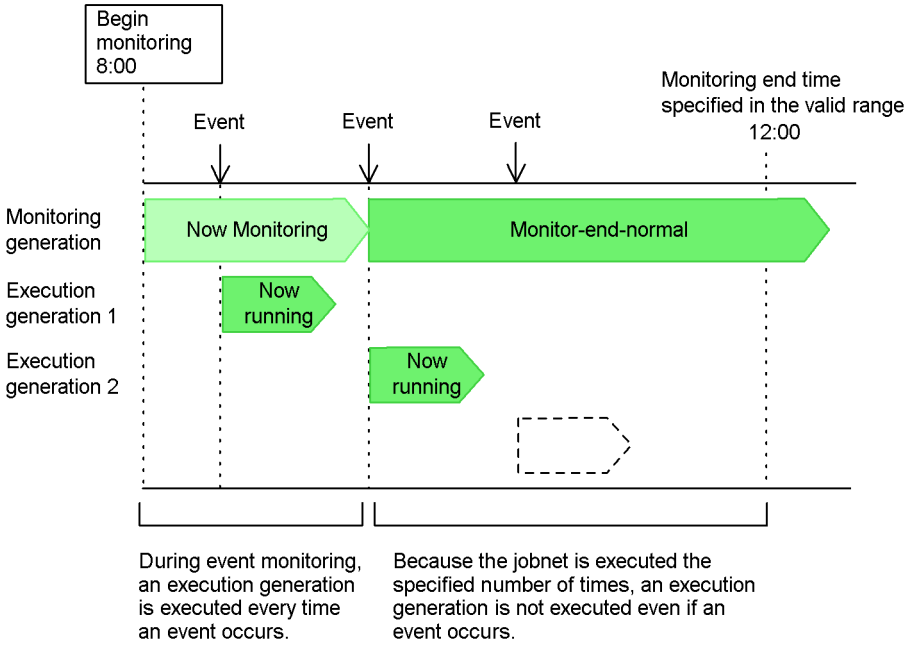
■ Valid range specified in absolute time and number of executions

When you specify both a period and an execution count as the valid range of a start condition, monitoring of the event ends when either the specified time arrives or the number of executions reaches the specified count.

The following figure shows an example of setting both an absolute time and number of executions.

Figure 3-44: Example of setting the valid range of a start condition (number of executions and period)

Start condition monitoring time: 8:00 - 12:00
 Number of executions: 2 times



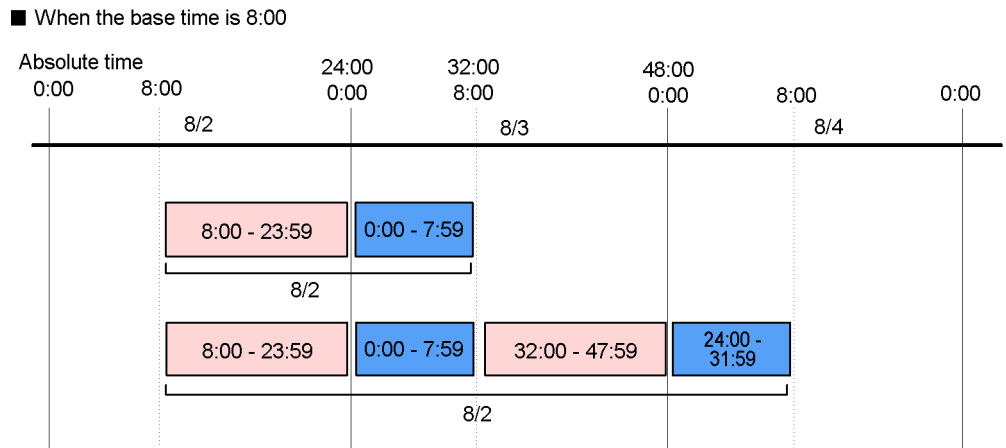
In the above example, when the number of jobnet executions reaches the specified count (2 times), subsequent occurrences of the event are regarded as invalid even though the monitoring end time (12 : 00) specified in the valid range has not yet arrived.

Cautionary notes

- When a base time (other than 00:00) is set for an upper-level job group, be aware of how the absolute times used to specify the valid range are handled in JP1/AJS3.

The following figure shows how times are handled when the valid range of the start condition is specified in absolute time for a jobnet with a set base time.

Figure 3-45: Handling of absolute time specified as the valid range of a jobnet with a set base time



Dates are calculated from the set base time, but times are handled as shown in the figure. For example, suppose that a jobnet is scheduled for execution on 8/2 using a 48-hour time format, and the valid range is set to 26:00 in absolute time. The jobnet's valid range in calendar time will actually be 2:00 on 8/4.

To avoid this problem, if you are running applications at or after 24:00, we recommend that you use a 48-hour schedule without setting a base time (i.e., base time is 0:00).

- For a jobnet with a start condition, the schedule options (schedule skip and multi-schedule) are disabled for monitoring generations and execution generations. To specify a processing cycle by using a jobnet schedule, you need to consider the concurrent execution of monitoring generations rather than schedule options. For details about the concurrent execution of monitoring generations, see 3.4.5 *Concurrent execution of monitoring generations*.
- A jobnet that has a start condition and a set processing cycle is scheduled accordingly when you register it for execution. However, if an unlimited number of executions and an unlimited period are set as the valid range of the start condition, two or more generations may be placed in *Wait for start condition* status. When the start condition is satisfied, all the waiting generations will be executed at the same time.

Supplementary notes

- If you set the valid range period to **Unlimited**, you should also set **Unlimited**

in **Time-out period** in the jobnet definition. Otherwise, if an unlimited start condition is satisfied after the time-out period, the execution generation may be placed in *Skipped so not executed* status.

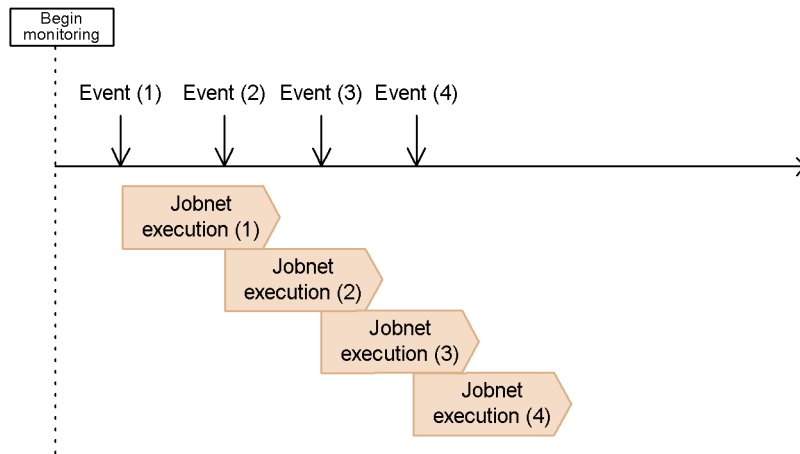
- Set the valid range period, in either absolute or relative time, so that it is later than the jobnet's execution start time. If you set an earlier time, the monitoring generation will be set to *Monitor terminated* status as soon as it starts execution.

3.4.4 Concurrent execution of execution generations

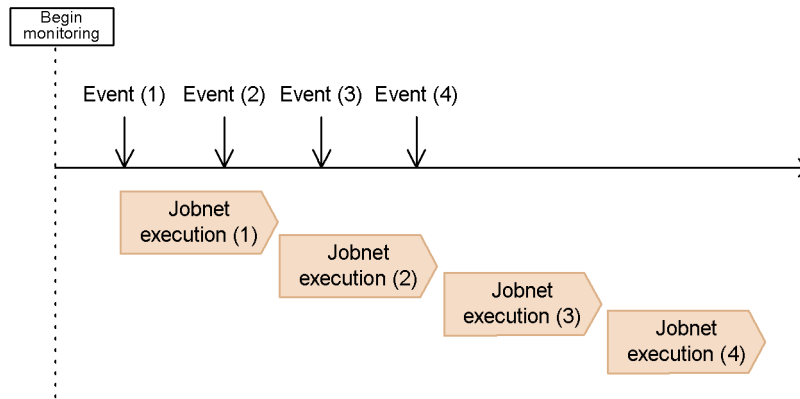
If a start condition is satisfied when a monitoring generation is in the *Now monitoring* status and an execution generation is in the *Now running* status, a new execution generation is generated. The behavior of the execution generation depends on whether concurrent execution is enabled or disabled. The following figure shows the behavior according to the concurrent execution setting.

Figure 3-46: Jobnet behavior according to the concurrent execution setting

■ Concurrent execution enabled



■ Concurrent execution disabled



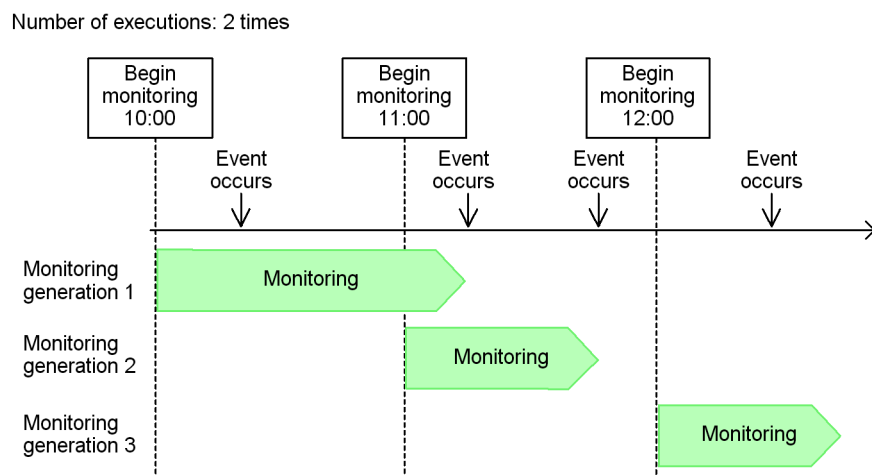
- When concurrent execution is enabled
An execution generation is executed every time the start condition is satisfied, even if the previous execution generation is still in progress.
- When concurrent execution is disabled
If the start condition is satisfied when no execution generation is running, the execution generation is executed. If the start condition is satisfied while the execution generation is running, the next execution generation waits until the current execution generation ends.

To enable the concurrent execution of execution generations, use the Define Details - [Jobnet] dialog box (for a root jobnet) or the Define Details - [Remote Jobnet] dialog box (for a root remote jobnet). For details, see 15.3.8 *Define Details - [Jobnet] dialog box (for a root jobnet)* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide* or 15.3.10 *Define Details - [Remote Jobnet] dialog box (for a root remote jobnet)* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

3.4.5 Concurrent execution of monitoring generations

When the start time of another jobnet is reached while the monitoring generation is in the *Now monitoring* status, a new monitoring generation is generated. By default, monitoring generations are executed concurrently as shown in the following figure.

Figure 3-47: Example of the concurrent execution of monitoring generations



In this example, suppose there are three event jobs, starting at 10:00, 11:00, and 12:00. Each of these event jobs is set to run until a start condition is satisfied twice in succession. If the second event does not occur between 10:00 and 11:00, monitoring generation 2 will start before monitoring generation 1 has ended.

You can select one the following three options to specify whether to start monitoring generations concurrently:

- **Do not start monitoring**

When a jobnet's start time arrives and another run starts while a monitoring generation with *Now monitoring* status is in progress, the monitoring generation for the second execution of the jobnet is placed in *Skipped so not executed* status and does not begin monitoring.

- **Wait for the end of monitoring**

When a jobnet's start time arrives and another run starts while a monitoring generation with *Now monitoring* status is in progress, the monitoring generation for the second execution of the jobnet is placed in *Wait for start time* status. When the preceding generation ends, the new monitoring generation moves from *Wait for start time* to *Now monitoring* status.

- **Start monitoring**

When a jobnet's start time arrives and another run starts while a monitoring generation with *Now monitoring* status is in progress, the monitoring generation for the second execution of the jobnet is placed in *Now monitoring* status and is executed concurrently with the preceding generation. By default, this setting is selected.

To enable the concurrent execution of monitoring generations, set **Concurrent exec. of monitoring gen.** in the Define Details - [Start Condition] dialog box. For details, see 15.4.5 *Define Details - [Start Condition] dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

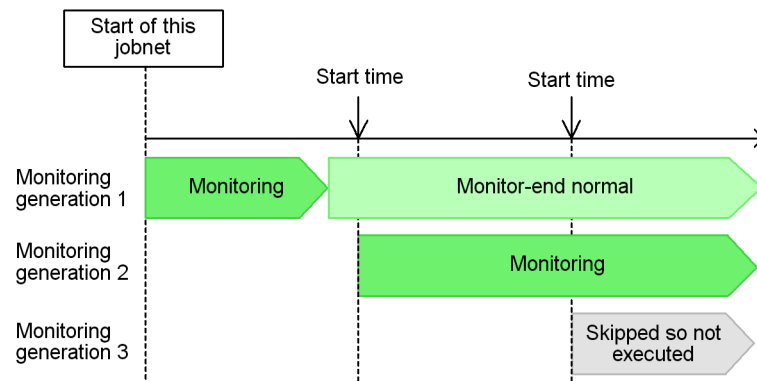
These behaviors are further described below.

When the hold attribute is set for a monitoring generation, it takes precedence and the concurrent execution setting is disabled.

(1) Behavior with the setting "Do not start monitoring"

The following figure shows the behavior of the monitoring generations when you select **Do not start monitoring**.

Figure 3-48: Behavior with "Do not start monitoring" defined in the jobnet's start condition



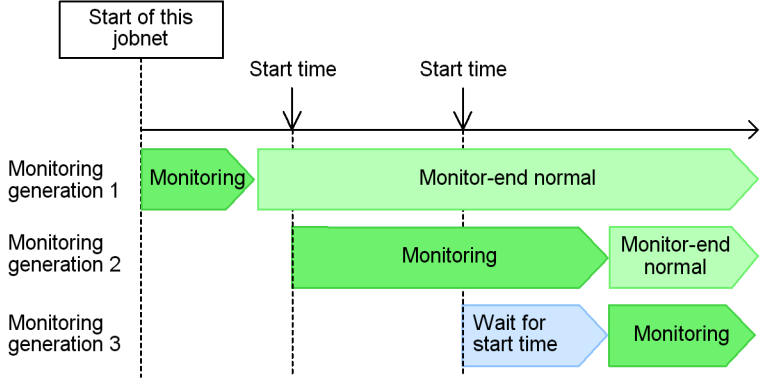
If the next start time arrives while monitoring generation 2 is in *Now monitoring* status, monitoring generation 3 enters *Skipped so not executed* status and does not start. We

recommend that you select this behavior for a jobnet that has a set processing cycle and a start condition with an unlimited valid range.

(2) Behavior with the setting "Wait for the end of monitoring"

The following figure shows the behavior of the monitoring generations when you select **Wait for the end of monitoring**.

Figure 3-49: Behavior with "Wait for the end of monitoring" defined in the jobnet's start condition



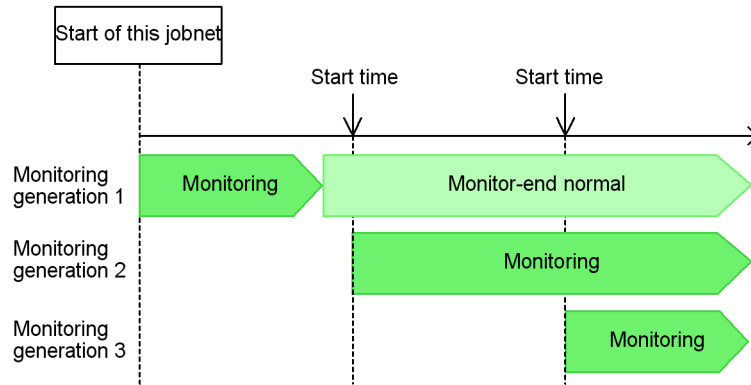
If the next start time arrives while monitoring generation 2 is in *Now monitoring* status, monitoring generation 3 enters *Wait for start time* status. When monitoring generation 2 ends, monitoring generation 3 moves from *Wait for start time* to *Now monitoring* status.

If the timeout period elapses while monitoring generation 3 is in *Wait for start time* status, monitoring for the start condition is cancelled and the generation moves to *Skipped so not executed* status.

(3) Behavior with the setting "Start monitoring"

The following figure shows the behavior of the monitoring generations when you select **Start monitoring**.

Figure 3-50: Behavior with "Start monitoring" defined in the jobnet's start condition



Monitoring generation 3 enters *Now monitoring* status when the jobnet's start time arrives even if monitoring generation 2 is in *Now monitoring* status. This behavior is selected by default.

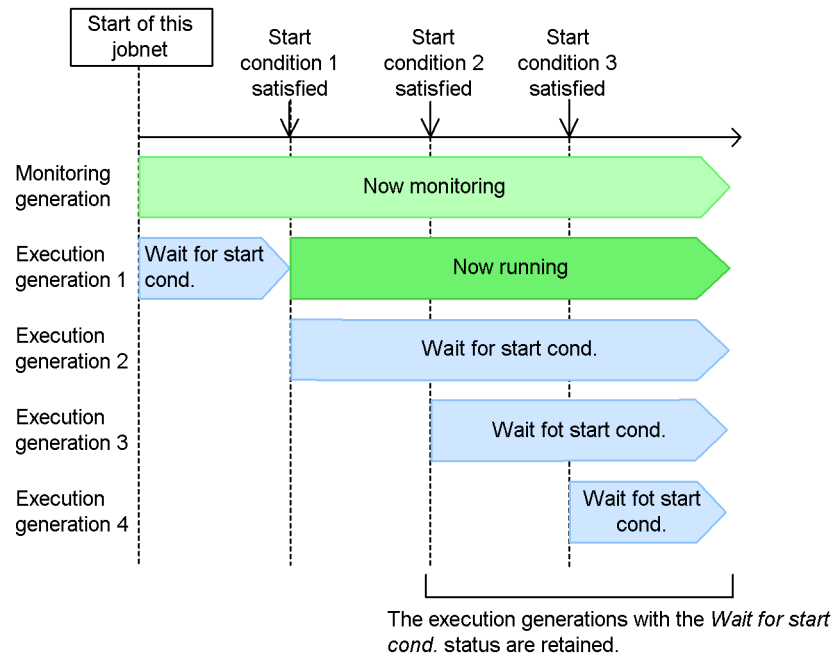
If the monitored event occurs while monitoring generations 2 and 3 are both active, each monitoring generation triggers an execution generation.

Particular care is needed when setting **Start monitoring** for a jobnet that has a set processing cycle if **Unlimited** is set as the valid range (times and period) of the jobnet's start condition.

3.4.6 Retention of execution generations with satisfied start conditions

If the concurrent execution of execution generations is not specified and a start condition is satisfied more than once while an execution generation is in the *Now running* status, multiple execution generations in the *Wait for start condition* status will be generated. By default, the generated execution generations are retained in the *Wait for start condition* status as shown in the following figure.

Figure 3-51: Example of retaining execution generations



In this example, start condition (2) is satisfied when execution generation 1 is in the *Now running* status and execution generation 2 is in the *Wait for start condition* status. Because execution generation 1 is running, execution generation 2 is retained in the *Wait for start condition* status even if start condition (2) is satisfied. Likewise, execution generation 3 is retained in the *Wait for start condition* status even if start condition (3) is satisfied.

You can select either of the following options to specify whether to retain an execution generation when the start condition is satisfied.

- **Does not stay with skip.**

If the start condition is satisfied while any preceding execution generation is in *Now running* status, the execution generation in the *Wait for start condition* status enters into the *Skipped so not executed* status and is not retained.

- **Stay without skip.**

If the start condition is satisfied while any preceding execution generation is in the *Now running* status, the execution generation in the *Wait for start condition* status is retained as is. By default, this setting is selected.

You can choose either behavior in the **Pause exec. gen. for start cond.** area of the Define Details - [Start Condition] dialog box. For details, see *15.4.5 Define Details -*

[Start Condition] dialog box in the Job Management Partner 1/Automatic Job Management System 3 Operator's Guide.

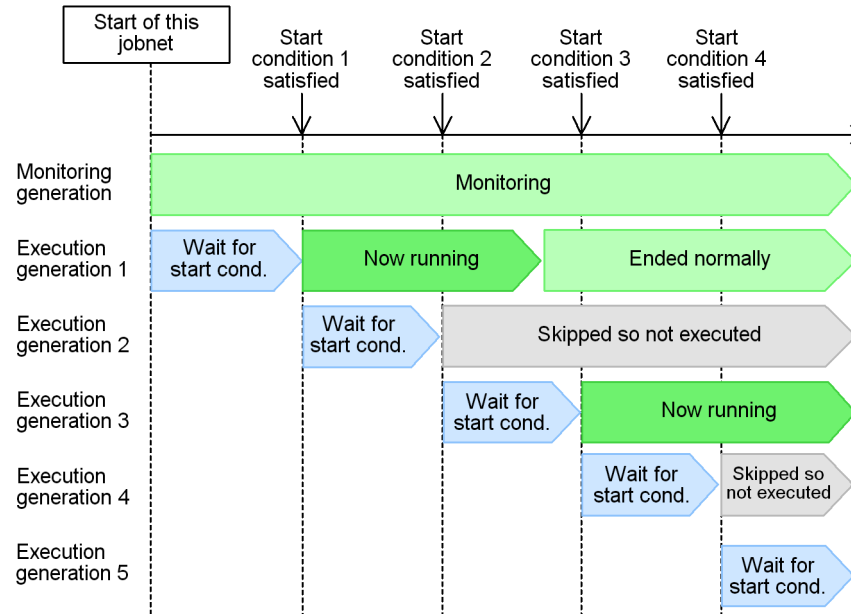
These behaviors are further described below.

Note that when concurrent execution is enabled in a jobnet definition, execution generations in the *Wait for start condition* status enter into the *Now running* status when the start condition is satisfied, regardless of which setting is selected.

(1) Behavior when "Does not stay with skip." is selected

The following figure shows the behavior of execution generations when the start condition is satisfied and **Does not stay with skip.** is selected.

Figure 3-52: Behavior when "Does not stay with skip." is selected



If start condition 2 is satisfied while execution generation 1 is running, execution generation 2 moves from *Wait for start condition* to *Skipped so not executed* status and is not retained.

Cautionary note

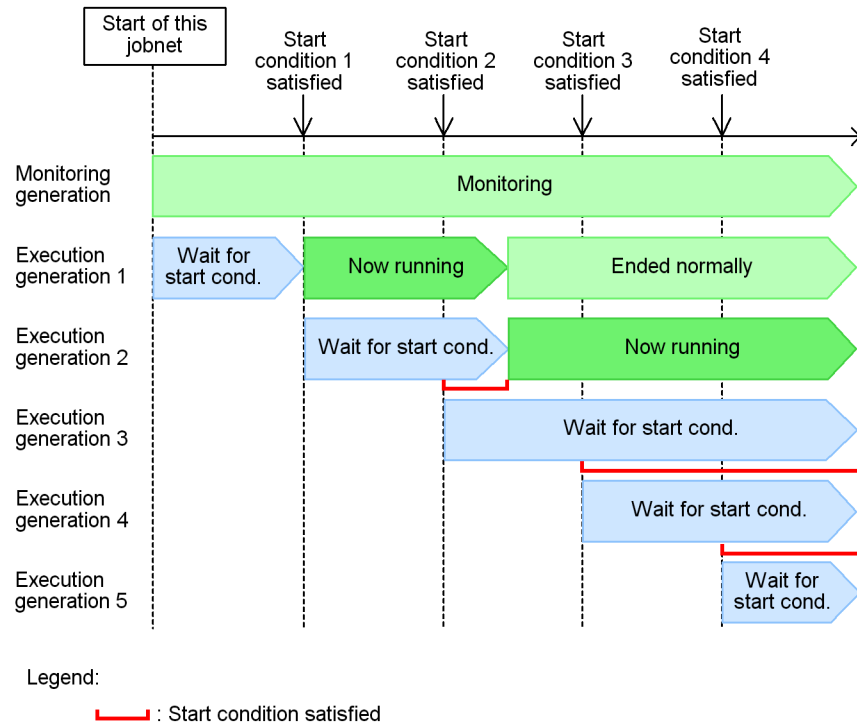
- When the hold attribute is set for an execution generation, or when **Hold start of jobnet** or **Stop monitoring of start conditions** is set in **Task after abend**, and an abnormally terminated generation is present, the new execution generation enters *Being held* status.
- If the start condition is satisfied while the executions of a job or a jobnet in

the scheduler service is suppressed, the new execution generation does not enter *Skipped so not executed* status even if a generation is running.

(2) Behavior when "Stay without skip." is selected

The following figure shows the behavior of execution generations when the start condition is satisfied and **Stay without skip.** is selected.

Figure 3-53: Behavior when "Stay without skip." is selected



If start condition (2) is satisfied while execution generation 1 is running, execution generation 2 is retained in the *Wait for start condition* status. When execution generation 1 finishes, execution generation 2 moves from *Wait for start condition* to *Now running* status. Similarly, execution generation 3 is retained until execution generation 2 finishes, and execution generation 4 is retained until execution generations 2 and 3 finish. By default, this behavior is selected.

In this way, if the start condition is satisfied while any execution generation is in *Now running* status, the same number of execution generations as satisfied start conditions will remain resident in *Wait for start condition* status. Take particular care in cases where an execution generation takes a long time to execute and the start condition is satisfied repeatedly.

3.4.7 Suppressing further executions of a jobnet with start conditions after it abnormally terminates

When a jobnet with a start condition has abnormally terminated, JP1/AJS3 can suppress subsequent execution of execution generations if the start conditions were satisfied. This suppression allows you to resume operation after correcting the cause of the abnormal termination.

JP1/AJS3 provides two suppression modes:

- **Hold start of jobnet**
Hold execution of an execution generation.
- **Stop monitoring of start conditions**
Stop monitoring for the start condition to be satisfied.

Supplementary note

To monitor a jobnet with start conditions from JP1/AJS3 - View, use the Daily Schedule (Hierarchy) window, which provides a multi-generation view.

(1) **Hold start of jobnet**

When **Hold start of jobnet** is set, if the start condition is satisfied after abnormal termination of a running execution generation, execution generations in the *Wait for start condition* status either remain in that status or are placed in the *Being held* status. This is a useful setting if you want to execute the pending generations after operation is resumed.

The suppression behavior in this mode differs according to whether the concurrent execution of execution generations is enabled. If concurrent execution is disabled, the suppression behavior also depends on whether any succeeding generations with satisfied start conditions are waiting for the completion of execution of the preceding generation when that generation terminates abnormally.

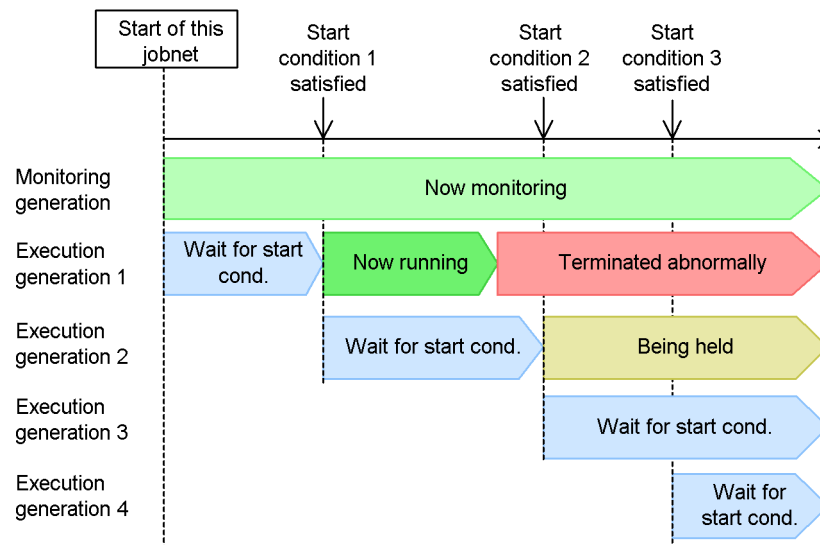
(a) **When the concurrent execution of execution generations is disabled**

When the concurrent execution of execution generations is disabled, the suppression behavior depends on whether any succeeding generations are waiting for the completion of execution of the preceding generation.

■ **When there are no succeeding generations waiting for the completion of execution of the preceding generation**

The following describes the behavior when an execution generation terminates abnormally and no generations with satisfied start conditions are waiting for the completion of execution of that generation. The following figure shows the effect of the **Hold start of jobnet** option.

Figure 3-54: Effect of the Hold start of jobnet option when the concurrent execution of execution generations is disabled and there are no succeeding generations waiting for the completion of execution of the preceding generation

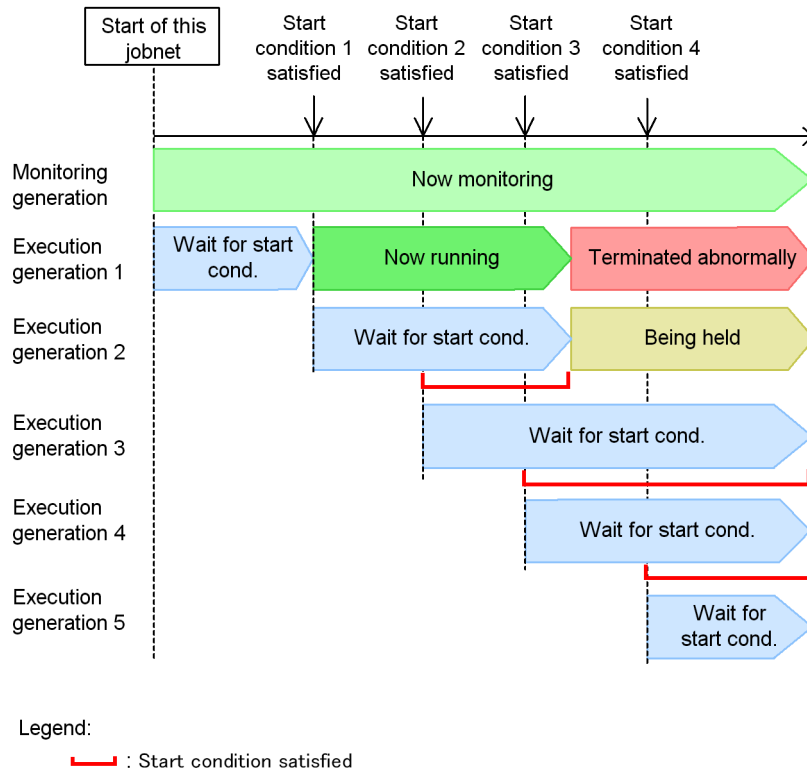


In this example, execution generation 1 terminates abnormally. Subsequently, start condition 2 is satisfied, triggering generation 2, which moves from *Wait for start condition* to *Being held* status. Start condition 3 is subsequently satisfied, but because the preceding generation (generation 2) has not ended, generation 3 remains in *Wait for start condition* status like generation 3.

■ **When succeeding generations are waiting for the completion of execution of the preceding generation**

The following describes the behavior when an execution generation terminates abnormally and generations with satisfied start conditions are waiting for the completion of execution of that generation. The following figure shows the effect of the **Hold start of jobnet** option.

Figure 3-55: Effect of the Hold start of jobnet option when the concurrent execution of execution generations is disabled and succeeding generations are waiting for the completion of execution of the preceding generation

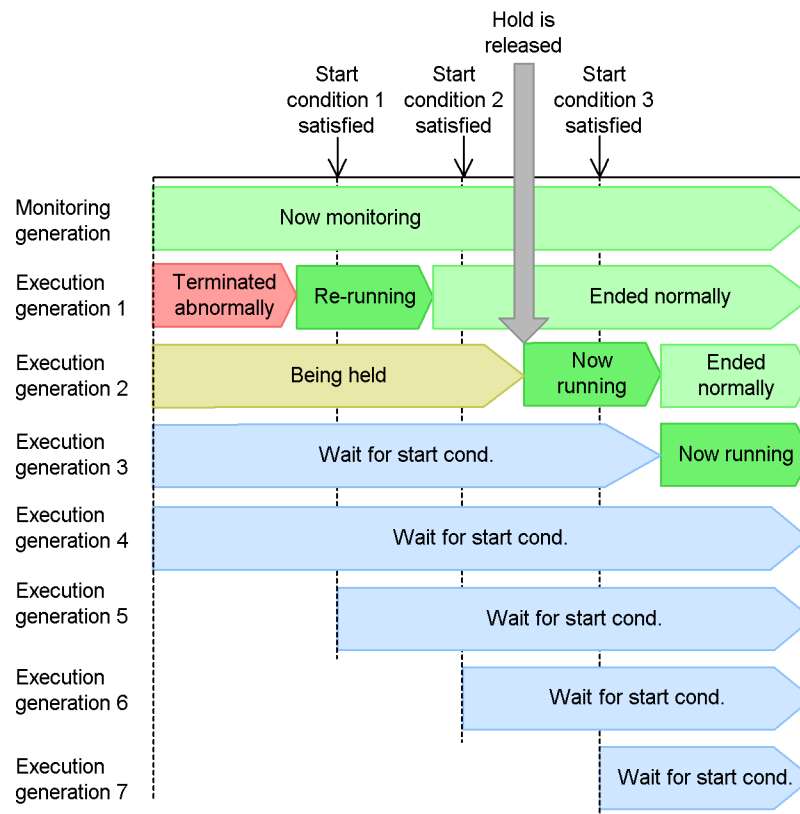


In this example, start conditions 2 and 3 are already satisfied when generation 1 terminates abnormally. Generation 2 is next to be executed. It moves from *Wait for start condition* to *Being held* status. Because the preceding generation (generation 2) has not finished, the succeeding generation 3 remains in *Wait for start condition* status, even though its start condition is satisfied. Similarly, generation 4 remains in *Wait for start condition* status even when start condition 4 is satisfied.

■ **Cautionary note**

As shown in the figure below, when concurrent execution is disabled in the jobnet definition, new generations 5, 6, and 7 will be triggered and enter *Wait for start condition* status if the start condition is satisfied while the abnormally terminated jobnet is in any of the following states: (1) Rerunning; (2) Rerun ended but **Hold start of jobnet** not yet released; (3) Running after **Hold start of jobnet** is released.

Figure 3-56: Behavior when concurrent execution is disabled and hold is released



Abnormal termination of a jobnet results in a large number of generations in *Wait for start condition* status. If there are more than 7,680 pending generations, the following message appears: KAVS0274-E The number of jobnets for execution registration exceeded the limit. (Jobnet: *jobnet-name*, code: *code*). In this case, the monitoring generation moves from *Now monitoring* status to *Monitor terminated* status, and monitoring for the start condition is terminated. For this reason, you need to restore any abnormally terminated generations and resume processing the application as soon as possible.

■ Procedure to resume operation

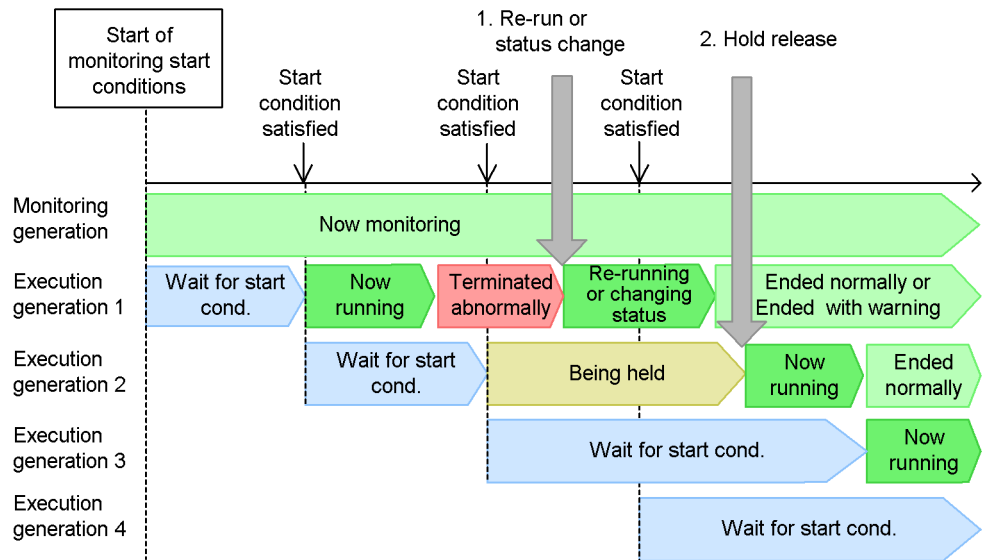
To resume operation:

1. Resolve the problem, and then either rerun the abnormally terminated generation or change its job status to *Ended normally* or *Ended with warning*.
2. Release the **Hold start of jobnet** option of the execution generation succeeding

the execution generation that terminated abnormally.

When the released execution generation has completed execution, the succeeding execution generations in *Wait for start condition* status are executed in turn.

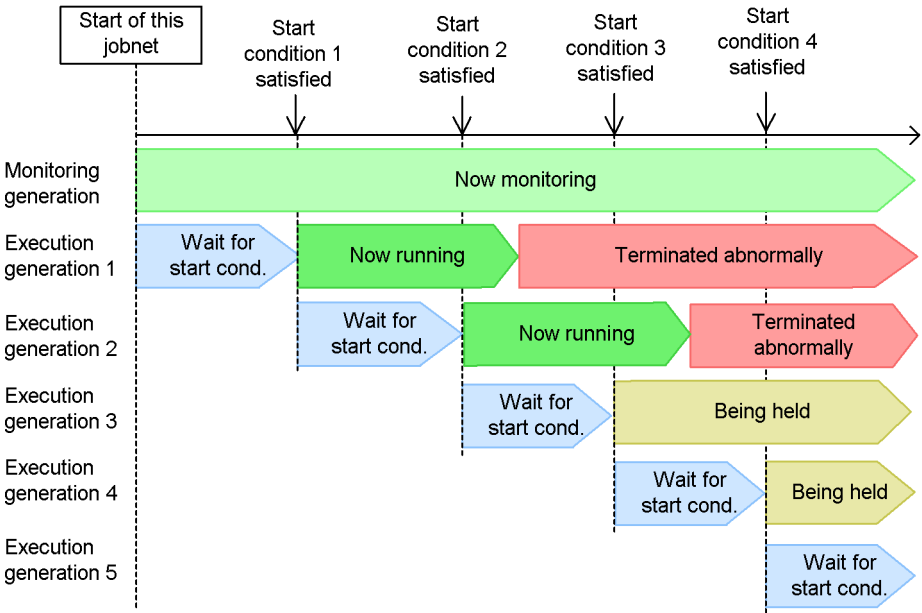
Figure 3-57: Resuming operation when concurrent execution is disabled and "Hold start of jobnet" is set



(b) When the concurrent execution of execution generations is enabled

The following figure shows the effect of the **Hold start of jobnet** option when concurrent execution is enabled for a jobnet.

Figure 3-58: Effect of the "Hold start of jobnet" option when concurrent execution is enabled

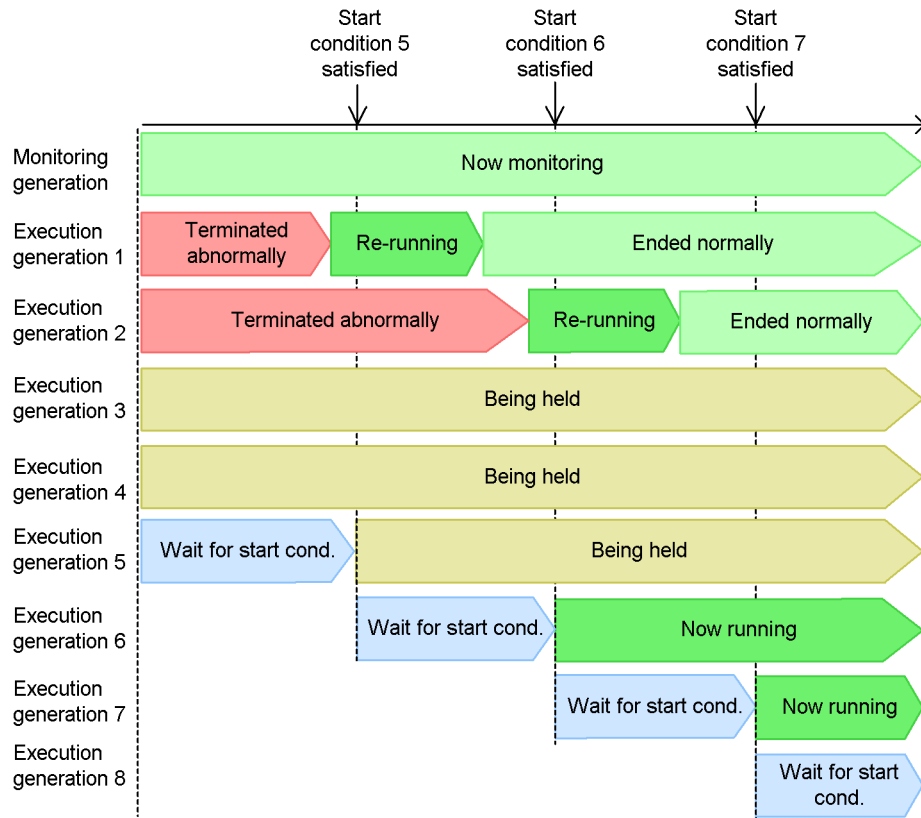


In this example, generation 2 runs concurrently with generation 1 and continues to run when execution generation 1 terminates abnormally. After generation 1 terminates abnormally, start conditions 3 and 4 are satisfied, triggering generations 3 and 4, which move from *Wait for start condition* status to *Being held* status.

Because succeeding jobnets are held when a jobnet terminates abnormally, you can release the **Hold start of jobnet** option of the jobnet anytime and resume operation.

The transition from *Wait for start condition* to *Being held* status does not occur if the abnormally terminated generations are no longer present. This applies to the timing of start conditions 6 and 7 in the figure below. When start condition 5 is satisfied, however, generation 2 is in *Terminated abnormally* status. Therefore, generation 5 moves from *Wait for start condition* status to *Being held* status.

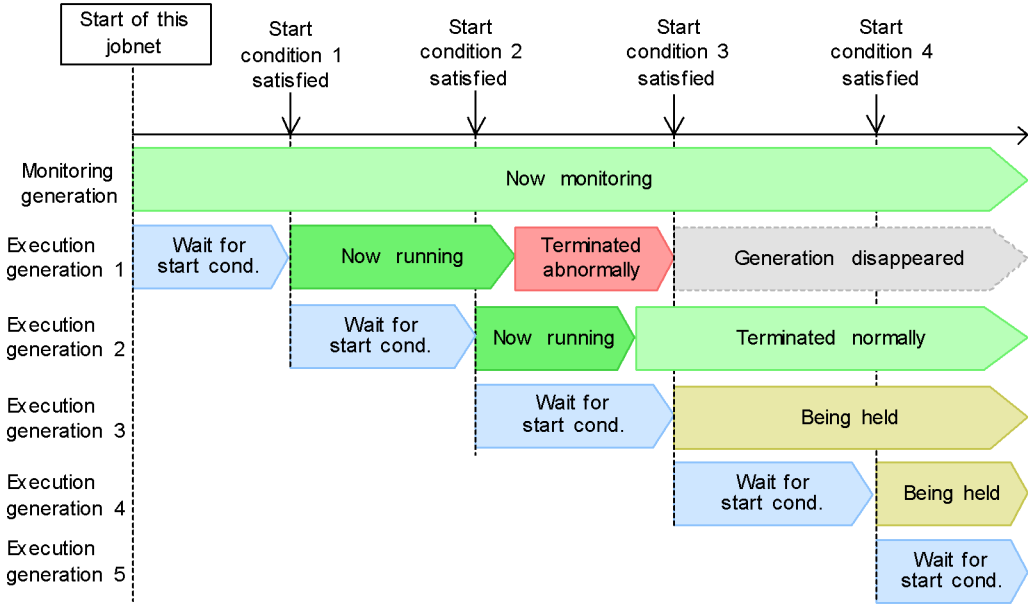
Figure 3-59: Timing of the transition from "Wait for start condition" to "Now running" status



When an abnormally terminated generation is deleted

If you set a number of logs to keep, and an abnormally terminated generation is deleted because it exceeds the set number, the generation triggered the next time the start condition is satisfied moves from *Wait for start condition* to *Being held* status. This applies to generation 4 in the following figure.

Figure 3-60: Suppression behavior when an abnormally terminated generation is deleted



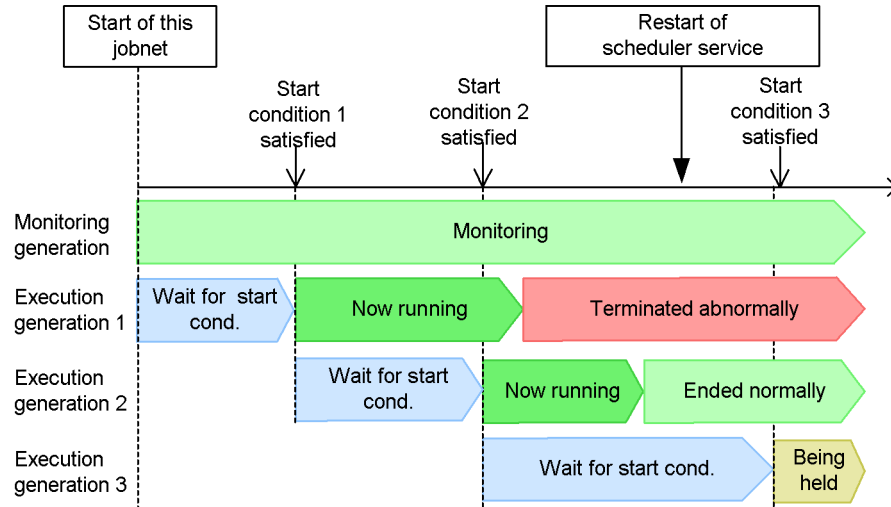
When the scheduler service is restarted

When the scheduler service is restarted (or a cluster system is failed over), JP1/AJS3 searches for terminated generations that are still present. If any abnormally terminated generations are found, succeeding execution generations move from the *Wait for start condition* status to the *Being held* status when their start condition is satisfied. If abnormally terminated generations are not found, the execution generations move from the *Wait for start condition* status to the *Now running* status when their start condition is satisfied.

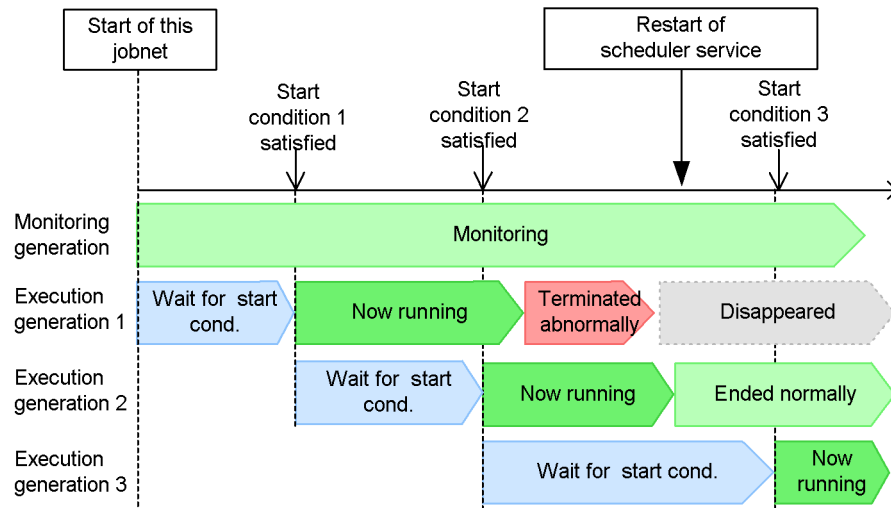
The following figure shows the effect of the **Hold start of jobnet** option when the scheduler service is restarted.

Figure 3-61: Effect of the Hold start of jobnet option when the scheduler service is restarted

■ When there are abnormally terminated generations



■ When there are no abnormally terminated generations



If execution generation 1 has terminated abnormally when the scheduler service is restarted, execution generation 3 moves from the *Wait for start condition* status to the *Being held* status when start condition (3) is satisfied.

If execution generation 1 disappears because of the specified value for the number of logs to keep when the scheduler service is restarted, execution generation 3

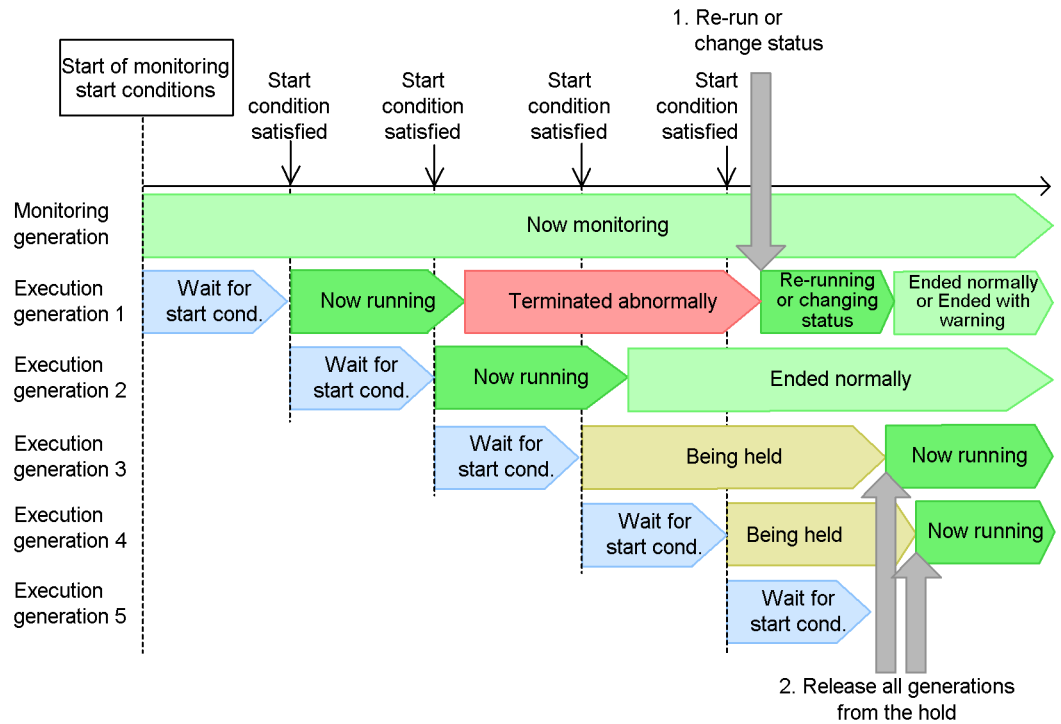
moves from the *Wait for start condition* status to the *Now running* status when start condition (3) is satisfied.

■ **Procedure to resume operation**

To resume operation:

1. Resolve the problem, and then either rerun all abnormally terminated execution generations or change their job status to *Ended normally* or *Ended with warning*.
2. After status transition of the abnormally terminated generation, release the **Hold start of jobnet** option of all execution generations in *Being held* status.

Figure 3-62: Resuming operation when concurrent execution is enabled and "Hold start of jobnet" is set



(2) Stop monitoring of start conditions

If an active execution generation terminates abnormally, the monitoring generation is placed in *Monitor terminated* status. This is a useful option if you do not want to continue monitoring after an abnormal termination.

The suppression behavior in this mode differs according to whether the concurrent execution of execution generations is enabled. If concurrent execution is disabled, the

suppression behavior also depends on whether any succeeding generations with satisfied start conditions are waiting for the completion of execution of the preceding generation when that generation terminates abnormally.

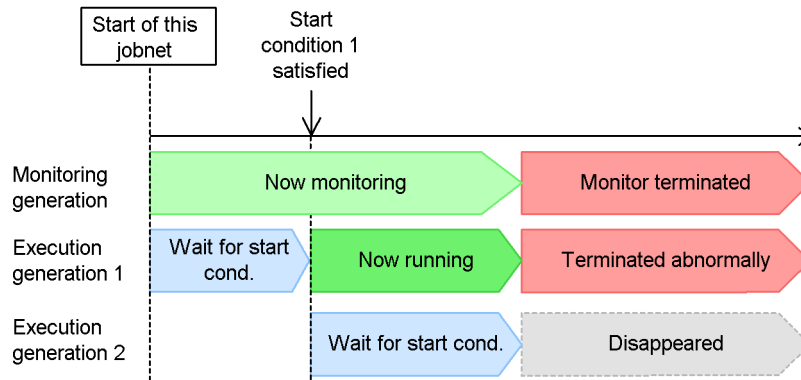
(a) When the concurrent execution of execution generations is disabled

When the concurrent execution of execution generations is disabled, the behavior when monitoring stops depends on whether any succeeding generations are waiting for the completion of execution of the preceding generation.

■ **When there are no succeeding generations waiting for the completion of execution of the preceding generation**

The following describes the behavior when an execution generation terminates abnormally and there are no generations with satisfied start conditions waiting for the completion of execution of that generation. The following figure shows the effect of the **Stop monitoring of start conditions** option.

Figure 3-63: Effect of the Stop execution monitoring of start conditions option when the concurrent execution of execution generations is disabled and there are no succeeding generations waiting for the completion of execution of the preceding generation

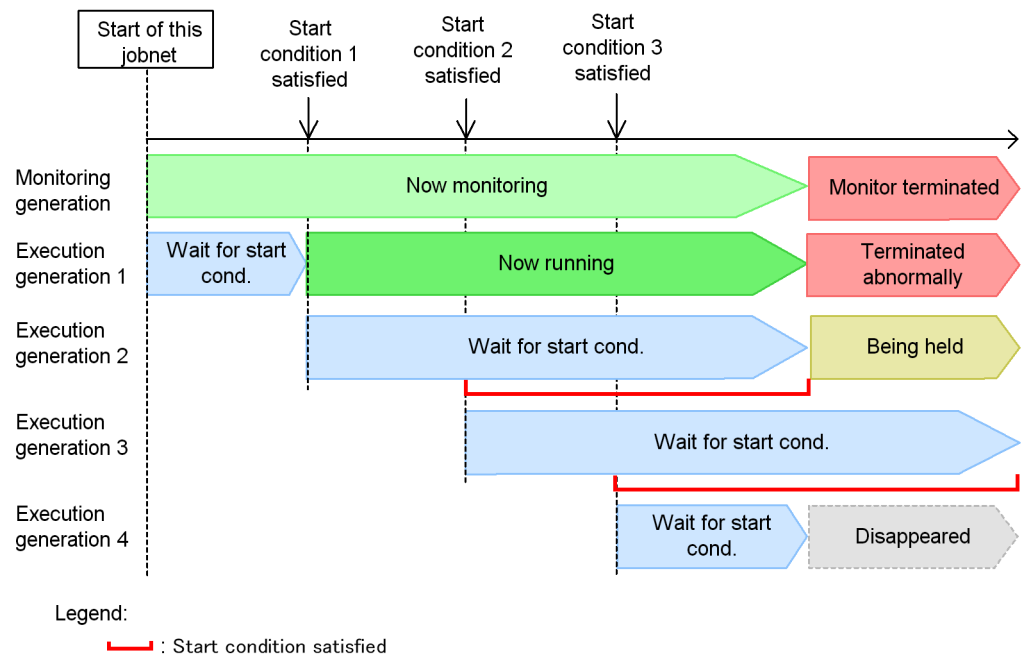


In this example, when execution generation 1 terminates abnormally, the monitoring generation moves from *Now monitoring* to *Monitor terminated* status. The waiting execution generation 2 disappears from the schedule.

■ **When succeeding generations are waiting for the completion of execution of the preceding generation**

The following describes the behavior when an execution generation terminates abnormally and generations with satisfied start conditions are waiting for the completion of execution of that generation. The following figure shows the effect of the **Stop monitoring of start conditions** option.

Figure 3-64: Effect of the Stop execution monitoring of start conditions option when the concurrent execution of execution generations is disabled and succeeding generations are waiting for the completion of execution of the preceding generation



When execution generation 1 terminates abnormally, the monitoring generation moves from the *Now monitoring* to the *Monitor terminated* status. If start conditions 2 and 3 are already satisfied, execution generation 2, scheduled to run next, moves from the *Wait for start condition* status to the *Being held* status. The succeeding generation, execution generation 3, remains in the *Wait for start condition* status, even though its start condition is satisfied, because concurrent execution is disabled for this jobnet and because execution of the preceding generation (execution generation 2) has not finished yet. Execution generation 4, which is in the *Wait for start condition* status, disappears from the schedule.

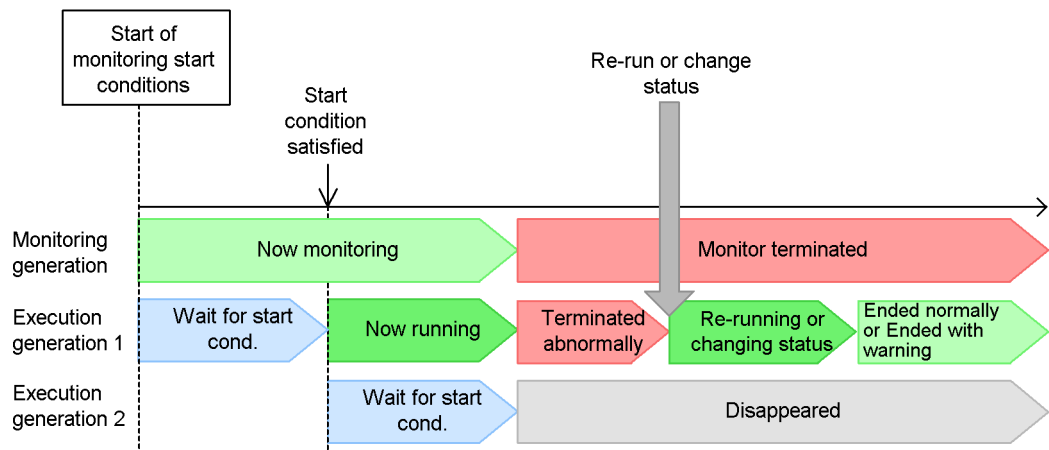
■ Procedure to resume operation

To resume operation:

When there are no succeeding generations waiting for the completion of execution of the preceding generation

Resolve the problem, and then either rerun all abnormally terminated execution generations or change their job status to *Ended normally* or *Ended with warning*.

Figure 3-65: Resuming operation when concurrent execution is disabled and "Stop monitoring of start conditions" is set (when there are no succeeding generations waiting for the completion of execution of the preceding generation)

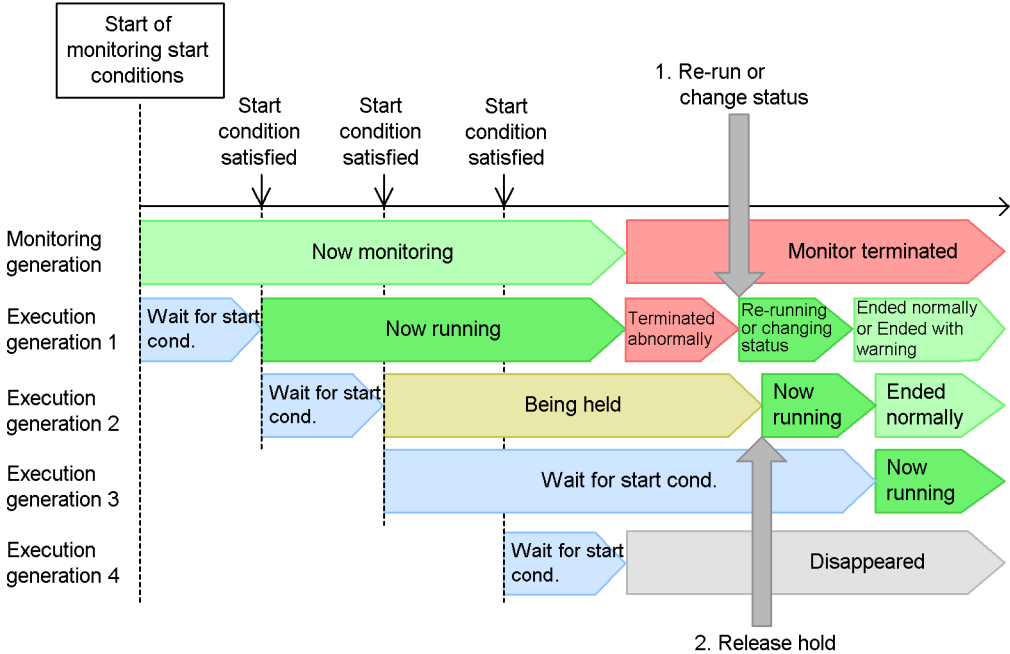


When succeeding generations are waiting for the completion of execution of the preceding generation

1. Resolve the problem, and then either rerun the abnormally terminated execution generation or change its job status to *Ended normally* or *Ended with warning*.
2. Release the **Stop monitoring of start conditions** option for the succeeding generations.

When the released execution generation has completed execution, the succeeding execution generations in *Wait for start condition* status are executed in turn.

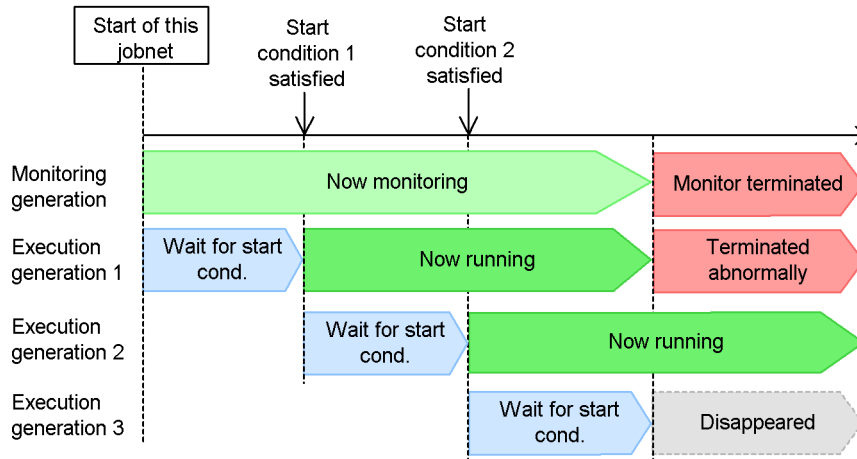
Figure 3-66: Resuming operation when concurrent execution is disabled and "Stop monitoring of start conditions" is set (when succeeding generations are waiting for the completion of execution of the preceding generation)



(b) When the concurrent execution of execution generations is enabled

The following figure shows the effect of the **Stop monitoring of start conditions** option when the concurrent execution of execution generations is enabled.

Figure 3-67: Effect of the Stop execution monitoring of start conditions option when the concurrent execution of execution generations is enabled

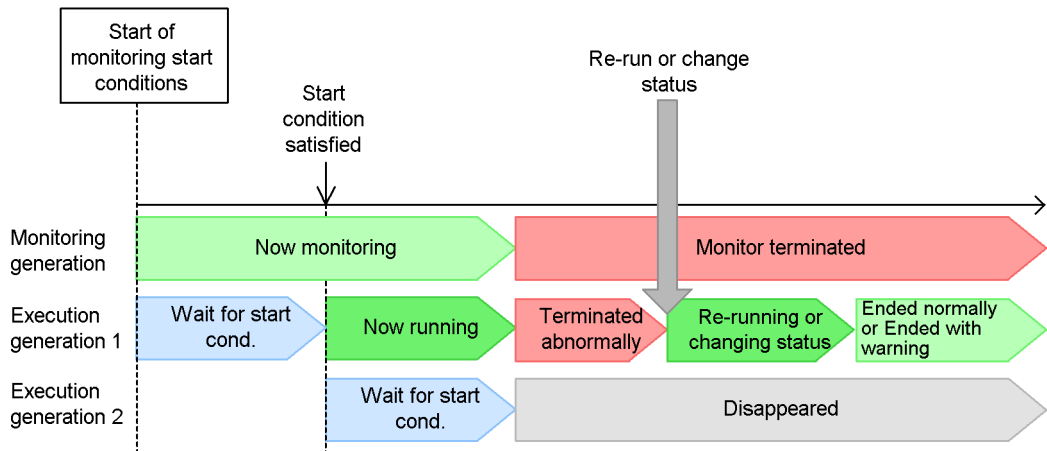


In this example, when execution generation 1 terminates abnormally, the monitoring generation moves from *Now monitoring* to *Monitor terminated* status. Generation 2, which is running concurrently with the abnormally terminated generation, continues execution. The waiting generation 3 disappears from the schedule.

■ Procedure to resume operation

Resolve the problem, and then either rerun all abnormally terminated execution generations or change their job status to *Ended normally* or *Ended with warning*.

Figure 3-68: Resuming operation when concurrent execution is enabled and "Stop monitoring of start conditions" is set



(3) Supplementary note

- Suppression of execution applies only to a root jobnet. For root jobnets, the following statuses are handled as abnormal termination:
 - Ended abnormally
 - Invalid-seq.
 - Interrupt
 - Killed
 - Skipped so not exe.
- When an execution generations terminates abnormally, the suppression setting applies only to those execution generations governed by the same monitoring generation as the abnormally terminated generation. The termination status of other monitoring generations and execution generations has no effect.
- You can suppress execution of the next monitoring generation when the last execution generation terminates abnormally. In the root jobnet's **Hold** definition, select **Hold if prev. = 'abend'** or **Hold if prev. = 'warning' or 'abend'**.
- To monitor a jobnet with a start condition from JP1/AJS3 - View, use the Daily Schedule (Hierarchy) window, which provides a multi-generation view.

3.4.8 Operations on a jobnet registered for execution with a start condition

You can perform many types of operations on a jobnet registered for execution with a start condition.

If you want to perform operations such as the forced termination or cancellation of the registration of an execution generation, perform operations according to the status of the execution generation.

- To perform an operation on an execution generation in the *Wait for start condition* status when the start condition is not satisfied or is only partially satisfied:
 Perform the operation on the monitoring generation in the *Now monitoring* status. The operation you perform on the monitoring generation in the *Now monitoring* status will apply to execution generations in the *Wait for start condition* status when the start condition has not been satisfied yet or is only partially satisfied.
- To perform the operation on an execution generation that is in the *Now running* status or that is in the *Wait for start condition* status and is waiting for the completion of execution of the preceding generation after a start condition is satisfied:

Perform the operation directly on the execution generation.

Cancel monitoring of a monitoring generation

You can cancel monitoring while still in progress by setting the monitoring generation to *Interrupted* or *Killed* status. For details about the procedure, see *9.1.1 Stopping a jobnet that is monitoring start conditions* in the *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*.

Monitor a start condition again after monitoring has finished

You can start monitoring of the start condition again by re-registering the jobnet for execution or by adding a scheduled generation of the root jobnet. You cannot monitor the start condition again by rerunning a monitoring generation that is in the *Unmonitored + Ended*, *Monitor terminated*, or *Monitor-end normal* status.

For details about how to add a scheduled generation, see *9.2 Adding an execution schedule for a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

Stop execution of an execution generation after the start condition is satisfied.

You can use the **Execution Prohibited** command to stop execution of an execution generation that is being held. This command can also stop the execution of a generation waiting for the completion of execution of the running execution generation with the start condition satisfied when concurrent execution is not specified.

For details about the procedure, see *9.5 Temporarily canceling the execution of jobnets and jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

Hold execution of an execution generation.

To hold execution of a generation, set **Hold** for the execution generation in question. The execution generation will be placed in *Being held* status when its start condition is satisfied. For details about the procedure, see *9.6 Holding the execution of jobnets and jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

Change an event job definition in the start conditions of a jobnet registered for execution

After a jobnet with start conditions is registered for execution, you can change the event job definition in a start condition during monitoring of the start conditions.

For details, see *9.1.2 Changing an event job definition in the start conditions of a jobnet registered for execution* in the *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*.

3.4.9 Start condition and number of logs to keep

A generation is created whenever a jobnet is registered for execution. The setting that specifies how many generations of execution results to save is called the *number of logs to keep*. It can be set only for a root jobnet. You can view execution results for the set number of generations in JP1/AJS3 - View or by running the `ajsshow` command. Jobnet generations are managed so that past execution results are automatically deleted when they exceed the set number of logs to keep.

When a jobnet has a start condition, jobnet generation management applies to all generations, including both monitoring generations and execution generations. For details about managing the execution logs of a jobnet that has a start condition, see *4.2.3(3) Example of generation management of a jobnet with a start condition*.

If you wish, after installation you can switch the generation management method. To do so, set the `SAVEGENTYPE` environment setting parameter. For details about this parameter, see *2.2 Setting up the scheduler service environment in the Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 2*.

Upgrade installation from version 8 or earlier

If you have performed an upgrade installation from version 8 or earlier, the method for managing the number of logs to keep differs from the method used when version 9 or later is installed. For details, see *4.2.3(4) Example of generation management of a jobnet with a start condition (upgrade installation from version 8 or earlier)*.

You can also switch from the method for managing the number of logs to keep used in version 8 and earlier to the method used in version 9 and later. However, the processing time needed to complete the switch could impact on operations. For the procedure and cautionary notes, see *4.2.3(5) Notes on setting a large number of logs to keep for a jobnet with a start condition (upgrade installation from version 8 or earlier)*.

Chapter

4. Executing an Application

In JP1/AJS3, a jobnet with defined schedule information is automatically scheduled after it is registered for execution.

This chapter describes how to register a jobnet for execution and how to check the execution schedule afterwards.

- 4.1 Registering a jobnet for execution
- 4.2 Managing jobnet generations
- 4.3 Checking execution registration information
- 4.4 Checking execution schedules
- 4.5 Operations on registered jobnets

4.1 Registering a jobnet for execution

When you have defined the schedule information for a jobnet, you can register the jobnet for execution. JP1/AJS3 schedules the jobnet, and starts operating it automatically.

This section describes how to register a jobnet for execution, how to manage jobnet generations, and how to check the execution schedule after registration.

4.1.1 Methods of registering a jobnet for execution

JP1/AJS3 begins processing a defined jobnet when you have registered it for execution. *Registration* is an operation performed on the root jobnet. When you register the root jobnet, all lower units are also registered.

JP1/AJS3 provides three methods of registering a jobnet for execution:

- Immediate execution
- Planned execution
- Fixed execution

The following table summarizes the features of each registration method.

Table 4-1: Methods of registering a jobnet for execution

Registration method	Features
Immediate execution	<p>Execution timing: The jobnet is executed as soon as it is registered, regardless of the calendar information and the schedule information set for the jobnet.</p> <p>Modifying the calendar information or schedule: Not applicable. The calendar information and jobnet schedule information are not referenced.</p> <p>Temporarily rescheduling the jobnet (by menu option or command): Not applicable. There is no execution schedule.</p> <p>Other features: You cannot restrict the number of executions. However, if you set a start condition for the jobnet, you can set an execution count or a start-condition monitoring time.</p> <p>Examples of usage: Mainly used for jobnets that are started manually or by command. Examples are a jobnet executed immediately at user request, a jobnet executed from a user program, and a jobnet executed in response to a processing result from an external program such as file transfer software.</p>

Registration method	Features
Planned execution	<p>Execution timing: The jobnet is scheduled and executed according to the calendar information and the schedule information set for the jobnet. The calculated schedule is treated as a dummy schedule (simulated execution schedule).</p> <p>Modifying the calendar information or schedule: If you change the calendar information or redefine the jobnet's schedule information, JP1/AJS3 reschedules the jobnet based on the new information.</p> <p>Temporarily rescheduling the jobnet (by menu option or command): No temporary changes can be made to a schedule calculated when the jobnet is registered (rescheduling is invalid because the schedule is simulated, not fixed). However, because the next dummy schedule will be fixed once the jobnet starts execution, you can make a temporary change to the next execution schedule.</p> <p>Examples of usage: Mainly used when changes to calendar information or schedule information may be necessary in the future, and for jobnets that never need to be temporarily rescheduled. This registration method lets you modify calendar information and schedule information without canceling the registration: for example, when open/closed days need to be redefined for the next business year, or when a jobnet schedule changes.</p>
Fixed execution	<p>Execution timing: The jobnet is executed for a specified period or for a specified number of generations. The schedule based on the specified period or generation count is calculated and fixed from the calendar information and the schedule information set for the jobnet.</p> <p>Modifying the calendar information or schedule: Changes to the calendar information or jobnet's schedule information take effect after the jobnet has run for the specified period or number of generations. To apply schedule changes, you must cancel and then re-register the jobnet.</p> <p>Temporarily rescheduling the jobnet (by menu option or command): Temporary changes can be made to a fixed schedule.</p> <p>Adding execution dates (from a menu or command): Execution dates can be added.</p> <p>Examples of usage: Mainly used for jobnets that run for a set period or for a set number of executions, and for jobnets with a fixed schedule that may require a temporary addition, change, or suspension. For example, you can change the execution schedule for a specified date, or suspend execution, without canceling the jobnet registration.</p>

The following pages describe jobnet behavior under each of these registration methods. Choose a method appropriate for your purposes, based on the features of each registration method. Jobnets can be registered for execution using JP1/AJS3 - View or by command. For details on registration procedures, see *7. Executing Jobnets*

in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide* and *ajsentry* in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*. For details on redefining or rescheduling a registered jobnet, see 4.5 *Operations on registered jobnets*.

(1) Immediate execution

When you register a jobnet for immediate execution, it is executed once only at the time of registration, regardless of the schedule definition and calendar definitions. The jobnet is executed immediately even if you have set schedule information.

Supplementary note

In JP1/AJS3, if a jobnet is not scheduled to be executed again, you can register that jobnet for execution multiple times. A jobnet registered for immediate execution is not scheduled to be executed again, so you can re-register the jobnet for execution even if the jobnet has already been registered for immediate execution.

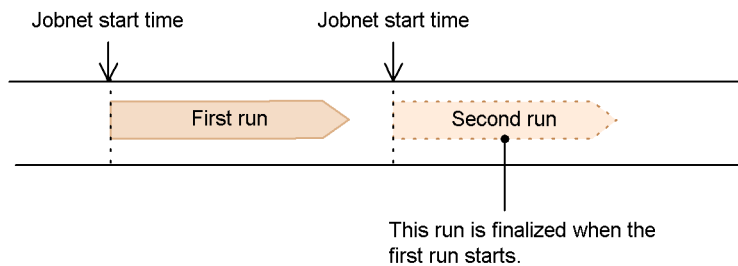
(2) Planned execution

In planned execution, the jobnet is scheduled based on its schedule definition and the calendar information set for the job group to which the jobnet belongs.

When you register a jobnet for planned execution, only its first execution is fixed, and subsequent executions are treated as dummy runs (simulated schedules). For details on dummy runs, see 4.4.2(1) *Schedule simulation*. The jobnet's next run is finalized when the current run starts.

The following figure shows generation of the next run when a jobnet is registered for planned execution.

Figure 4-1: Generation of the next execution schedule at registration for planned execution



A jobnet registered for planned execution can be re-registered when the execution schedule of the root jobnet is in *Not scheduled to execute* status.

Also, once you have registered a jobnet for planned execution, if you then redefine its schedule rule or change the calendar of the job group to which the jobnet belongs, JP1/

AJS3 immediately reschedules the jobnet according to the new settings. If you change the schedule rule of a jobnet that has an exclusive schedule, JP1/AJS3 will reschedule other jobnets on the same hierarchical level as that jobnet according to the new schedule rule settings. Note that changes to the schedule definition or calendar definition apply to the next scheduled generation. They do not affect the generation now running (zero or positive jobnet generation number). For details on jobnet generation numbers, see *4.2.2 Jobnet generation number*.

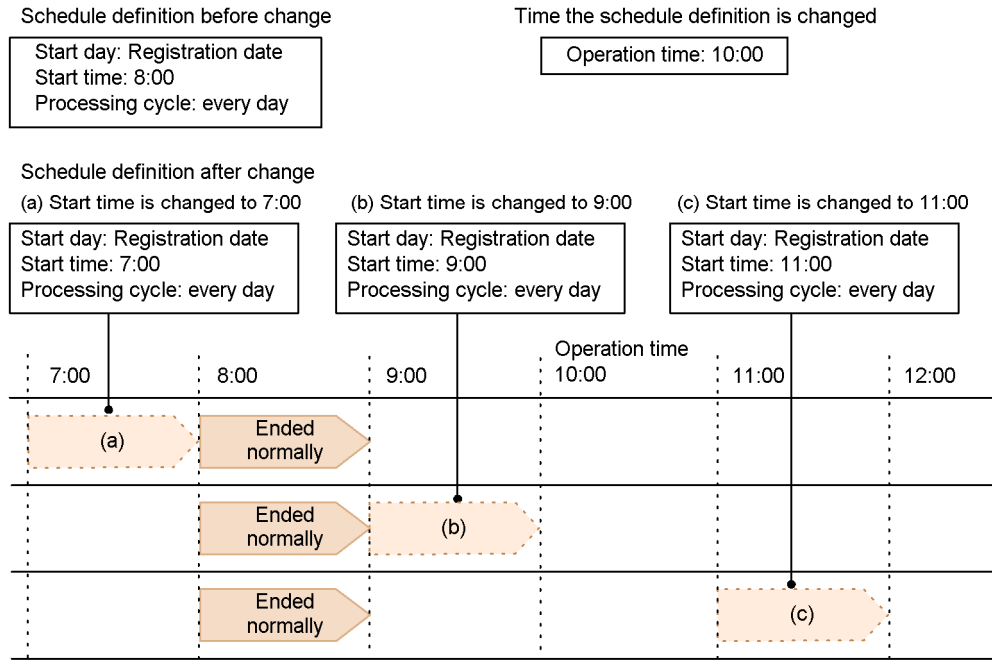
Cautionary note

Do not change the base time of a jobnet that has been registered for execution because it would complicate the schedule calculations. Also, depending on what time you set and when you make the change, the jobnet might be scheduled on that day and execute immediately. If you need to change the base time, unregister the jobnet first.

Supplementary note

While an application is running in a jobnet registered for planned execution, if the schedule definition of the jobnet is changed, JP1/AJS3 immediately recalculates the next execution time based on the new schedule definition. Depending on the change, JP1/AJS3 may create a schedule that executes the jobnet immediately, on the same day the change was made and the new schedule was calculated. The following figure shows examples of changing the schedule definition for a jobnet.

Figure 4-2: Examples of changing the schedule definition



- (a) An execution schedule for that day is not created.
- (b) An execution schedule for that day is created and the jobnet is executed immediately (at 10:00).
- (c) An execution schedule for that day is created and the jobnet is executed at the user scheduled time (11:00).

In these examples, the jobnet start time of 8:00 set in the schedule definition is changed to (a) 7:00, (b) 9:00, and (c) 11:00.

(a) Start time is changed to 7:00

The 8:00 run has already finished. No further runs will be created for that day.

(b) Start time is changed to 9:00

The 8:00 run has already finished. Because there is no execution schedule at 9:00 on that day, a 9:00 run is generated. The start time was changed at 10:00, so the jobnet is executed immediately.

(c) Start time is changed to 11:00

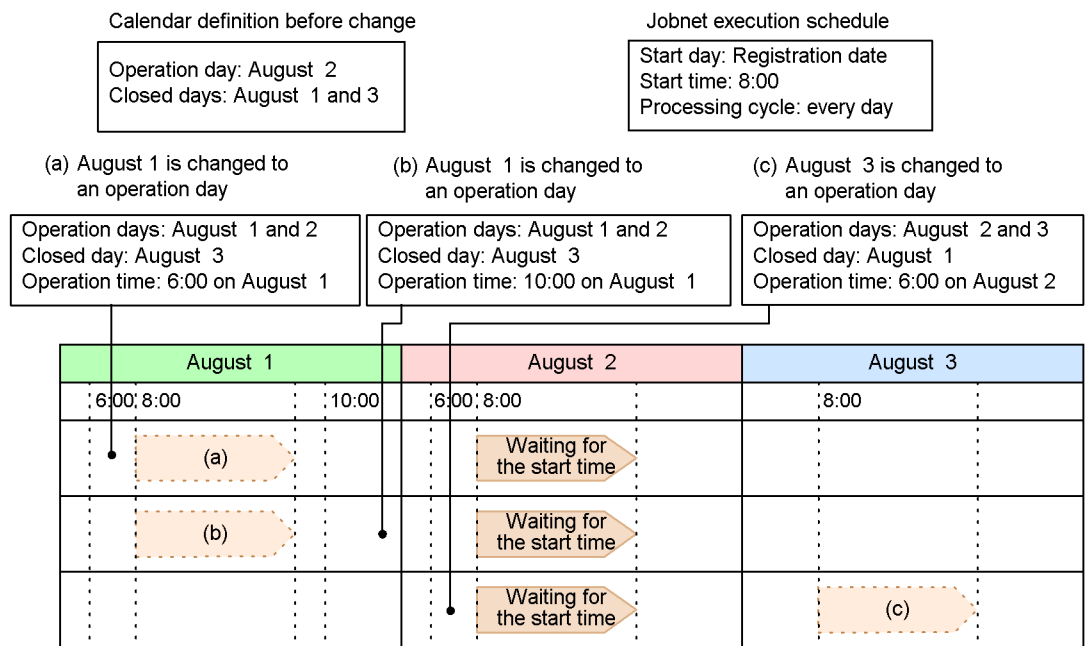
As in (b), an 11:00 run is generated. The start time was changed at 10:00, so the jobnet will be executed at 11:00.

As described above, depending on the change to the schedule definition, JP1/

AJS3 may create a schedule for execution prior to the current time, and may execute it immediately. If you want to avoid executing the new schedule on the same day the changes were made, as in examples (b) and (c), set a later start day when changing the schedule definition.

In the same way as the change to the schedule definition, if the calendar definition is changed, the schedule of jobnets registered for planned execution is re-calculated. Depending on the change, JP1/AJS3 may create a schedule that executes the jobnet on the same day, executing the jobnet immediately. The following figure shows examples of changing the calendar definition for a jobnet.

Figure 4-3: Examples of changing the calendar definition



- (a) An execution schedule for August 1 is created.
- (b) The execution schedule for August 1 is created and the jobnet is executed immediately.
- (c) A pseudo schedule for August 3 is created.

In these examples, August 2 is set in the jobnet's calendar definition as an open day, and August 1 is set as a closed day. Changing the open day to August 1 will have the following effects on the jobnet schedule:

(a) August 1 is changed to an open day (scheduled start time is after the calendar change)

Because there is no execution schedule on August 1, an execution schedule is created as the next run. The jobnet will be executed on August 1 at the 8:00

start time set in the schedule definition.

(b) August 1 is changed to an open day (scheduled start time is before the calendar change)

Because there is no execution schedule on August 1, an execution schedule is created as the next run. The 8:00 start time set in the schedule definition is already past, so the jobnet is executed as soon as the calendar definition is changed.

(c) August 3 is changed to an open day (the execution schedule on August 2 is after the calendar change)

Because the next run is scheduled for August 2, the execution schedule on August 3 is created as a dummy run. Once the jobnet starts execution on August 2, the dummy run on August 3 will become the next run and the jobnet will be executed at 8:00 on that date.

(3) Fixed execution

There are three ways of registration for fixed execution: specifying a fixed schedule period; specifying the number of future generations (number of executions); and specifying a specific date and time in addition to the schedule definition for the jobnet.

Specifying a fixed schedule period

JP1/AJS3 creates fixed execution schedules within the specified period, based on the jobnet schedule definition and the calendar information set for the job group to which the jobnet belongs.

No schedule information is generated after the fixed schedule period. (No dummy runs are generated either.)

However, you can register the same jobnet multiple times for period-based fixed execution, even if the periods overlap. If multiple schedules are generated for exactly the same time, the jobnet runs the number of times it was registered.

Specifying the number of future generations

JP1/AJS3 creates fixed schedules for the specified number of generations, based on the jobnet schedule definition and the calendar information set for the job group to which the jobnet belongs.

Dummy runs (simulated schedules) are created for each generation after the specified generation count. For details on dummy runs, see *4.4.2(1) Schedule simulation*. JP1/AJS3 continues to fix the execution schedules for the specified number of generations. At the same time, as each succeeding generation starts execution, a new execution schedule is created, the next dummy run is changed into a firm schedule. For details on generations, see *4.2 Managing jobnet generations*.

Specifying a specific date and time

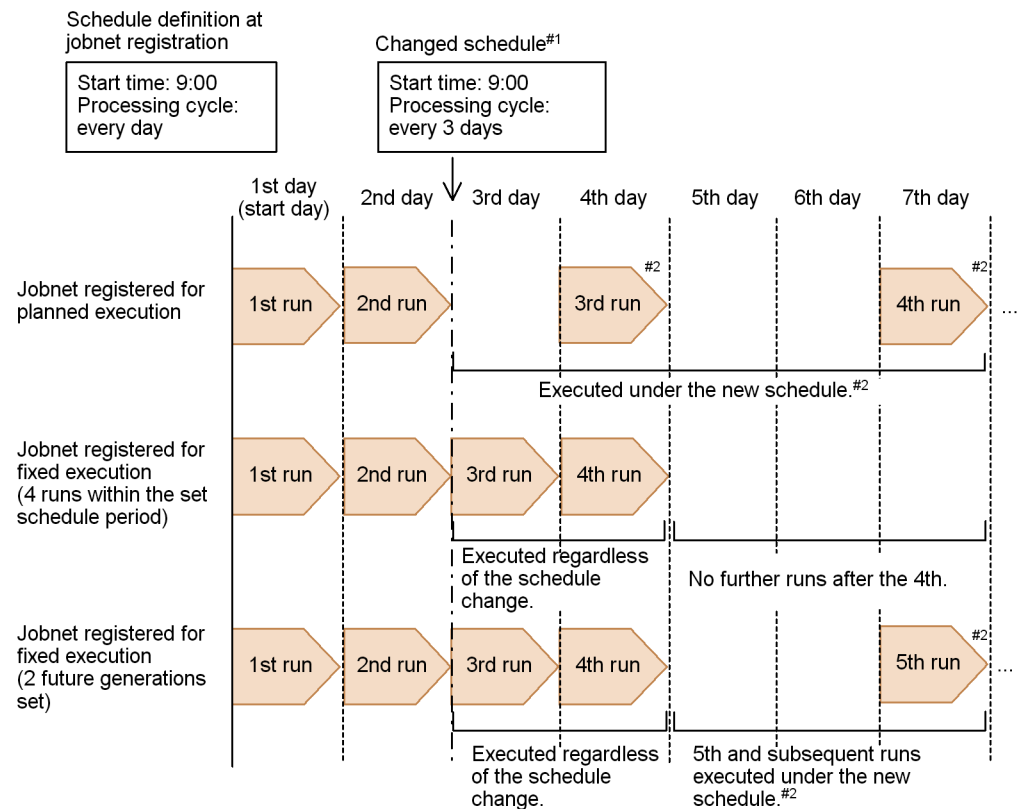
JP1/AJS3 adds a schedule for a jobnet to be executed on the specified date and time, regardless of the schedule definition for the jobnet.

For details, see 4.5.2 *Adding an execution schedule to a jobnet*.

Unlike planned execution, a jobnet registered for fixed execution is not immediately rescheduled if you redefine the jobnet's schedule rule or change the calendar information set for the job group to which the jobnet belongs.

The following figure illustrates the difference between fixed execution and planned execution.

Figure 4-4: Difference between fixed execution and planned execution



#1: Assume that the schedules were changed at completion of the 2nd run.

#2: New schedule calculated with the start day as the starting point.

When a jobnet is registered for planned execution, the schedule is recalculated as soon

as any change is made to the schedule rule or calendar definitions. In the above example, the changes made after the second run are immediately applied to the third run. In the case of the jobnet registered for fixed execution, however, the execution schedules are fixed for the set period (**Fixed schedule period**) or for the set number of generations (**Future generation**), so those execution schedules are not recalculated.

When fixed execution is specified by generation count, JP1/AJS3 continues fixing each succeeding execution schedule up to the specified number. When the first generation starts running, a new execution schedule is generated (fixed). In the above example, **2** is set in **Future generation**. This means that the third execution schedule is generated when the first starts running, and the fourth is generated when the second starts running. Each new execution schedule is based on the schedule information and calendar definition current at the time at which that schedule was generated. Therefore, in this example, the new schedule information affects the fifth and subsequent runs. In contrast, when fixed execution is specified by period, because there are no scheduled runs after that specified period, any changes to the schedule information take effect the next time the jobnet is registered for execution.

Cautionary note

- If you set both **Fixed schedule period** and **Future generation** when registering a jobnet for fixed execution, the period setting applies if there are more generations during that period than the specified generation count, but the generation count applies if there are fewer generations than specified during the set period. Dummy runs are generated for executions after the end of the specified period or generation count. These dummy runs become fixed schedules in the same way as when **Future generation** only is set.
- You can increase the number of scheduled generations after registering a jobnet for fixed execution. To do so, you must first cancel registration, change the **Future generation** setting, then re-register the jobnet for fixed execution.
- If an invalid unit (non-existent unit, for example) is specified in an exclusive schedule or in a job group that references the calendar information, the schedules after the specified generation count will not be generated.
- When different schedules are set for the root jobnet and for a nested jobnet, the nested jobnet's start time is automatically changed to the root jobnet's start time only if the root jobnet is rescheduled to a different day.
- When you register a jobnet for fixed execution by using the `ajsentry` command or JP1/AJS3 - View, a large amount of memory might be needed under certain conditions. For example, a large amount of memory is needed when the jobnet contains many units or if the jobnet has many generations. If registration of a jobnet requires too much memory, reduce the number of units in the jobnet, shorten the fixed schedule period, or reduce the number of future generations to reduce the scale of processing. See the *Release Notes*

for details on how to estimate memory requirements.

Possible errors when registering a jobnet for execution

The following settings result in an error at jobnet registration:

- Past date set as the start day, and no processing cycle set in the schedule rule.
- Invalid date (for example, 2/30) set as the start day in the schedule rule.
- Closed day set as the start day, but no closed day set in the calendar definition in the schedule rule.
- Closed day set as the start day, but **Do not execute** set in **Substitute schedule of closed day job** in the schedule rule.
- **Do not execute** set in **Substitute schedule of closed day job**, but a closed day set for every day in the calendar definition.
- Schedule rule for the root jobnet also set for a jobnet specified as an exclusive schedule (all execution schedules will therefore be treated as exclusive schedules).
- Definition such that the same execution date cannot be calculated from the schedule rule number set for the upper-level jobnet and the number of the corresponding schedule rule for the nested jobnet.
- Invalid unit (non-existent unit, for example) specified in an exclusive schedule or in a job group that references the calendar information.

In this case, the jobnet is placed in *Shutdown* status.

- Jobnets scheduled close together.

Under planned execution, schedules are adjusted dynamically according to their status and execution times. This might result in runs not being generated as scheduled.

When a schedule with any of the above settings is set for a nested jobnet, the jobnet will be placed in *Not scheduled to execute* status. You must temporarily rescheduled the jobnet to execute it.

(4) Jobnet with a start condition

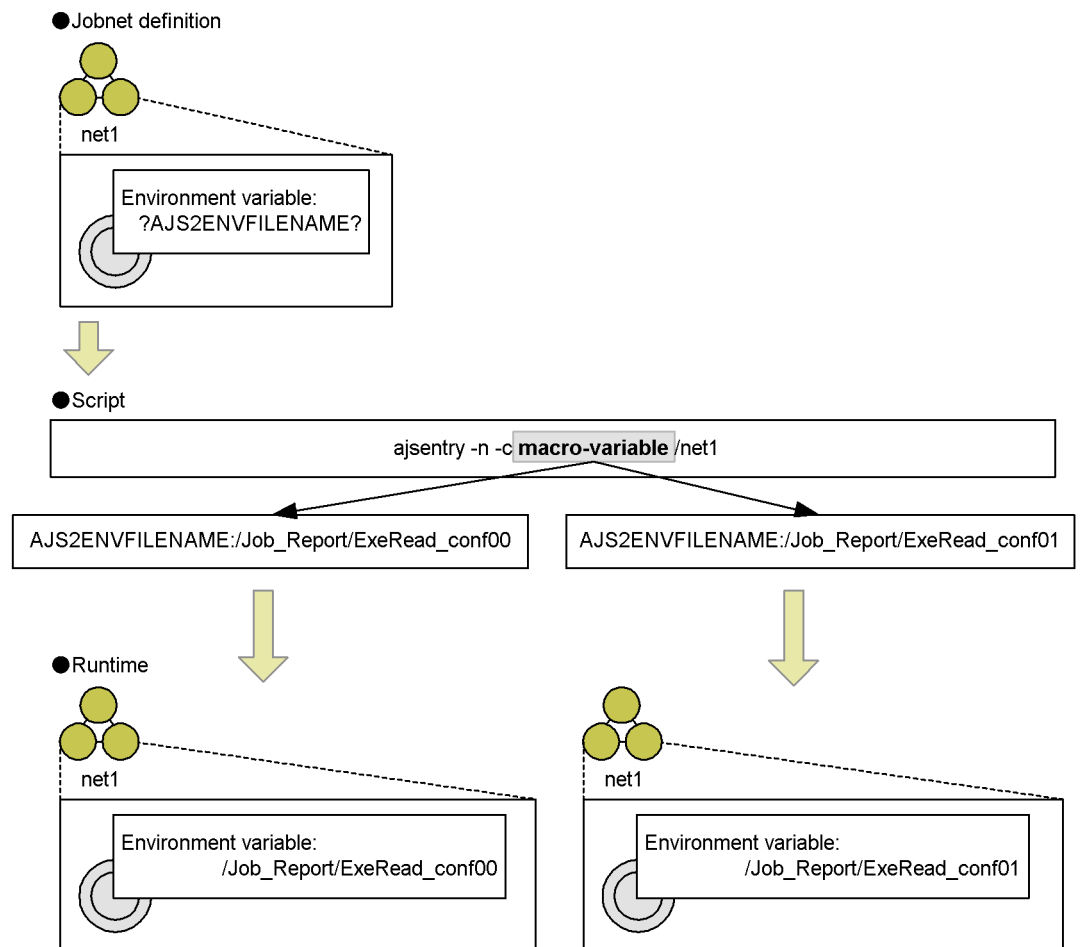
When a jobnet with a start condition is registered for planned or fixed execution, monitoring for the start condition begins at the start time specified in the schedule rule. If you register the jobnet for immediate execution, monitoring begins as soon as the jobnet is registered.

For details on the behavior of a jobnet with a start condition after it is registered, see *3.4 Defining a start condition*.

4.1.2 Specifying macro variable values during registration for execution

During registration for execution, you can specify macro variables and their values (passing information). If the specified macro variables are used in a subordinate unit, they are replaced with the specified values at runtime. With this function, you only have to create one jobnet for a number of jobnets that are the same except for some parameters. For example, to change an environment variable at each execution of a jobnet, you needed to script the following three steps for each execution: copying the jobnet (`ajscopy` command), changing the environment variable (`ajchange` command), and registering the jobnet for execution (`ajseentry` command). If you use a macro variable, however, you can perform this simply by specifying a desired value during registration for execution. The following figure illustrates processing when values of a macro variable are specified during registration for execution.

Figure 4-5: Processing when values of a macro variable are specified during registration for execution



Macro variable value specification during registration for execution is performed using JP1/AJS3 - View and the `ajsentry` command. For details about how to specify the value of a macro variable, see 7.2 *Specifying macro variable values during registration for execution* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide* and the `ajsentry` in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(1) Scope of the macro variable value

The scope of the macro variable value is as follows.

■ Macro variable value for each method of registration for execution

Macro variables and passing information can be used during registrations for planned execution, fixed execution, and immediate execution. The following table describes how the macro variable value is treated for each registration method.

Table 4-2: How the macro variable value is treated for each method of registration for execution

Method of registration for execution (option of the <code>ajentry</code> command)		How the macro variable value is treated
Registration for immediate execution (<code>-n</code>)		The specified macro variable value is applied.
Registration for planned execution (<code>-s</code>)		The specified macro variable value is applied to all the schedules. If you want to change the macro variable value that was specified during registration for execution, cancel the registration, change the macro variable value, and then re-execute registration.
Registration for fixed execution	Specifying a fixed period (<code>-p</code>)	The specified macro variable value is applied to all the generations of the jobnet to be executed within the specified fixed period. If you want to change the macro variable value for each generation, divide the period and perform registration for each division. If you want to change the macro variable value that was specified during registration for execution, cancel the registration for the desired period, change the macro variable value, and then re-execute registration.
	Specifying the number of generations (<code>-g</code>)	The specified macro variable value is applied to all of the specified future generations of the jobnet. If you want to change the macro variable value that was specified during registration for execution, cancel the registration, change the macro variable value, and then re-execute registration.
	Addition (<code>-a</code>)	The specified macro variable value is applied to the execution at the specified date and time. If you want to change the macro variable value that was specified during registration for execution, cancel the registration, change the macro variable value, and then re-execute registration.

■ Macro variable values during re-execution

The macro variable and passing information specified during registration for execution are applied without change during re-execution also.

■ Macro variable values during a temporary plan change

Macro variables and passing information specified at the time of registration for execution continue to apply even during a temporary plan change.

■ Macro variable values with identical names

If the name of the macro variable specified during registration for execution has

already been specified for the preceding job, the passing information for the macro variable of the preceding job is applied wherever the preceding job uses the passing information. If the same macro variable name is specified in the start condition, the name specified in the start condition prevails.

■ Macro variable values when concurrent execution is enabled

The macro variable value specified during registration for execution is applied to all the generations of the jobnet, regardless of whether the jobnet allows concurrent execution.

(2) How to check macro variables

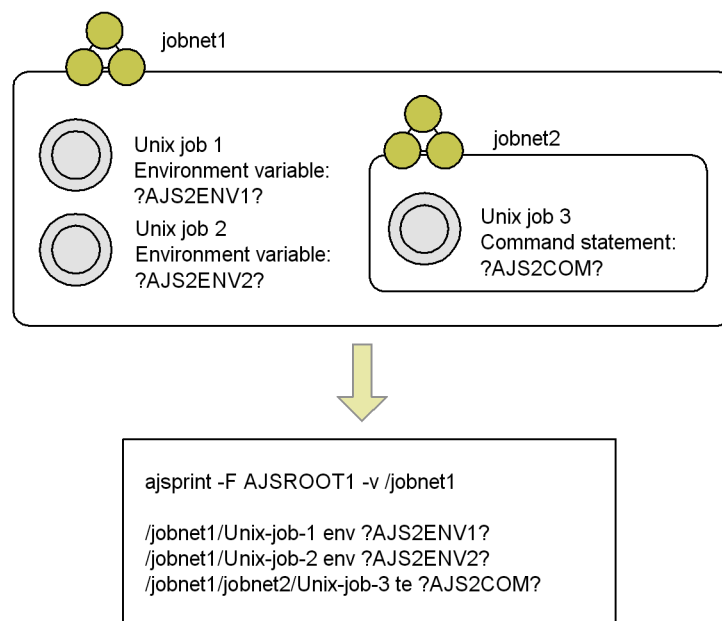
The following describes how to check macro variables before and after registration for execution.

(a) Checking macro variables before registration for execution

JP1/AJS3 - View or the `ajsprint -v` command can display a list of macro variables used in subordinate units. For details, see *15.3.21 Variables Used dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide* and the `ajsprint` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

The following figure illustrates an example of listing macro variables by using the `ajsprint` command.

Figure 4-6: Example of listing macro variables used in subordinate units



(b) Checking macro variables after registration for execution

JP1/AJS3 - View can display the macro variables and passing information specified during registration for execution. Also, JP1/AJS3 - View can display the macro variables passed during job execution and the results of passing. For details, see *15.7.10 Inherit Result dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

You can also use the `ajsshow` command to check macro variables after registration for execution. By specifying a `%MV` two-byte format indicator in the `-i` option, the command can display the macro variables passed during job execution and the results of passing. When there are multiple output results, they are separated using a comma (`,`). When there are multiple generations, they are output one to a line. An example of the output results of the `ajsshow` command is shown below:

```
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"
```

For details about the `ajsshow` command, see the *ajsshow* in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

The following table lists the unit types, unit statuses, and whether the results of passing can be displayed.

Table 4-3: Whether the results of passing can be displayed

Unit type	Unit status	Whether the results of passing can be displayed
Root jobnet	<i>Not registered</i>	--
	Other statuses	Yes ^{#1}
Nested jobnet	--	--
Job ^{#2, #3}	<i>Not registered</i>	--
	<i>Not sched. to exe.</i>	--
	<i>Waiting for prev. to end</i>	--
	<i>Being held</i>	--

Unit type	Unit status	Whether the results of passing can be displayed
	<i>Now queuing</i>	--
	<i>Waiting to execute</i>	--
	<i>Now running</i>	--
	<i>Ended normally</i>	Yes
	<i>Normal end + False</i>	Yes
	<i>Ended with warning</i>	Yes
	<i>Ended abnormally</i>	Yes
	<i>Failed to start</i>	Yes
	<i>Unknown end status</i>	Yes
	<i>Killed</i>	Yes
	<i>Not executed</i>	--
	<i>Bypassed</i>	--
	<i>Interrupted monitoring</i>	Yes
	<i>Shutdown</i>	--

Legend:

Yes: The results can be displayed.

--: The results cannot be displayed.

#1

Remote jobnets are excluded.

#2

The following jobs are excluded:

- The jobs subordinate to remote jobnets
- The jobs deleted during suspension
- The jobs added during suspension (other than those added after suspension is canceled)

#3

If the job status is changed, whether results can be displayed depends on the new status.

The results of passing for a job are displayed based on the passing information at that time. By displaying the results of passing, you can check the information to be passed when the job is re-executed.

Supplementary note

The results of passing differ from the previous execution in the following cases:

- When the preceding job from which information is passed is being-re-executed:
The information passed from the preceding job is not displayed.
- When the preceding job from which information is passed was re-executed and the re-execution has finished:
The passing information after the preceding job is re-executed is displayed.
- When the preceding job from which information is passed was deleted during suspension:
The information passed from the preceding job is not displayed.

(3) Setting the status when replacement of a macro variable has failed

The `MACROCHANGEFAIL` environment setting parameter can set whether to change the status of the job to *Ended abnormally (Failed to start)* if replacement of a macro variable has failed.

For details about how to use the parameter, see *6.2.10 Setting the action to be taken if the value of a macro variable cannot be passed* in the *Job Management Partner 1/ Automatic Job Management System 3 Configuration Guide 1 (Windows)* or *14.2.10 Setting the action to be taken if the value of a macro variable cannot be passed* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

(4) Cluster system

Macro variable values specified during registration for execution can be used in all the cluster software products supported by JP1/AJS3.

The following subsections describe the behavior if a failover has occurred.

(a) When the service start mode is set to "Cold"

Only the jobnet and job definition information immediately before the failover has occurred is inherited, and all the jobnets are unregistered. Therefore, the macro variables and values specified during registration for execution are canceled.

(b) When the service start mode is set to "Warm"

The jobnet is placed in the *Interrupted* status. The jobnets that were not started will be executed according to the schedule. For a jobnet abnormally terminated by a warm start, check the status, and then re-execute the jobnet manually. When the jobnet is re-executed, the macro variables and values specified during registration for execution are applied.

(c) When the service start mode is set to "Hot"

The status immediately before the failover is inherited. If the actual job status was successfully acquired, the jobnet is automatically resumed based on the definition, and the macro variables and values specified during registration for execution are applied. If JP1/AJS3 has failed to acquire information from the server where the job was executed, JP1/AJS3 places the job in the *Abnormal end* status. If this has occurred, check the job status, and then re-execute the jobnet manually. When the jobnet is re-executed, the macro variables and values specified during registration for execution are applied.

(5) Cautionary Notes

This subsection provides notes on specifying macro variable values during registration for execution.

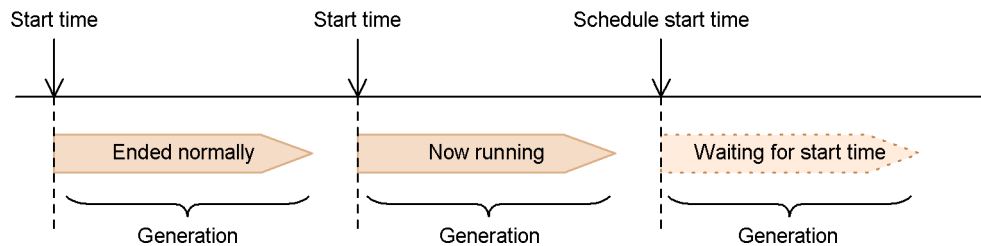
- If macro variable values are specified during registration for execution, the specified information is stored in a file for each root jobnet or for each generation. Since the file must be deleted when the registration for execution is canceled, specification of macro variable values during registration for execution degrades the system performance. To reduce the turnaround time for unregistration, set asynchronous unregistration for the scheduler service (by setting the BACKGROUNDLEAVE environment setting parameter to *yes*). For details about settings for BACKGROUNDLEAVE, see *6.1.6 Changing the mode in which unregistration or generation management deletes the generations of a jobnet in the Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or *14.1.6 Changing the mode in which unregistration or generation management deletes the generations of a jobnet in the Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.
- For a root jobnet that is a remote jobnet, macro variable values cannot be specified during registration for execution. Also, for a remote jobnet that is a nested jobnet, macro variable values cannot be specified for any dependent jobnet during registration for execution.

4.2 Managing jobnet generations

When you register a jobnet for execution, JP1/AJS3 generates the required execution runs, based on the schedule definition. Each of these runs, executed at a specific start time, is known as a *generation*.

The following figure illustrates jobnet generations.

Figure 4-7: Jobnet generations



For example, if a jobnet is executed at 10:00, 11:00, and 12:00, it has three generations. Each jobnet generation has a specific status.

4.2.1 Managing the scheduled generations of a jobnet

The forthcoming generations of a jobnet registered for execution are known as *scheduled generations*.

When you register a jobnet for planned execution, an initial scheduled generation is created at registration, and each subsequent scheduled generation is created as the previous generation starts running.

When you register a jobnet for period-based fixed execution, the runs that fall within the specified schedule period, calculated from the schedule definition and calendar definitions, will be the scheduled generations. When you register a jobnet for generation-based fixed execution, the runs up to the specified generation count, calculated from the schedule definition and calendar definitions, will be the scheduled generations.

When a jobnet has a preset start condition, the next scheduled generation is created when the jobnet enters *Now monitoring* status. For details on start conditions, see 3.4 *Defining a start condition* and 4.2.3(3) *Example of generation management of a jobnet with a start condition*.

4.2.2 Jobnet generation number

Jobnet generations are managed using numbers called *execution IDs*. An execution ID is a unique serial number assigned to each generation of a jobnet as it is scheduled for

execution. Each generation also has a *generation number*, assigned in order of execution. For root jobnets in a planning group, numbers are assigned in alphabetical order of the root jobnet names.

The next scheduled generation has 0 as its generation number. Past generations are numbered 1, 2, 3 and so on, going back into the past, and future generations are numbered -1, -2, -3 and so on, going forward into the future. An example of assigning generation numbers is shown below.

Figure 4-8: Example of assigning generation numbers

Current date/time is 10:00 on 8/10

Date	8/7	8/8	8/9	8/10	8/11	8/12	8/13
Scheduled start time	12:00	12:00	12:00	12:00	12:00	12:00	12:00
Generation	3	2	1	0	-1	-2	-3
Execution ID	@A100	@A101	@A102	@A103	@A104	@A105	@A106

If the current time is 10:00 on 8/10, the next generation, which is due to start at 12:00 on 8/10, will be generation 0. Generation numbers change as jobnet execution progresses. For example, when the 8/10 generation starts execution, its generation number changes from 0 to 1, and all the previous generation numbers are also incremented by 1. The 8/11 generation number changes from -1 to 0, and all the future generations from 8/12 also increase by 1. This assignment of generation numbers applies even when **Hold if prev. = 'abend'** or **Hold if prev. = 'warning' or 'abend'** is set as the **Hold** attribute in the jobnet definition. For example, if **Hold if prev. = 'abend'** is set as the hold attribute, and the 8/9 generation ends abnormally, the 8/10 generation will become generation 0, which will be executed after being placed in held status.

Execution IDs, on the other hand, remain unchanged after each is assigned when a jobnet generation is created (scheduled for execution). That is, when the 8/10 generation is executed, its generation number changes from 0 to 1, but its execution ID (@A103) remains unchanged.

However, if you perform an operation that changes the execution order of the generations, such as temporarily changing a schedule or adding an execution schedule, the generation numbers are immediately reassigned. The following example shows how the generation numbers change when an execution schedule is added.

Figure 4-9: Example of changed generation numbers after adding an execution schedule

Fixed schedule period: 8/7 - 8/13
 Processing cycle: Every 1 days

Current time: 10:00

Date	8/7	8/8	8/9	8/10	8/11	8/12	8/13
Scheduled start time	12:00	--	12:00	--	12:00	--	12:00
Generation	1	--	0	--	-1	--	-2
Execution ID	@A100	--	@A101	--	@A102	--	@A103

Legend:
 -- : Not scheduled

↑
 When a 8/10 run is added to these schedule rules, the generation numbers change as follows.

Current time: 10:00

Date	8/7	8/8	8/9	8/10	8/11	8/12	8/13
Scheduled start time	12:00	--	12:00	12:00	12:00	--	12:00
Generation	1	--	0	-1	-2	--	-3
Execution ID	@A100	--	@A101	@A104	@A102	--	@A103

Legend:
 --: Not scheduled

↑ ↑ ↑
 Each generation becomes the preceding number.

↑
 The new generation for fixed execution (@A104) starts between generations @A101 and @A102, so it is assigned generation number -1.

In this example, the current time is 10:00 on 8/9. Because the 8/9 generation is generation number 0, the new 8/10 generation is assigned generation number -1. Consequently, the 8/11 generation (previously -1) becomes -2, and the 8/13 generation becomes -3. The execution IDs, however, are fixed and unchanged. Because execution IDs are assigned as runs are generated (in the order in which the schedules are fixed), the additional run on 8/10 is assigned a new execution ID (@A104 in the example).

If you temporarily change a plan, add an execution schedule, or perform any other operation that results in two or more generations having the same scheduled start time, the generation whose start time was changed becomes the succeeding generation. This also applies to the root jobnets in a planning group.

The following figure shows the resulting generation numbers when a schedule is temporarily changed so that it has the same start time as another schedule.

Figure 4-10: Generation numbers when a temporary change results in two schedules having the same start time

Fixed schedule period: 8/7 - 8/13

Processing cycle: Daily

Current time: 10:00

Date	8/7	8/8	8/9	8/10	8/11	8/12	8/13
Scheduled start time	12:00	12:00	12:00	12:00	12:00	12:00	12:00
Generation	3	2	1	0	-1	-2	-3
Execution ID	@A100	@A101	@A102	@A103	@A104	@A105	@A106

When the scheduled start time of the 8/11 generation (@A104) is shifted to 12:00 on 8/10, and 'Not in next schedule' is selected, the generation numbers change as follows.

Current time: 10:00

Date	8/7	8/8	8/9	8/10		8/11	8/12	8/13
Scheduled start time	12:00	12:00	12:00	12:00	12:00	--	12:00	12:00
Generation	3	2	1	0	-1	--	-2	-3
Execution ID	@A100	@A101	@A102	@A103	@A104	--	@A105	@A106

Legend:

--: Not scheduled

The new start time of the 8/11 generation (@A104) coincides with the start time of the 8/10 generation (@A103), so generation @A104 comes after generation @A103.

In this example, the schedule due to start at 12:00 on 8/11 is moved to 8/10, resulting in two generations with the same start time. The changed generation (the generation previously scheduled for 8/11) becomes the succeeding generation. The jobnet is actually executed once only.

4.2.3 Jobnet generation management

The number of logs to keep is a setting that specifies how many generations are to be saved as the execution results of a jobnet. The setting applies only to a root jobnet. Once you set the number of logs to keep, you can view the execution results for the set number of generations (executions) in the Daily Schedule or Monthly Schedule window. You can also view past execution results by running the `ajsshshow` command. For details on the `ajsshshow` command, see `ajsshshow` in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*. As the number of logs to keep, you can set any number from 1 to 99. Using the extended feature, you can set a maximum of 999 generations, depending on the

environment settings for the Manager's scheduler service. For details on setting the Manager's scheduler service environment, see `MAXSAVEGEN` in 2.2 *Setting up the scheduler service environment* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 2*. However, a high setting will simply increase the number of saved records (given by the following equation: number of logs to keep setting x number of registered units). This means that all sorts of operations involving database access, such as canceling jobnet registration, will take a very long time. If you are using the extended feature for maximum number of logs to keep, make sure the setting will not degrade system performance. You should also promptly delete past results (by date specification) when they are no longer required.

Note that the extended feature is not supported in JP1/AJS3 - Manager 06-71 or earlier versions.

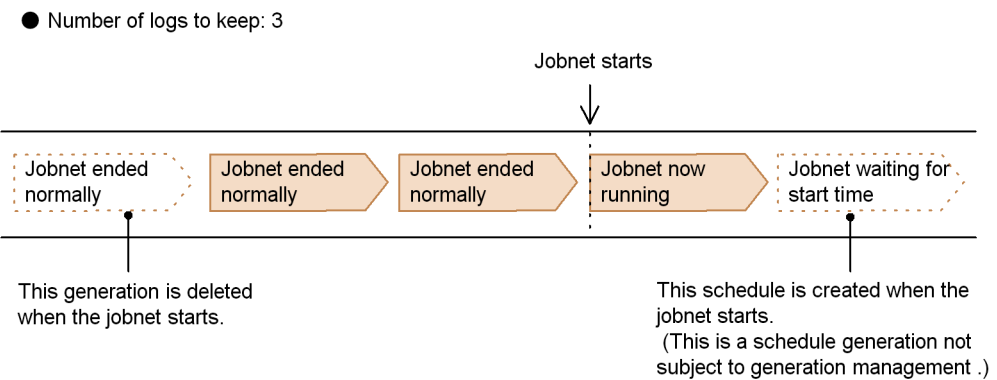
Past execution results are automatically deleted when they exceed the number of generations set as the number of logs to keep. This is referred to as *jobnet generation management*. Examples of jobnet generation management when executing a jobnet and notes on using this feature are given next.

(1) Example of jobnet generation management

Jobnet generation management is performed at the execution start time of the next generation.

The following figure shows an example of jobnet generation management.

Figure 4-11: Example of jobnet generation management



In this example, 3 is set as the number of logs to keep in the jobnet definition. When the fourth generation starts execution, the generation executed first is deleted.

(2) Example of jobnet generation management when "Now running" generations exceed the number of logs to keep

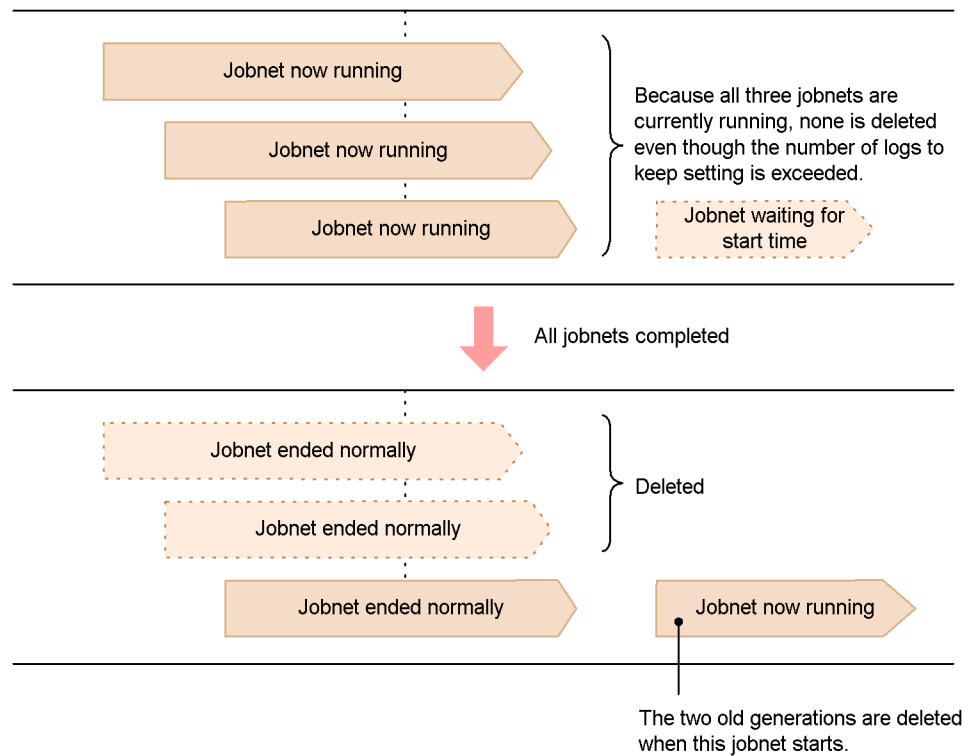
Generations in *Now running* status, and generations in *Waiting for start time* status

when concurrent execution is disabled, are not deleted when the number of logs to keep is exceeded. However, *Now running* and *Waiting for start time* generations are counted among those subject to generation management. If the number of *Now running* or *Waiting for start time* generations exceeds the number of logs to keep, the excess generations are deleted when execution completes.

An example of this situation is shown below.

Figure 4-12: Jobnet generation management when the "Now running" generations exceed the number of logs to keep (at rerun)

- Number of logs to keep: 2



In this example, 2 is set as the number of logs to keep in the jobnet definition. Even though this value is exceeded while the three generations are in *Now running* status, none of the generations is deleted. Generation management is performed when the next scheduled generation starts. At this point, generations that exceed the setting (2), counting the new generation now running, are deleted, starting from the oldest.

(3) Example of generation management of a jobnet with a start condition

When a jobnet has a start condition, generation management applies to all generations, including the monitoring generations (generations in *Now monitoring* status) and execution generations initiated by the start condition (generations in *Wait for start condition* status). Generation management is performed at the following times.

- When monitoring for the start condition begins
- When the start condition is satisfied

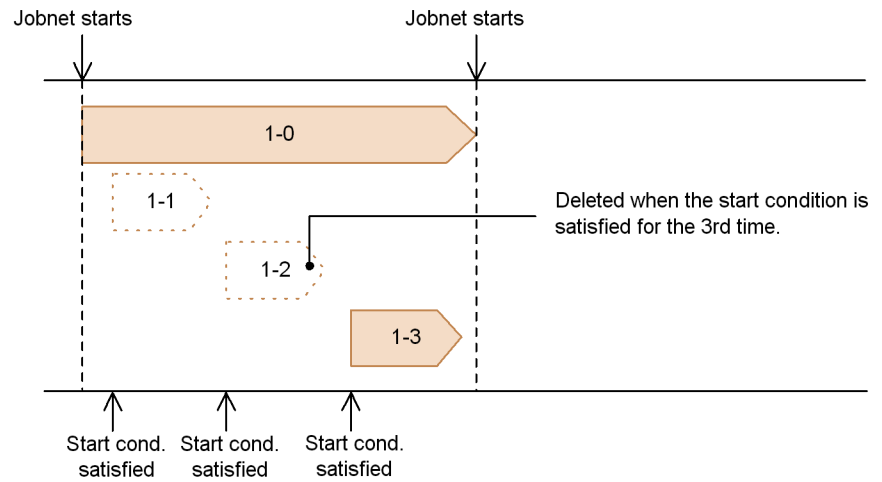
In both cases, if the set number of logs to keep is exceeded, execution generations are deleted one by one, starting from the oldest. The monitoring generation is deleted with the last execution generation. If you have performed an upgrade installation from version 8 or earlier, note that the generation management method changed in version 9. For an example, see (4) *Example of generation management of a jobnet with a start condition (upgrade installation from version 8 or earlier)*.

The following figure shows an example of generation management of a jobnet that has a start condition.

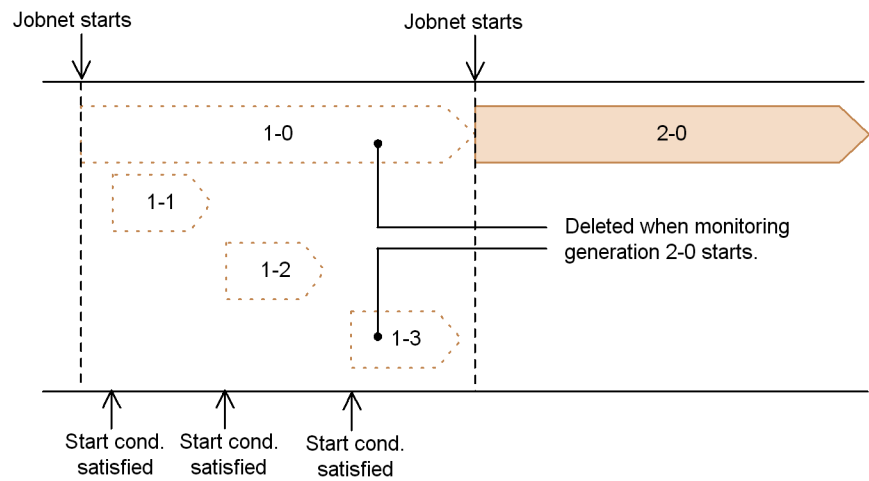
Figure 4-13: Generation management of a jobnet with a start condition

- Number of logs to keep: 2


- When the start condition is satisfied for the 3rd time



- When monitoring generation 2-0 starts



Legend:

 : Generation deleted by jobnet generation management

When a jobnet with a start condition is registered for execution, a monitoring generation is created whenever the start time arrives (generations 1-0 and 2-0 in the figure). At the same time, one or more execution generations in *Wait for start condition* status are created, and are executed when the start condition is satisfied (generations,

1-1, 1-2, and 1-3 in the figure).

When the set number of logs to keep is exceeded, the execution generations triggered by the start condition are deleted one by one, starting from the oldest. Deletion is performed when monitoring for a start condition starts or when a start condition is satisfied. The monitoring generation is deleted with the last execution generation.

In the above example, 2 is set as the number of logs to keep. In jobnet generation 1-0, when the start condition is satisfied for the second time, generation 1-3 is created (and placed in *Wait for start condition* status) and generation 1-1 is deleted. Likewise, generation 1-2 is deleted when the start condition is satisfied a third time. Generation 1-3 is deleted when the start time arrives for the next run of the jobnet, and jobnet generation 2-0 begins. Because generation 1-3 is the last execution generation, generation 1-0 is deleted with it. When specifying the number of logs to keep, bear in mind that all generations, both monitoring and execution generations, are counted and subject to jobnet generation management in this way.

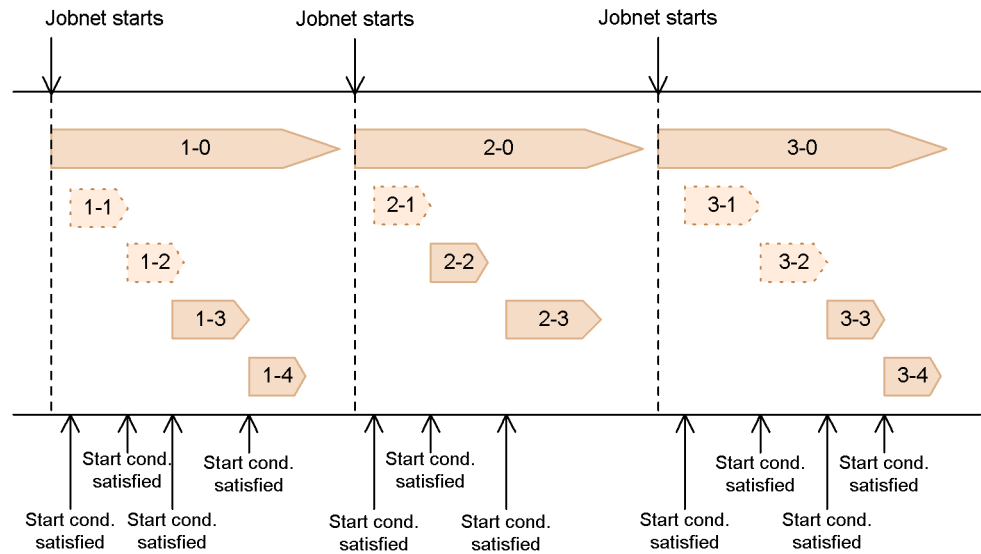
(4) Example of generation management of a jobnet with a start condition (upgrade installation from version 8 or earlier)

If you have performed an upgrade installation from version 8 or earlier, generation management of a jobnet that has a start condition is performed separately for the monitoring generations and the execution generations (generations in *Wait for start condition* status that will be executed when the start condition is satisfied).


The following figure shows an example of generation management of a jobnet that has a start condition.

Figure 4-14: Generation management of a jobnet with a start condition (upgrade installation from version 8 or earlier)

- Number of logs to keep: 2



Legend:

 : Jobnet deleted due to the number of logs to keep being exceeded

When a jobnet with a start condition is registered for execution, a monitoring generation is generated at the set start times (generations 1-0, 2-0, and 3-0 in the above figure). At the same time, the system creates one or more execution generations in *Wait for start condition* status, which are executed when the start condition is satisfied (generations 1-1 to 1-4, 2-1 to 2-3, and 3-1 to 3-4 in the above figure).

Like ordinary jobnet generations, monitoring generations 1-0, 2-0, 3-0 are also subject to generation management based on the number of logs to keep. If a monitoring generation is deleted when the number of logs to keep setting is exceeded, the system also deletes all generations that were triggered by the established start condition and executed while that deleted generation was in *Now monitoring* status. However, when a monitoring generation has completed, even if the number of generations exceeds the number of logs to keep setting, that monitoring generation will not be deleted as long as there is any generation triggered by the start condition and still in *Now running* state.

Management of generations initiated by the start condition is performed when the start condition is satisfied. This applies to all the generations initiated by a start condition

being monitored by one generation in *Now monitoring* status. When the start condition is satisfied, the generation in *Wait for start condition* status is placed in *Now running* status, and a new *Wait for start condition* generation is created to wait for the start condition to be satisfied again. This new generation is added to the total resident generations, and the oldest generation exceeding the number of logs to keep is deleted.

In the above example, **2** is set as the number of logs to keep in the jobnet definition. Therefore, in jobnet generation 1-0, when the start condition is satisfied for the second time, generation 1-3 is created (and placed in *Wait for start condition* status) and generation 1-1 is deleted. When the start condition is satisfied for the third time, 1-4 is created and 1-2 is deleted. The start condition is not satisfied again within the valid range, so the two generations (1-3 and 1-4) are left in generation 1-0. The next generation of the jobnet (2-0) begins running when its start time arrives. Only two monitoring generations (1-0 and 2-0) are now monitoring for the start condition, so generation 1-0 (plus 1-3 and 1-4) still remain. While 2-0 is monitoring, 2-1 is deleted and 2-2 and 2-3 remain. But when generation 3-0 begins running, the number of logs to keep setting is exceeded, so generation 1-0 (plus 1-3 and 1-4) are all deleted at this point.

In this way, when a jobnet has a start condition, not only the monitoring generation but all execution generations executed within that generation are subject to generation management. When setting the number of logs to keep, you should therefore bear in mind how many generations could be initiated by the start condition. Because generation management also applies to generations in *Wait for start condition* status, we recommend that you set a value of at least 2 as the number of logs to keep. A value of only 1 could result in the execution result of the previous generation being immediately deleted.

However, in regard to setting a high number of logs to keep, note the following points:

(5) Notes on setting a large number of logs to keep for a jobnet with a start condition (upgrade installation from version 8 or earlier)

A start condition might be satisfied dozens or even hundreds of times during one monitoring generation. If you set an exceptionally large value as the number of logs to keep, remember that not only the generations initiated by the start condition but also the monitoring generations will be saved. Suppose, for example, that you set 100 as the number of logs to keep for a daily jobnet where the start condition arises 100 times a day. This means that a total of 10,100 generations will be saved (the 100 monitoring generations, plus 100 generations for each of the past 100 days). In other words, the number of saved generations will be more than the square of the value set as the number of logs to keep.

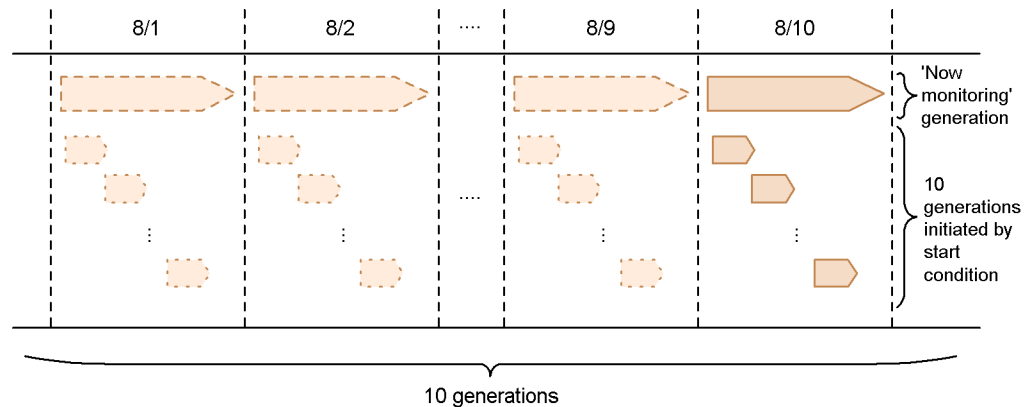
The number of saved records increases in proportion to the value given by the following equation: number of logs to keep \times number of registered units. This has a significant impact on all sorts of operations involving database access, such as canceling jobnet registration. You should therefore consider the effects on system

performance when setting the number of logs to keep. Also, be sure to promptly cancel the registration of start-condition monitoring generations that do not need to be kept, and administer the system so as to prevent huge numbers of records resulting from a large number of kept generations.

The following figure shows an example of canceling registration to prevent mass record generation.

Figure 4-15: Example of canceling registration to prevent mass record generation

Number of logs to keep: 10
Processing cycle: Daily



Legend:

- : No need to keep
- : Must be kept

In this example, the daily jobnet has a start condition that is satisfied 10 times a day. To save each day's execution results, 10 is set as the number of logs to keep. Ten days after registering this jobnet, the number of saved generations will total 110 (*Now monitoring* generations \times 10 days + 10 daily condition-dependent generations \times 10 days). But only one day's execution results actually need to be kept. So, by the time the generations on 8/10 complete execution, there will be 99 generations of execution results, from 8/1 to 8/9, that you do not need. To delete these unnecessary generations, you must cancel registration of the jobnets in the period from 8/1 to 8/9. Canceling these jobnets will reduce the number of saved records to just the 11 generations on 8/10, thereby lessening the impact on database operations.

From these considerations, if you anticipate that the start condition will often arise, and you therefore set a high number of logs to keep, you should make a daily habit of

canceling registration of the previous day's jobnet at completion of the current day's jobnet.

For details on canceling jobnet registration, see *7.3 Releasing the jobnet from registration for execution* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide* and *ajsleave* in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

You can change this generation management method to the method used in version 9. In version 9, records are saved only for the specified number of logs to keep. Consequently, there is no need to cancel registration of monitoring generations to prevent mass record generation. To change the method of managing generations of jobnets that have a start condition, set the value of the `SAVEGENTYPE` environment setting parameter in the `jajs_config` command. For details about this parameter, see *2.2 Setting up the scheduler service environment* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 2*.

When you change from the generation management method used in version 8 and earlier to the method used in version 9, records equivalent to the number of generations saved with the earlier method, minus the number of generations saved with the version 9 method, will be deleted when the scheduler service starts. The time required to delete these records is given by the following equation: *number of registered root jobnets with start conditions* × *specified number of logs to keep* × *specified number of logs to keep*. The time taken will also depend on the hardware you are using. If the time required to delete records is likely to affect operations, delete unwanted generations manually before you change the management method. Deleting records in advance helps to reduce the time taken for the scheduler service to start.

The scheduler service startup time can also be substantially reduced by changing the generation deletion process based on generation management and cancellation of jobnet registration to an asynchronous process. For details, see *6.1.6 Changing the mode in which unregistration or generation management deletes the generations of a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or *14.1.6 Changing the mode in which unregistration or generation management deletes the generations of a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

4.3 Checking execution registration information

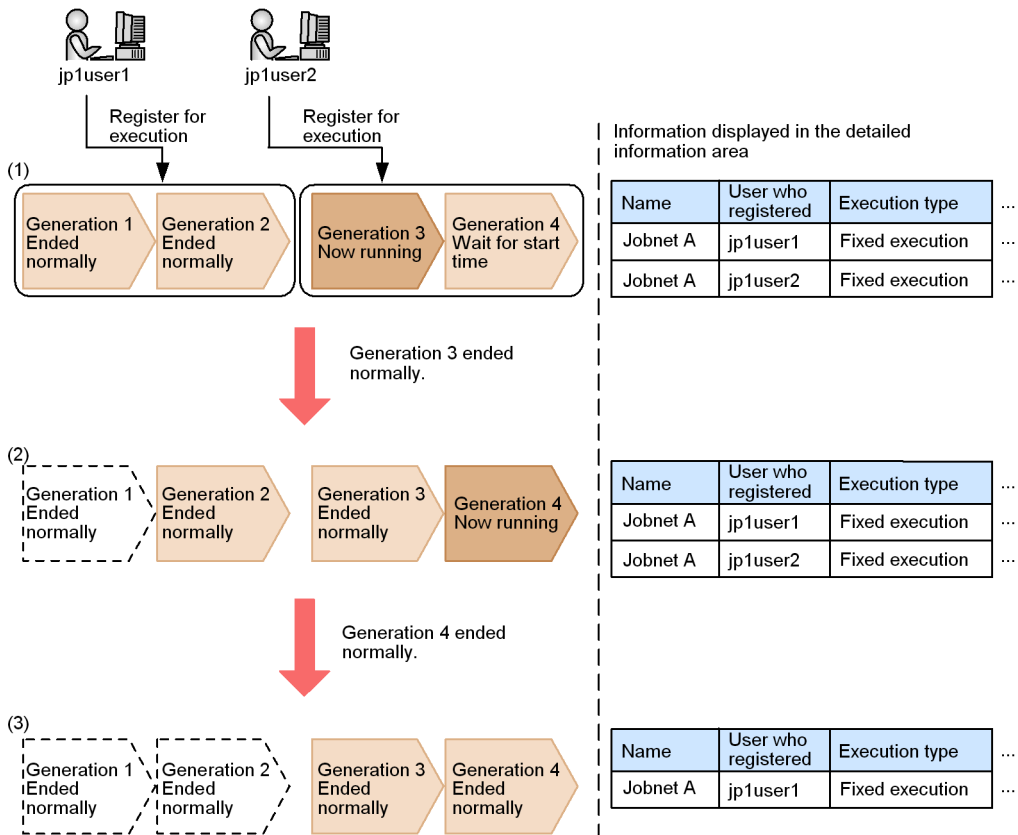
You can check the execution registration information for a jobnet from the JP1/AJS3 - View main window. From the **Function** menu, choose **Register for Execution**. You can then display execution registration information about a selected unit in the list area and detailed information area.

When you select a jobnet in the list area, the settings entered in the Register for Execution dialog box for that jobnet appear in the detailed information area. No information is shown about generations that have been deleted by generation management or by canceling jobnet registration.


The following figure illustrates the relationship between the jobnet generation management process and the execution registration information displayed in the detailed information area.

Figure 4-16: Relationship between jobnet generation management and the information displayed in the detailed information area

● Number of logs to keep: 2



Legend:

 : Generation deleted by the jobnet generation management process

This figure shows the information displayed for jobnet A, which has 2 set as the number of logs to keep. Two users (`jp1user1` and `jp1user2`) have registered the jobnet for execution. Generations 1 to 4 have been created. `jp1user1` registered generations 1 and 2, and `jp1user2` registered generations 3 and 4.

When generation 3 is running, all the generations registered by the two JP1 users are still resident (1). Therefore, all the settings and operations performed by both users are displayed as execution registration information. When generation 3 ends normally,

generation 1 registered by jp1user1 is deleted by the generation management process (2). At this point, generation 2 registered by jp1user1 is still resident, so the settings and operations performed by jp1user1 remain displayed. When generation 4 ends normally, generation 2 is deleted (3). At this point, all the generations registered by jp1user1 have been deleted, so the jp1user1 settings and operations disappear.

If either user cancels registration or stops execution of the jobnet, generation information is erased in the same way as for deletion by the generation management process.

If the same JP1 user registers a jobnet for execution on the same day and at the same time (within one second) from multiple instances of JP1/AJS3 - View, the execution registration information is handled as relating to a single unit and is displayed in one line in the detailed information area.

4.4 Checking execution schedules

This section describes how to check the execution schedule of a registered jobnet, and explains how jobnet schedules are simulated in JP1/AJS3.

4.4.1 Checking a jobnet schedule

You can check the schedule of a jobnet registered for execution in the following JP1/AJS3 - View windows:

- Daily Schedule window (Hierarchical View or All Jobs)
- Monthly Schedule window

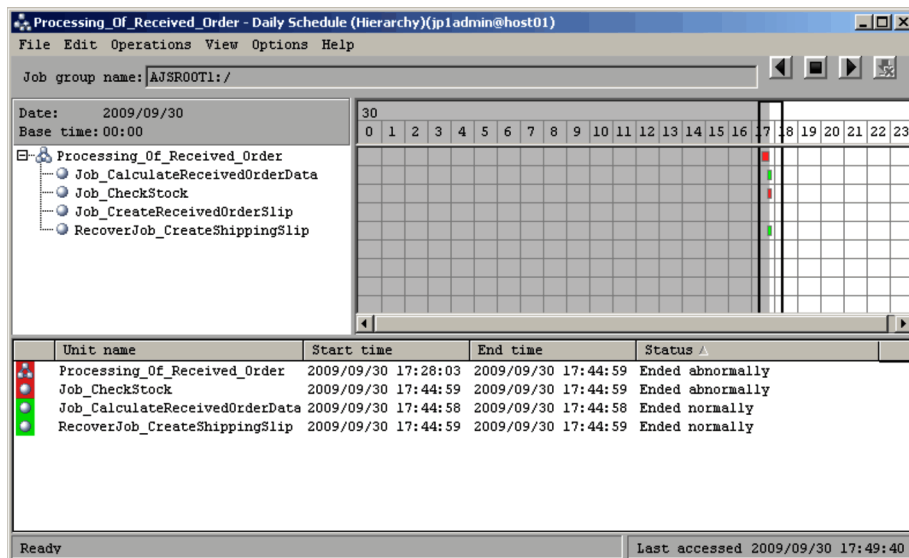
The following provides a brief introduction to checking schedules in these two windows. For details on window operations, see the *Job Management Partner 1/ Automatic Job Management System 3 Operator's Guide*.

(1) Checking schedules in the Daily Schedule window

In the Daily Schedule window, you can check the execution schedules, status, and results of the jobnets and jobs scheduled for a specific day.

The following figure shows the Daily Schedule [Hierarchy] window.

Figure 4-17: Daily Schedule [Hierarchy] window



The Daily Schedule [Hierarchy] window is divided into hours of the day. Here you can check the daily schedules, their execution status and results. In the schedule area, the

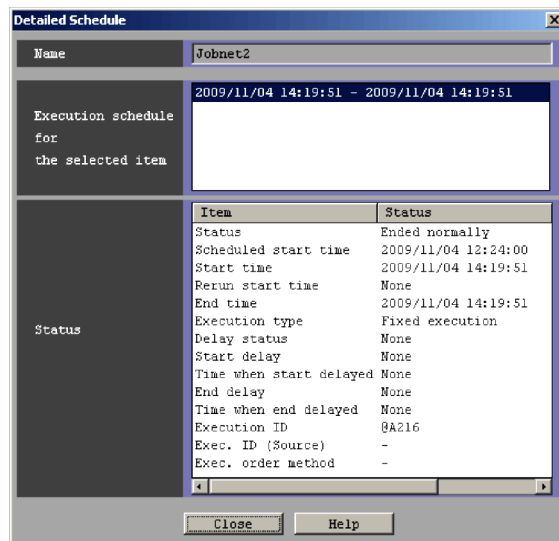
gray part shows past schedules and the white part shows future schedules. When you click a unit schedule in the gray area, the execution result list displays detailed information about that unit's execution results or status. When you click a schedule in the white area, you can view detailed information about that unit's execution schedule. But generations without execution IDs do not appear in the execution result list.

The execution result list shows the start time and end time of each of the listed units. For details on the start and end times appearing here, see 4.4.2(2) *Execution simulation*.

By double-clicking on a schedule in the schedule area, or on a unit in the execution result list, you can view a detailed schedule for that unit.

The following figure shows the Detailed Schedule window.

Figure 4-18: Detailed Schedule dialog box



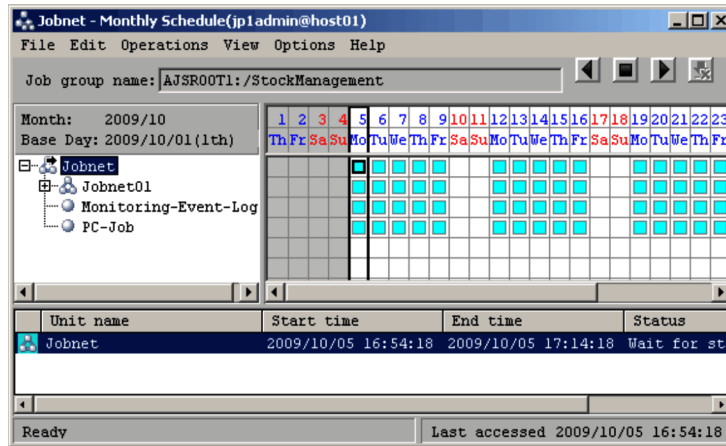
(2) Checking schedules in the Monthly Schedule window

In the Monthly Schedule window, you can check the execution schedules, status, and results of the jobnets and jobs scheduled for a specific month.

A square appearing in the schedule area indicates a *dummy schedule* (an execution schedule that has not been finalized) for a jobnet registered for planned execution or for fixed execution with a specified generation count. Circles indicate the finalized execution schedules for a jobnet registered for period-based fixed execution, or execution schedules corresponding to the generation count specified for generation-based fixed execution.

The following figure shows the Monthly Schedule window.

Figure 4-19: Monthly Schedule window



The Monthly Schedule window shows the date and the day of the week for each day of the month. Here you can check the schedules for each unit during the month, their scheduled times and execution status. In the schedule area, the gray part shows past schedules and the white part shows future schedules. When you click a unit schedule in the gray area, the execution result list displays detailed information about that unit's execution results or status. When you click a schedule in the white area, you can view detailed information about that unit's execution schedule. Generations without execution IDs do not appear in the execution result list.

The execution result list shows the start time and end time of each of the listed units. For details on the start and end times appearing here, see *4.4.2(2) Execution simulation*.

You can view a detailed schedule for a unit by double-clicking a schedule name in the schedule area or a unit name in the execution result list. For the Detailed Schedule dialog box, see *Figure 4-18*.

4.4.2 Simulation

JP1/AJS3 uses two types of simulation:

- Schedule simulation
- Execution simulation

Schedule simulation and execution simulation are used to calculate the information about executed and scheduled generations appearing in the execution result list in the Daily Schedule and Monthly Schedule windows. The budget and result information output by the `ajsshow` command with the `-b`, `-e`, `-v`, or `-w` option specified is also calculated by schedule simulation and execution simulation. For the `ajsshow` command syntax, see *ajsshow* in *2. Commands* in the manual *Job Management*

Partner I/Automatic Job Management System 3 Command Reference 1.

The following describes these two types of simulation.

(1) Schedule simulation

When a jobnet is registered for execution, the finalized schedules are stored in the JP1/AJS3 database.

Jobnet registered for planned execution

Upon registration, only the first schedule is stored in the database. Thereafter, at each jobnet start time, a generation is created and its schedule is stored in the database.

Jobnet registered for fixed execution over a specified period

Only the schedules for generations within the specified period are stored in the database.

Jobnet registered for fixed execution with a specified generation count

Schedules for the specified number of generations are stored in the database. Thereafter, at each execution of the jobnet, the next generation is created and its schedule is stored in the database, so that the number of stored schedules always matches the specified number of generations.

During this process, JP1/AJS3 generates unfinalized schedules if the jobnet is registered for planned execution or for fixed execution with a specified generation count. These unfinalized schedules, or *dummy schedules* as they are known, are simulated from the jobnet schedule definition. Simulation of execution schedules based on a jobnet schedule definition is known as *schedule simulation*.

The following figure shows an example of schedule simulation for a jobnet registered for planned execution.

Figure 4-20: Example of schedule simulation for a jobnet registered for planned execution

	8/7	8/8	8/9	8/10	8/11	8/12	8/13	8/14	8/15	8/16
net1	■	■	■	□	□	□	□	□	□	□
job1	■	■	■	□	□	□	□	□	□	□
job2	■	■	■	□	□	□	□	□	□	□

Legend:

- : Normal end
- : Execute next time
- : Dummy schedule

In this example, the next execution schedule due to start on 8/9 is fixed and stored in the database. The execution schedules from 8/10 onward are dummy schedules generated by schedule simulation.

The following is an example of when the `ajsshow` command is executed at 08:00 on August 9, 2009. The example assumes that `/Net` will be started at 12:00, and has been registered for planned execution.

Example

```
ajsshow -i "start=%BB type=%ii %JJ" -b 2009/8/9 -e 2009/8/11 /Net
```

```
start=Aug 9 2009 12:00 type=planned /Net
start=Aug 10 2009 12:00 type=simulation /Net
start=Aug 11 2009 12:00 type=simulation /Net
```

Cases when dummy schedules cannot be scheduled

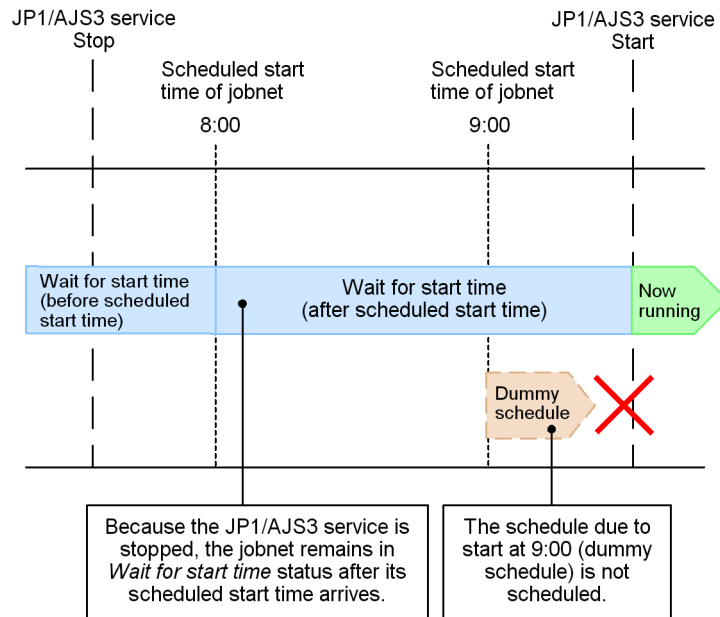
A *dummy schedule* is one that has not been finalized.

Under planned execution, the next schedule to be executed is finalized when the jobnet starts execution. For this reason, if the next schedule remains in *Waiting for start time* status or *Being held* status without being executed, beyond the time at which the dummy schedule was calculated, the dummy schedule cannot be scheduled. This occurs in the following cases:

- When the JP1/AJS3 service stops
- When a job or jobnet is held

The following figure shows an example of when a dummy schedule cannot be scheduled.

Figure 4-21: Example case when a dummy schedule cannot be scheduled



In this example, the jobnet due to execute at 8:00 failed to do so because the JP1/AJS3 service was stopped at that time. It waits for the service to start, and then begins running.

Jobs will enter *Now running* status after service restart only if the JP1/AJS3 service is restarted in either of the following ways:

- Hot start
- Warm start if the jobnet was due to execute before the scheduler service is restarted, and is set to start at the same time as the scheduler service

To specify that a jobnet that was due to execute before the scheduler service starts is to start when the scheduler service starts, specify `exec` (default) in the `OVERSCHEDULE` environment setting parameter. If you specify a value other than `exec`, the jobnet will be placed in *Skipped so not executed* status.

The schedule due to start at 9:00 (dummy schedule) will be finalized if the jobnet starts execution at 8:00. If the jobnet has not started running by 9:00, the dummy schedule will not be scheduled.

For details on dummy schedules, see 3.3.3(2) *Schedule option* and 4.5.3(4) *Changing to a time that extends beyond the dummy schedule*.

(2) Execution simulation

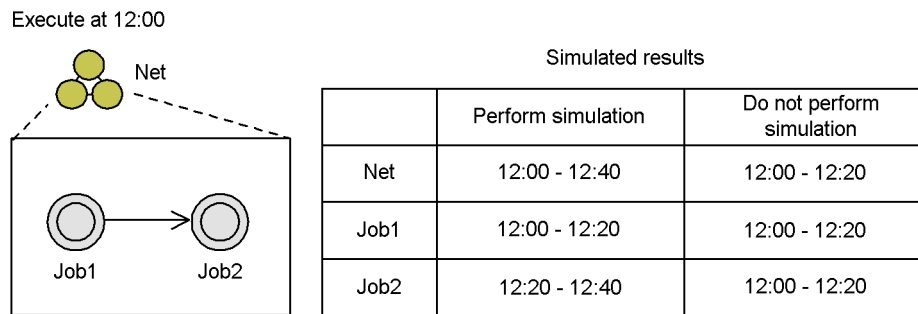
In addition to schedule simulation, JP1/AJS3 calculates jobnet start times and end times from their scheduled start times, the required flow of jobs defined in a jobnet, and other such information. This procedure is known as *execution simulation*.

In execution simulation, JP1/AJS3 simulates the scheduled start and end times of a jobnet and jobs to more closely match actual operation. The simulated times are based on various information, including the current time, scheduled start time of the jobnet, its execution status, past execution time, types of units and unit flow. If the jobnet has never been executed before, the time (in seconds) set in **First execution time** in the Preferences dialog box (or 60 seconds if using the `ajsshow` command) is taken as the required execution time of the first job defined in the jobnet. The scheduled start and end times of the jobnet and its lower units are then calculated from that assumed execution time.

In the Preferences dialog box, you can select whether to perform simulation. For details on settings in the Preferences dialog box, see 15.3.36 *Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Note that for the `ajsshow` command, the settings in the Preference dialog box are not relevant.

When you choose not to perform simulation, the displayed start time is based on the jobnet's schedule rule, and the scheduled end time is calculated taking the setting in **First execution time** in the Preferences dialog box as the required execution time. For the jobnet's lower units, however, the jobnet's start and end times are displayed and simulation is not performed. The following figure illustrates the differences in scheduled start and end times depending on whether simulation is performed.

Figure 4-22: Differences between start and end times depending on whether execution simulation is performed



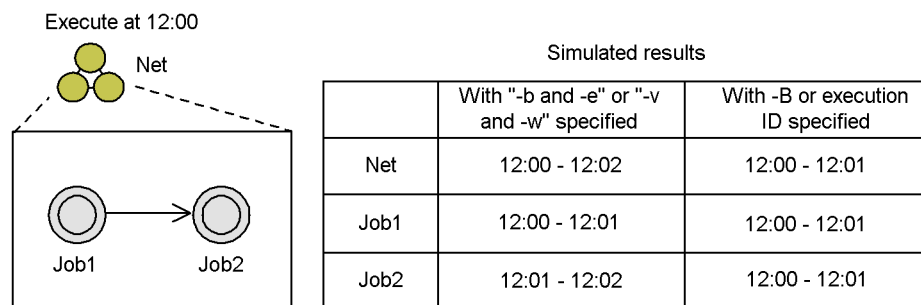
In this example, assume that 1,200 seconds (default) is set in **First execution time** in the Preferences dialog box, and that the jobnet has never been executed since it was registered.

When execution simulation is performed, since the defined job sequence is Job1 followed by Job2, Job1's end time will be Job2's start time. Each unit is simulated using the required execution time (1,200 seconds = 20 minutes) set in **First execution time**. The simulated execution times will therefore be 12:00 to 12:20 for Job1, and 12:20 to 12:40 for Job2. As a result, the execution time for the "Net" jobnet consisting of Job1 and Job2 will be 12:00 to 12:40.

When simulation is not performed, the end time of the "Net" jobnet is calculated from the value set in **First execution time**. Because simulation is not performed for the units defined under this jobnet, the execution time is displayed as 12:00 to 12:20 for Net, Job1, and Job2 alike.

When the `-b` and `-e` options or the `-v` and `-w` options are specified, the `ajsshow` command always performs simulation. When the `-B` option or execution ID is specified, the `ajsshow` command does not perform simulation. The following figure shows the difference in output depending on the specified option.

Figure 4-23: Difference in output depending on the option specified in the `ajsshow` option



In this example, it is assumed that the jobnet has not yet been executed after the jobnet was registered for execution.

When the `-b` and `-e` options or the `-v` and `-w` options are specified in the `ajsshow` command, because the execution of Job1 is followed by the execution of Job2, the end time of Job1 is the start time of Job2. Since the simulation assumes 60 seconds (one minute) as the first execution time of each unit, Job1 is executed from 12:00 to 12:01, and Job2 is executed from 12:01 to 12:02. Therefore, the `Net` jobnet that contains Job1 and Job2 is executed from 12:00 to 12:02.

The following shows usage examples of the `ajsshow` command. These examples assume that the command is executed at 08:00 on August 9, 2009.

Example 1

```
ajsshow -i "start=%BB end=%OO %JJ" -b 2009/8/9 -e 2009/8/9
-R /Net
```

4. Executing an Application

```
start=Aug 9 2009 12:00 end=Aug 9 2009 12:02 /Net
start=Aug 9 2009 12:00 end=Aug 9 2009 12:01 /Net/job1
start=Aug 9 2009 12:01 end=Aug 9 2009 12:02 /Net/job2
```

When the `-B` option or execution ID is specified in the `ajsshow` command, the command calculates the end time, assuming 60 seconds (one minute) as the first execution time for the `Net` jobnet. However, because the command does not perform simulation for the subordinate units, the command displays 12:00 as the start time and 12:01 as the end time for the `Net` jobnet, as well as for both `Job1` and `Job2`.

Example 2

```
ajsshow -i "start=%BB end=%OO %JJ" -B 20090809001 -R /Net

start=Aug 9 2009 12:00 end=Aug 9 2009 12:01 /Net
start=Aug 9 2009 12:00 end=Aug 9 2009 12:01 /Net/job1
start=Aug 9 2009 12:00 end=Aug 9 2009 12:01 /Net/job2
```

Supplementary notes

- If the jobnet enters *Abnormal end* or *Running + Abend* status while execution simulation is being performed, the subsequent units will be shown as *Not scheduled to execute*. When execution simulation is not being performed, however, the subsequent units are shown as *Waiting to execute*.
- Execution simulation is not performed in regard to the specified schedule option (**Schedule skip** or **Multi-schedule**).
- When a jobnet has a start condition, execution simulation is not performed for a generation that will be in *Wait for start condition* status but has not yet been created.
- When a jobnet uses a jobnet connector, execution simulation for the jobnet connector is based on its statistical information. Because statistical information is deleted when a jobnet is unregistered, if you cancel registration of either the root jobnet on the jobnet connector side or the jobnet that the connector is linked to, inconsistencies will arise in subsequent execution simulations.

4.4.3 Statistical information

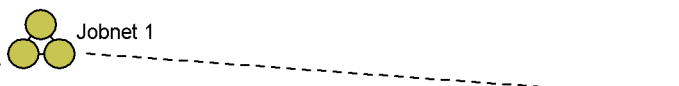
Simulation is performed based on statistical information. For details on execution simulation, see 4.4.2(2) *Execution simulation*.




(1) Statistical information and simulation

The execution time of a unit is calculated by averaging the time of normally terminated executions of the unit. The total normally terminated execution time and the total number of normally terminated executions are stored as statistical information for each root jobnet.

The following figure shows management of statistical information.

Figure 4-24: Management of statistical information



Unit name	Total normally terminated execution time	Total number of normally terminated executions
 Job 1	180 seconds	3
 Job 2	140 seconds	2
 Job 3	210 seconds	7

Statistical information such as that shown above is used to perform simulation.

In this example, the average execution times of Job1, Job2, and Job3 in Jobnet1 are 60 seconds, 70 seconds, and 30 seconds, respectively. When the execution order of these jobs is Job1, Job2, and Job3 and the start time of Job1 is 12:00:00, the start time of Job2 is 12:01:00, and the start time of Job3 is 12:02:10. Execution simulation is performed for each root jobnet.

(2) Deleting statistical information

The statistical information of a root jobnet is stored until registration of the root jobnet is canceled. When registration of the root jobnet is canceled, all of the statistical information for the root jobnet stored up to that point is deleted.

(3) Statistical information and the number of logs to keep

Statistical information and the number of logs to keep do not depend on each other. Statistical information also includes results for generations deleted when the number of logs to keep is exceeded. Statistical information is used for simulation.

For example, assume that you have a job in a root jobnet whose number of logs to keep is 1, and the job is executed three times. In this case, if the execution time of the executions are as shown below, the average execution time is 20 seconds. Simulation uses 20 seconds as the execution time.

1st execution: 10 seconds

2nd execution: 20 seconds

3rd execution: 30 seconds

4. Executing an Application

For details on the number of logs to keep, see *4.2.3 Jobnet generation management*.

4.5 Operations on registered jobnets

You can carry out a variety of operations on jobnets that have been registered for execution. Possible operations include changing the schedule of the jobnet, and interrupting or killing a running jobnet or job.

The following table shows the processing you can perform on jobnets and jobs that have been registered for execution.

Table 4-4: Processes available for jobnets and jobs that have been registered for execution

No.	Process	Content
1	Cancel jobnet registration	Cancels registration of a root jobnet for execution. When registration is cancelled, all results from previous executions are also deleted.
2	Add scheduled execution times and dates	When the schedule of a root jobnet whose status is Being held or Waiting for start time is pushed ahead, this process adds the next scheduled execution. When a root jobnet is registered for fixed execution, this process adds a schedule with a specific time and date.
3	Temporarily change execution time	Temporarily changes the scheduled execution start time of a unit. ^{#1}
4	Temporary immediate execution	Temporarily changes the scheduled execution start time of a unit to a scheduled execution start time that allows the unit to execute immediately. ^{#1} <ul style="list-style-type: none"> • With root jobnets: the scheduled execution start time is changed to the current time. • With nested jobnets: the scheduled execution start time is changed to the base time of the execution date.
5	Prohibit execution	Cancels the next scheduled execution, or the execution in a fixed schedule.
6	Temporarily change hold attribute	Temporarily holds or releases the next scheduled execution of a jobnet or its execution within a fixed schedule. Specify the hold attribute to rerun a job or jobnet after specific operations such as changing the execution schedule. ^{#2}
7	Temporarily change delay monitor	Temporarily changes the delay monitor time (in hours) of a jobnet that is waiting to execute.
8	Temporarily change jobnet priority	Temporarily changes the execution priority of a root jobnet that is waiting to execute.

No.	Process	Content
9	Interrupt jobnet	Interrupts the root jobnet when the current job ends. Nested jobnets under the root jobnet are also interrupted. Note that the interrupted jobnets are treated as having ended abnormally. If you interrupt a jobnet that is monitoring for a start condition, monitoring for a start condition is canceled.
10	Kill	Kills the unit being executed. ^{#3} When you kill the root jobnet, all running jobs are killed, and the jobnet is interrupted. If you kill a root jobnet that is monitoring for a start condition, monitoring for a start condition is canceled.
11	Rerun	Reruns a jobnet or job. The rerun is not counted in the number of generations (execution count).
12	Change job status	Changes the status and return value of a job. For example, if an error occurs in the host and the status that is reported to JP1/AJS3 is not the actual status of the job, this operation changes the status to the actual status.
13	Release jobnet	Automatically switch the definition of an active jobnet at a specified date and time.
14	Suspend root jobnet	Temporarily suspends execution of units for all generations under the root jobnet, and allows you to change definitions that are registered for execution.

#1

You cannot change the execution start time if the job or jobnet you specify is already running. You cannot change the execution start time if the root jobnet has already finished execution or is being rerun.

#2

If you restart JP1/AJS3 - Manager before the jobnet or job whose hold attribute you temporarily changed has finished, the status of the job or jobnet remains **Being held**.

#3

When processing in a Unix job successfully executes the `setpgrp` (set process group) system call, the processes in the process group in question cannot be canceled. To terminate processes in a process group created in this manner, using other means such as the `kill` command.

4.5.1 Canceling registration of a jobnet

You can cancel registration of a jobnet that is registered for execution.

You can only cancel registration of a jobnet that is not running at the time. If you want to cancel registration of a jobnet that is currently running, you can either wait for the jobnet to terminate, or kill the jobnet and then cancel it.

When you cancel a jobnet registration, you can choose to cancel all generations of the jobnet, or only those generations within a certain period. With the `ajsleave` command, you can also cancel registration of a specific generation by specifying the registration number.

When you cancel registration of a root jobnet, all nested jobnets are also canceled. You cannot cancel registration of a nested jobnet by itself.

If you cancel the registration of all jobnets, the execution results are erased up to the previous generation.

Special note for planning groups

You can specify a cancellation period for canceling the registration of jobnets in a planning group. You can then add new jobnets for execution during that period.

Cautionary note

The time that it takes to cancel registration increases in proportion to the number of units (including jobnets, lower-level nested jobnets and jobs) to be canceled, and the number of logs to keep. If the number of units multiplied by the number of logs to keep is in the tens of thousands, then cancellation will take more than 30 minutes. This can place a high I/O load on the scheduler service over a prolonged period, and could affect the execution of other jobnets. In order to minimize the load on the scheduler service, we recommend that you try either of the following methods when you need to cancel a very large jobnet.

- Specify an execution ID or execution registration number in the `ajsleave` command, and delete past generations one by one.
- Avoid peak operation times.

Supplementary note

When a large number of jobnets are registered for immediate execution, and these jobnets remain registered, system performance may be adversely affected. We recommend that you periodically cancel the registration of jobnets when they terminate if you do not need to refer to the results.

You can use either the GUI or the `ajsleave` command to cancel jobnet registration. For details on using the GUI, see *7.3 Releasing the jobnet from registration for execution* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajsleave` command, see the description of `ajsleave` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

4.5.2 Adding an execution schedule to a jobnet

You can add an execution schedule to a root jobnet, specifying an execution start date and time. This is possible only for a jobnet that is registered for fixed execution or not scheduled to execute.

Also, you can add an execution schedule to a nested jobnet that is subordinate to a root jobnet that has already been scheduled for execution.

(1) Adding an execution schedule to a root jobnet

When you add an execution schedule, it is added as a fixed execution schedule. Any schedule definitions that were specified for the nested jobnets or the root jobnets are ignored from this point. Also, start conditions for jobnets are not monitored, and jobnets with start conditions are executed immediately. When there are nested jobnets defined under the root jobnet to which you have added the execution schedule, the nested jobnets inherit the same scheduled execution time as the root jobnet.

You can perform this operation using either the GUI or the `ajsentry` command. For details on using the GUI, see *9.2 Adding an execution schedule for a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajsentry` command, see the description of `ajsentry` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(2) Adding an execution schedule to a nested jobnet

You can add an execution schedule to a nested jobnet that is subordinate to a root jobnet that has already been scheduled for execution. However, you cannot add an execution schedule to a nested jobnet that has already been scheduled for execution. The units in the nested jobnet to which an execution schedule is added are scheduled as described below.

- Jobnet that has been registered for fixed execution

Because the jobnet has already been scheduled, the execution schedule is added only to the nested jobnet additionally specified.

- Jobnet that has been registered for planned execution when the schedule depends on the upper-level unit

The execution schedule is also added to the units that are subordinate to the nested jobnet additionally specified.

- Jobnet that has been registered for planned execution when the schedule does not depend on the upper-level unit

The schedule is recalculated and re-created.

You can add an execution schedule to a nested jobnet from the GUI or by the `ajsp1an` command. For details on the GUI operations, see *9.2 Adding an execution schedule for*

a jobnet in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. For details on the command operations, see the *ajsplan* in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

4.5.3 Temporarily changing the execution start time of a jobnet

You can change the execution start time of a jobnet generation that is scheduled for execution. Since this change is temporary, only the execution schedule you select is changed. All other execution schedules and schedule rules remain unchanged. The normal schedule applies once the jobnet has been executed at the new time.

If you want to move the execution schedule of a root jobnet forward, you can leave the original execution schedule unchanged and establish a new one. In this case, the jobnet is executed at both the original time and the new time.

If you change the day on which the jobnet is next scheduled for execution to the current day, and the scheduled execution time has already passed, the jobnet starts immediately.

You cannot change the execution time of a jobnet that does not have an execution schedule.

Once you have temporarily changed the execution time of a jobnet, you can also cancel the changes and return the jobnet to its original execution time. However, you cannot cancel the changes if you have pushed ahead an execution schedule that is scheduled for fixed execution, and then added a new execution schedule. Before you can undo the changes, you must cancel the execution schedule that you added.

You can perform this operation using either the GUI, or the *ajsplan* command. For details on using the GUI, see 9.3 *Temporarily changing and restoring the execution start time of a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the *ajsplan* command, see the description of *ajsplan* in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(1) Changing the execution schedule for a nested jobnet

When you change the execution time of a nested jobnet, the new time must fall within 48 hours of the execution start date of the root jobnet. If you specify the following day's date when you change the execution time, the new time will still be valid, provided that it is no more than 48 hours from the execution start date of the root jobnet.

Assume for example that you specify an execution start time of 1:00 on 8/12 for a nested jobnet, when the execution start date of the root jobnet is 8/11. In this case, the time for the nested jobnet is treated as if it falls on 8/11, and the nested jobnet is executed.

(2) When the root jobnet uses the 48-hour schedule

Imagine you are using the 48-hour schedule to define the schedule of the root jobnet, and want to temporarily change the execution time of the root jobnet. You must take note of the relationship between the execution start time you are changing and the execution date. For example, imagine you temporarily change the execution start time to either 36:00 on August 11, or 12:00 on August 12. Although the execution start time is actually the same for both settings, JP1/AJS3 considers the execution dates to be different. Therefore, depending on what execution date you specify, you may find that the nested jobnet does not execute.

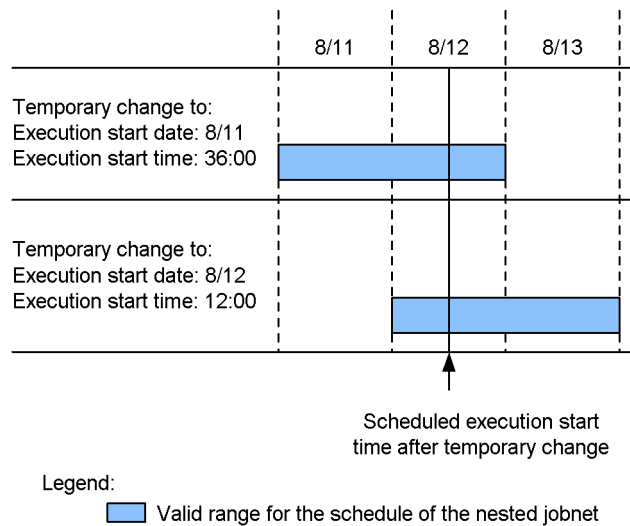
If you temporarily change the execution start time to 36:00 on August 11, the execution date becomes August 11. This means that you can define a schedule between 0:00 and 47:59 on August 11 for a nested jobnet.

If you temporarily change the execution start time to 12:00 on August 12, the execution date becomes August 12. This means that you can define a schedule between 0:00 and 47:59 on August 12 for a nested jobnet.

For details about using the 48-hour schedule to define schedules for root jobnets, see *3.3.1 Time formats and schedule rules for root jobnets*.

The following figure shows the relationship between the specified start time and execution date.

Figure 4-25: Relationship between specified start time and execution date



Cautionary note

- When the root jobnet uses a 48-hour schedule, you must specify an absolute time to temporarily change the execution start time. If you specify the change

by relative time specification, the time from 24:00 onward is considered to be the next day. For example, if a jobnet is scheduled for 22:00 on 8/1, and you change the execution time to 12 hours later by relative time specification, the jobnet will be executed not at 34:00 on 8/1 but at 10:00 on 8/2.

- When the root jobnet is defined using a 48-hour schedule and a time other than 0:00 is set as the base time, specified dates and times will be different from those scheduled in JP1/AJS3. This will complicate operations and care is required. For details, see *3.3.2(1)(a) Setting the execution start time*.

(3) Recalculating the schedules of nested jobnets when you temporarily change the execution schedules of upper-level jobnets

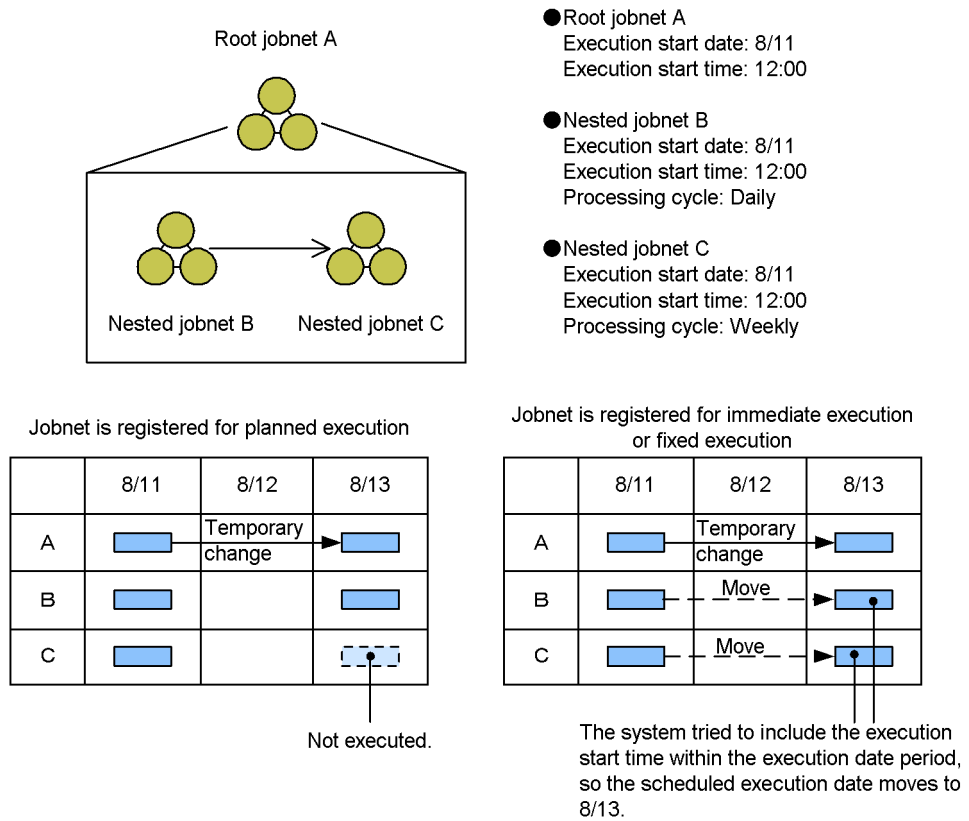
When you temporarily change the execution schedule of a jobnet under which nested jobnets are defined, you can pass the changes on to all lower jobnets. If you shift the start time of the upper-level jobnet by two hours, then provided you have set JP1/AJS3 to pass the changes on, the start times of the lower jobnets are also shifted by two hours. If you choose not to pass on the changes, the start times of lower-level jobnets remain unchanged when you change the start time of the upper-level jobnet.

Even if you choose not to pass schedule changes on to lower jobnets, if you change the scheduled execution date of the root jobnet to a different day, the schedules for the nested jobnets are recalculated automatically. The schedules are recalculated differently depending on the method used to register the root jobnet for execution. The effect of changing the schedule is described below for each of the different methods you can use to register the root jobnet for execution.

If you do not want to consider the different types of schedule recalculation, we recommend that you specify that changes are passed on to lower jobnets when you change the execution time of the root jobnet.

The following figure shows how nested jobnet schedules are recalculated when the execution date of the root jobnet is temporarily changed.

Figure 4-26: Schedule recalculation for nested jobnets on temporary change in the execution date of the root jobnet



When the root jobnet is registered for planned execution

When you temporarily change the execution date of the root jobnet, the schedules of the nested jobnets are recalculated based on the schedule rules. If the execution date of a nested jobnet does not comply with the schedule rules, an execution schedule is not generated for that nested jobnet. Since the processing cycle of nested jobnet B has it executing daily, the schedule is valid for August 13. However, since the processing cycle of nested jobnet C has it executing weekly, the schedule is not valid for August 13. Jobnet C is therefore not executed on August 13.

When the root jobnet is registered for immediate execution or fixed execution

In the figure, the scheduled execution dates of nested jobnets B and C are fixed for August 11. Normally, a fixed execution schedule is not changed. However, if

you change the execution date of the root jobnet, the execution times of the nested jobnets are changed to conform to the new execution date of the root jobnet. The execution times of the nested jobnets are changed so as to fit within 48 hours of the base time on the execution date of the root jobnet. As a result, the execution schedule for nested jobnets B and C moves to August 13.

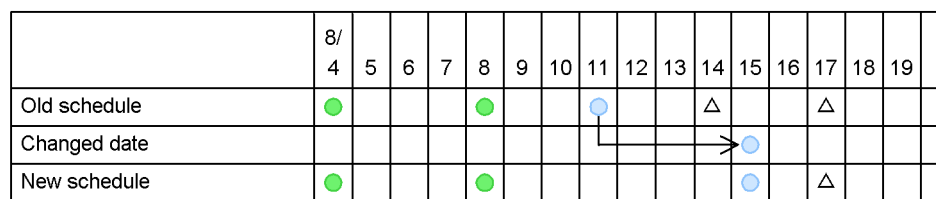
(4) Changing to a time that extends beyond the dummy schedule

A dummy *schedule* refers to a tentative execution schedule for a jobnet that is either registered for planned execution, or registered for fixed execution with a specific number of future generations. A dummy schedule for a jobnet registered for planned execution shows a schedule for succeeding executions of the jobnet. A dummy schedule for a jobnet registered for fixed execution with a specific number of future generations shows a schedule for the jobnet after the specific number of future generations.

Imagine you change the start time of a jobnet registered for planned execution, or a jobnet registered for fixed execution with a specific number of future generations, to a time that is beyond the dummy schedule. In this case, the dummy schedule is not executed.

The following figure shows an example of changing the scheduled execution date of a jobnet to a date beyond the dummy schedule.

Figure 4-27: Changing the scheduled execution date to a date beyond the dummy schedule



Legend:

- : Ended normally
- : Waiting for start time
- △ : Pseudo-schedule

In the above example, the scheduled execution date is changed from August 11 to August 15. The dummy schedule for August 14, which now falls before the new execution date, is not executed. As a result, the scheduled execution dates become August 15 and August 17.

4.5.4 Executing a registered jobnet immediately

Immediate execution generates an execution schedule defined for the current time and runs the registered jobnet immediately. This operation is available for root jobnets

registered for planned execution, and for jobnets that are registered for fixed execution and are scheduled for the next execution.

When you immediately execute a root jobnet that is scheduled for the next execution, you can choose to either leave the next execution schedule intact (**Also in next schedule**), or cancel it (**Not in next schedule**).

When you try to immediately execute a nested jobnet, if the upper-level jobnet does not have an execution schedule, the nested jobnet is not executed. The nested jobnet is not executed until the upper-level jobnet is executed.

When a root jobnet is executed immediately, whether the lower-level nested jobnets are executed depends on the method of execution specified when the root jobnet was registered, as follows:

Immediate execution of a root jobnet registered for planned execution

Execution schedules are generated for the nested jobnets.

Immediate execution of a root jobnet registered for fixed execution

Execution schedules are not generated because with the fixed execution method the schedule is preset.

The scheduled start time of the nested jobnets is adjusted so that they execute within 48 hours of the base time on the day that the root jobnet is scheduled to run.

If immediate execution is performed for a jobnet linked with lower-level jobnets[#], the start time of each jobnet that belongs to the selected jobnet also changes accordingly.

#

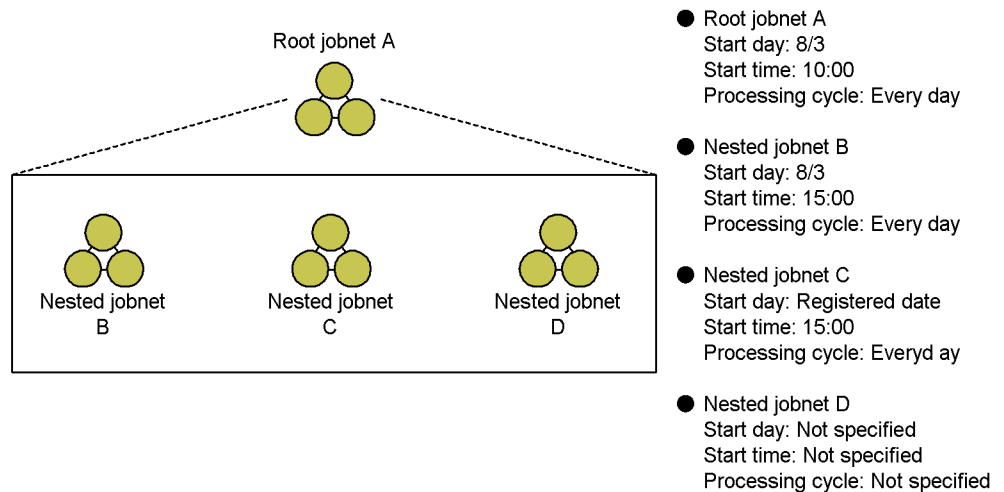
For JP1/AJS3 - View, this applies when the **Change date and time of all lower jobnets** check box is selected, and immediate execution is performed by selecting **Execute Immediately** from **Change Plan**. For a command, this applies when immediate execution is performed by executing the `ajsp1an` command with the `-i` option specified.

Cautionary note

An immediately executed root jobnet runs at the point in time that immediate execution is performed, even if it was defined using a 48-hour schedule. For example, if a jobnet due to start at 34:00 on August 1 is executed immediately at 8:00 on August 2, it will be executed as of 8:00 on August 2, not as of 32:00 on August 1.

The following figure shows an example of a jobnet definition. The example is used to explain operation of immediate execution performed for a jobnet registered for planned execution or fixed execution when the current time is 12:00 on August 1.

Figure 4-28: Example of a jobnet definition



When the jobnet is registered for planned execution (base time: 0:00)

	8/1	8/2	8/3
Root jobnet A			<input type="checkbox"/>
Nested jobnet B			<input type="checkbox"/>
Nested jobnet C			<input type="checkbox"/>
Nested jobnet D			

Legend:
: Registration for planned execution

When the jobnet is registered for fixed execution (base time: 0:00)

	8/1	8/2	8/3
Root jobnet A			<input type="radio"/>
Nested jobnet B			<input type="radio"/>
Nested jobnet C			<input type="radio"/>
Nested jobnet D			

Legend:
: Registration for fixed execution

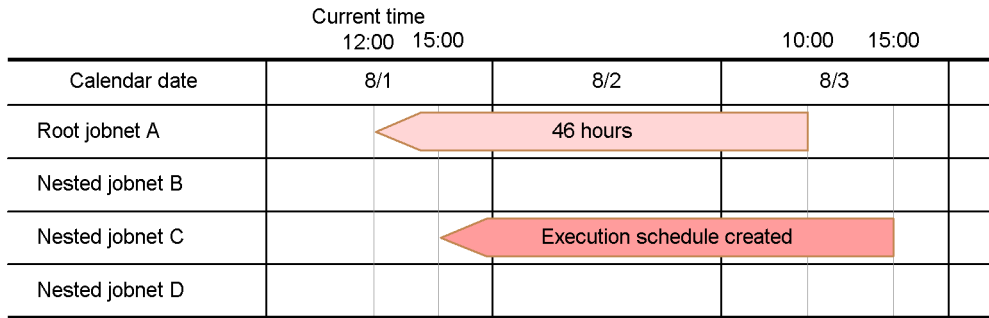
■ When the root jobnet is executed immediately

When the root jobnet is executed immediately, changes to the execution schedules of the lower-level nested jobnets depend on the method of execution specified when the root jobnet was registered, as follows:

- Root jobnet registered for planned execution

The following figure shows how the time changes if the root jobnet registered for planned execution is executed immediately.

Figure 4-29: When the root jobnet registered for planned execution is executed immediately



For root jobnet A, a schedule for execution at 12:00 on August 1 (8/1) is created. Root jobnet A is executed immediately based on this schedule.

Nested jobnet B is not executed because it has not been scheduled for execution on August 1 (8/1).

For nested jobnet C, the start day for the schedule definition is the registered day and overlaps the execution date of root jobnet A that is executed immediately. Therefore, a schedule for execution at 15:00 on August 1 (8/1) is created for nested jobnet C, and nested jobnet C is executed based on this schedule.

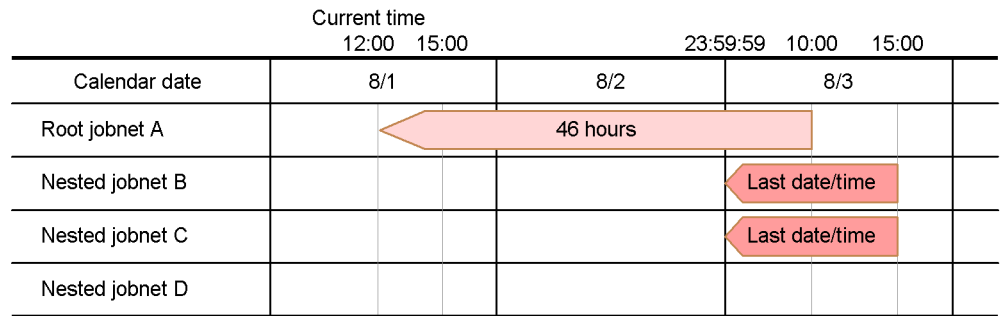
Nested jobnet D is not executed because it has not been scheduled for execution on August 1 (8/1).

If you want to execute nested jobnets B and D, select nested jobnets B and D, and execute them immediately before immediately executing the root jobnet.

■ Root jobnet registered for fixed execution

The following figure shows how the time changes if the root jobnet registered for fixed execution is executed immediately.

Figure 4-30: When the root jobnet registered for fixed execution is executed immediately



For root jobnet A, a schedule for execution at 12:00 on August 1 (8/1) is created. Root jobnet A is executed immediately based on this schedule.

For nested jobnets B and C, no execution schedule is created. However, because the time adjustment for root jobnet A is within 48 hours of the base time, execution of these nested jobnets is rescheduled for the last date and time (23:59:59 on August 2 (8/2)).

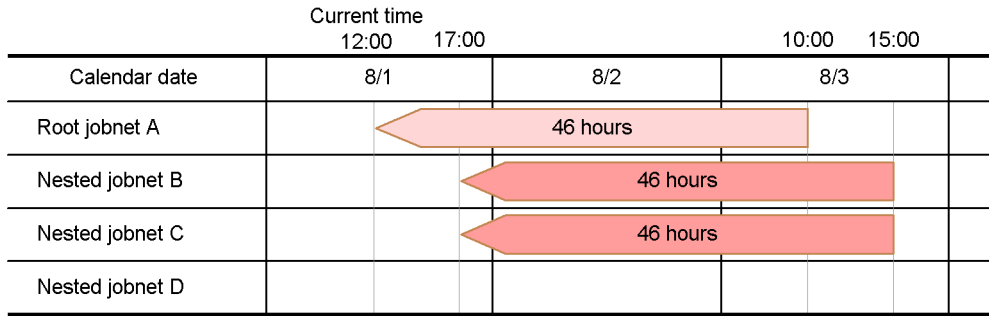
Nested jobnet D is not executed because it has not been scheduled for execution on August 1 (8/1).

If you want to execute nested jobnet D, select nested jobnet D, and execute it immediately before immediately executing the root jobnet.

■ When a root jobnet linked with lower-level jobnets is executed immediately

The following figure shows how the time changes if the root jobnet registered for planned execution or fixed execution is immediately executed with lower-level jobnets linked.

Figure 4-31: When a root jobnet linked with lower-level jobnets is executed immediately



For root jobnet A, a schedule for execution at 12:00 on August 1 (8/1) is created. Root jobnet A is executed immediately based on this schedule.

The start times of nested jobnets B and C are moved back in proportion to the time difference between the start time of root jobnet A and the current time. In the above figure, the execution time of root jobnet A changes by 46 hours. Accordingly, execution of jobnets B and C is rescheduled for 17:00 on August 1 (8/1), which is 46 hours earlier than the time they were originally scheduled.

Nested jobnet D is not executed because it has not been scheduled for execution on August 1 (8/1).

If you want to execute nested jobnet D, select nested jobnet D, and execute it immediately before immediately executing the root jobnet.

You can perform this operation using either the GUI, or the `ajsp1an` command. For details on using the GUI, see *9.4 Immediately executing the execution schedule of a jobnet* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajsp1an` command, see the description of `ajsp1an` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

4.5.5 Prohibiting execution of a job or jobnet

This operation cancels the next execution schedule or fixed schedule of a job or jobnet.

If you prohibit execution of a jobnet that is registered for planned execution, you can then undo the changes, returning the jobnet to its original schedule. However, if you prohibit execution of a jobnet that is registered for fixed execution, you cannot undo the changes.

When a nested jobnet or job that you have prohibited satisfies the execution conditions, its status level changes from **Not sched. to exe.** to **Bypassed**. Succeeding jobnets or jobs are then executed, skipping the bypassed jobnet or job.

You can also use this operation to prohibit the execution of lower-level jobnets in a planning group. After you prohibit execution of a jobnet, you can add new jobnet definitions for the period that becomes available.

When you prohibit the next execution schedule of a root jobnet that is registered for planned execution, the execution schedule featured next is brought forward, and becomes the next execution schedule.

For a root jobnet, you can prohibit a succession of execution schedules. For a nested jobnet or job defined under a root jobnet that is registered for planned execution, execution is canceled only for the execution schedule corresponding with the next execution schedule of the root jobnet.

When you prohibit the execution of a root jobnet, information[#] on the temporary changes made to the root jobnet or nested jobnets is inherited by the following generation. Schedules for nested jobnets are then recalculated according to the next generation of the root jobnet.

You can perform this operation using either the GUI, or the `ajsp1an` command. For details on using the GUI, see *9.5 Temporarily canceling the execution of jobnets and jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajsp1an` command, see the description of `ajsp1an` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

#

If execution is prohibited, information created by the following temporary change operations is passed to the jobnet generation that is to be executed the next time:

Root jobnet:

- Temporarily change hold attribute
- Temporarily change delay monitor
- Temporary change jobnet priority

Nested jobnet:

- Prohibit execution
- Temporarily change execution time
- Temporarily change hold attribute
- Temporarily change delay monitor
- Temporary change jobnet priority

Job:

- Prohibit execution

- Temporarily change hold attribute

4.5.6 Temporarily changing the hold attribute of a job or jobnet

You can temporarily suspend a registered jobnet or job, or temporarily release a suspended jobnet or job. However, if the job or jobnet is already running, you cannot place it in held status.

If you temporarily release a jobnet or job that has a hold attribute, the hold attribute is set again when the jobnet or job terminates.

If a jobnet completes before execution of a jobnet or job that has been temporarily suspended or released, the hold attribute is inherited, and takes effect when the next generation is executed.

A jobnet or job whose hold attribute has been temporarily changed might terminate without starting if the preceding unit ends abnormally. In this case, if you attempt to re-execute the jobnet or job, the jobnet or job is re-executed with the changes applied.

You can temporarily change the hold attribute using either the GUI or the `ajsplan` command. For details on using the GUI, see *9.6 Holding the execution of jobnets and jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the command operation, see the description of `ajsplan` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

Supplementary note

For a root jobnet, the hold setting can be applied in all circumstances, or only if the preceding unit ends abnormally or ends with a warning. In the latter case, the held status can be temporarily released, or it can be released when the termination status of the preceding unit changes to normal termination. The system performs the release operation when recalculating the schedule after you change the schedule definition or calendar information. If you want to retain the root jobnet's held status after the termination status of the preceding unit changes, perform the hold setting again.

4.5.7 Temporarily changing the delay monitor settings of a jobnet

You can temporarily change the delay monitor settings for a jobnet.

However, you cannot temporarily change the start delay time of a jobnet that is already running. If a jobnet is already running, you cannot change the start delay and end delay times and the time required for execution for the jobnet at the same time. Temporarily changing the delay monitor settings for a jobnet that has already terminated has no effect.

If you temporarily change the end delay time or the time required for execution of a jobnet that is already running, the decision on whether to monitor delays is based on whether a delay was detected before the change.

- If no delay was detected before the temporary change:
Detect the end delay when the temporarily changed delay time is reached.
- If a delay was detected before the temporary change:
Do not detect the end delay, even on reaching the temporarily changed delay time.

You can select any of the following monitoring methods to monitor for a delayed start or delayed end.

- **Absolute time**
- **Relative from start of root jobnet**
- **Relative from start of upper-l jobnet**
- **Relative from start of the jobnet**
- **Do not monitor**

With absolute time, specify a value between 00:00 and 47:59. With relative time, specify a value between 0 and 2,879 minutes. Either will result in the application of a delay time that falls within one day of the start day of the specified jobnet.

You can also monitor delay based on the time required for execution of the jobnet. You can set the time required for execution to a value between 1 and 2,879 minutes. With this monitoring method, the delay time is applied regardless of whether it falls within one day of the specified start day for the jobnet.

You can perform this operation using either the GUI, or the `ajsplan` command. For details on using the GUI, see *9.7 Temporarily changing the delay monitor date and time of jobnets* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajsplan` command, see the description of `ajsplan` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

4.5.8 Temporarily changing the priority of a jobnet

You can temporarily change the priority of a jobnet that is registered for execution. This applies only to lower-level jobs without a specified execution priority. Jobs that have a specified priority are unaffected.

You cannot change the priority of a job that is already running.

The temporarily changed priority is invalid once the job completes, except if the job is rerun.

Execution priority is expressed as a number from 1 to 5, where 1 is the lowest priority, and 5 is the highest. For Unix jobs, you can only specify a priority of 4 or 5 only if the OS user who executes the job has superuser privileges.

Jobs or jobnets without a specific execution priority are executed with a priority of 1.

The following table shows execution priority values and their meanings.

Table 4-5: Execution priority values and their meanings

Value	Priority in Windows	Priority in UNIX
1	Lower than interactive processing	JP1/AJS3 service nice value + 20
2		JP1/AJS3 service nice value +10
3	Same as interactive processing	JP1/AJS3 service nice value
4	Higher than interactive processing	JP1/AJS3 service nice value -10
5		JP1/AJS3 service nice value -20

Note

In Windows, three priority levels are available. You can set the following three priority classes to start job processes:

- If you set 1 or 2 as the execution priority value for a job, the job is executed when the system is idle (`IDLE_PRIORITY_CLASS`, which is defined in Windows, is set).
- If you set 3 as the execution priority value for a job, the job is executed as a general process (`NORMAL_PRIORITY_CLASS`, which is defined in Windows, is set).
- If you set 4 or 5 as the execution priority value for a job, the job is executed before the threads of the processes assigned the above priority classes (`HIGH_PRIORITY_CLASS`, which is defined in Windows, is set).

In UNIX, JP1/AJS3 service's nice value used when `ja js_spm` command is executed is used as the default (base) nice value. When the JP1/AJS3 service's nice value has not been set, 20 is assumed as the nice value.

For example, when the priority setting is 1 and the nice value is 20, the priority value is calculated as follows:

$$39 \approx 20 \text{ (default)} + 20 \text{ (increment)}$$

If the resulting value is not within the possible nice value range (0-39), the maximum value is 39, and the minimum value is 0.

If **Queueless Agent** is selected as the execution service, you can set only 39, 30, 20, 10, or 0 as a nice value, according to the job execution priority; 0 corresponds to the highest execution priority. When you change the nice value, you must specify the job execution priority corresponding to the nice value.

You can perform this operation using either the GUI, or the `ajsplan` command. For details on using the GUI, see *9.8 Temporarily changing the execution priority of jobnets* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajsplan` command, see the description of `ajsplan` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

4.5.9 Interrupting a jobnet

You can interrupt a running root jobnet. Once a jobnet is interrupted, no more jobs in the jobnet can start. The jobnet is interrupted once all currently running jobs have completed. The interrupted jobnet is then treated as if it ended abnormally.

You can rerun an interrupted jobnet using either of two rerun methods: **From abnormally ended job** or **From after abnormally ended job**.

Interrupting the root jobnet results in the interruption of all jobnets including nested jobnets. You cannot interrupt a specific nested jobnet.

You can perform this operation using either the GUI, or the `ajsintrpt` command. For details on using the GUI, see *9.9 Interrupting a jobnet being executed* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajsintrpt` command, see the description of `ajsintrpt` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

4.5.10 Killing a job or jobnet

You can kill a running job or jobnet. When you kill a running job, the processes in the running job are forcibly terminated. When you kill a jobnet, no new jobs are allowed to start in the jobnet, and all running jobs are forcibly terminated.

You can perform this operation using either the GUI, or the `ajskill` command. For details on using the GUI, see *9.10 Killing the jobnets and the jobs being executed* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajskill` command, see the description of `ajskill` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(1) Killing a job

You can kill a job that is currently running. When you kill a job, execution is terminated for the job you specified. The killed job is then treated as having ended abnormally. You cannot kill a job that is not currently running.

Note that when you kill a job, other jobs in the same jobnet keep running. To prevent new jobs in the jobnet from executing, you must first interrupt the jobnet before killing the jobs.

When you kill a Windows job, you only terminate the processes started by JP1/AJS3.

Hence, killing a PIF file (*.pif) or a batch file (*.bat) running as a Windows job does not result in any of the commands running in those files being killed. Killing the job may not free up the resources being used by the process.

Killing a Unix job sends a SIGKILL signal that kills the process group for the job. If a process within the job has successfully executed a setgrp (set process group) system call, the processes in that process group are not killed. You will therefore need to terminate the processes in such a process group using other means, such as the kill command.

(2) Killing a jobnet

When you kill a jobnet, all running jobs including any nested jobnets are killed, and no new jobs can start. The killed jobnet is treated as having ended abnormally. You cannot kill a specific nested jobnet by itself.

The difference between killing and interrupting a jobnet is that all running jobs are forcibly terminated in the former case, but not the latter.

(3) Canceling monitoring of a jobnet with a start condition

To stop a jobnet from monitoring for a start condition, kill the root jobnet whose status is **Now monitoring**.

When a start condition is specified for a jobnet, there is typically one jobnet with the status **Now monitoring**. When the start condition is satisfied, a new generation of the jobnet is created and executed. Accordingly, if you want to stop monitoring altogether, select the original jobnet with the **Now monitoring** status and kill it.

(4) Unit statuses that can be killed

You can only kill a job or root jobnet in one of the following statuses:

Job statuses

- Now queuing
- Now running

A queueless job in *Waiting to execute* status can also be killed.

Root jobnet statuses

- Now running
- Running + Warning
- Running + Abend
- Now monitoring

4.5.11 Rerunning a job or jobnet

You can specify the execution results of a jobnet or job that has completed execution,

and rerun it.

For details about rerunning a jobnet when the execution sequence of the root jobnet is controlled using a jobnet connector, see *2.2.4(5) Re-execution of jobnet connectors and connection-destination jobnets* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

There are two methods of rerunning a job or jobnet. You can rerun the root jobnet, or rerun a nested jobnet or job. These methods are described below.

You can perform this operation using either the GUI, or the `ajsrrerun` command. For details on using the GUI, see *9.11 Re-executing jobnets and jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

Alternatively, for details on the `ajsrrerun` command, see the description of `ajsrrerun` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(1) Rerun the root jobnet

You can choose one of the following rerun methods when you rerun a root jobnet.

From abnormally ended job

This method reruns from an abnormally ended job. If the rerun job ends normally, succeeding jobs are executed and processing continues.

When a nested jobnet has ended abnormally, you can rerun from the specific job in the nested jobnet that ended abnormally.

From after abnormally ended job

This method reruns from the job that follows the abnormally ended job.

When a nested jobnet has ended abnormally, you can rerun from the job that follows the job that ended abnormally in the nested jobnet.

From abnormally ended jobnet

This method reruns from the beginning of an abnormally ended nested jobnet.

When you rerun from an abnormally ended jobnet, the jobs that ended normally in the jobnet also rerun.

If you rerun from an abnormally ended job, jobs that ended normally before the abnormally ended job are not rerun.

From top of jobnet

This method reruns from the top of the root jobnet.

Only jobs that ended with warning

This method reruns only jobs that ended with a warning. When a nested jobnet is defined, you can rerun the jobs that ended with a warning in the nested jobnet.

(2) Rerun a nested jobnet or job

You can choose from the following methods to rerun a nested jobnet or job.

From the specified jobnet or job

This method reruns from the specified jobnet or job. If the rerun jobnet or job ends normally, succeeding jobs are executed and processing continues.

From the job after the specified jobnet or job

This method reruns from the job that follows the specified job or nested jobnet.

Only the specified jobnet or job

This method reruns only the specified job or nested jobnet.

Rerunning a jobnet or job results in an error in the following cases:

- The status level of the job or jobnet you attempt to rerun is **Now running, Running + Abend, Running + Warning, Waiting for prev. to end, Waiting for start time, or Being held.**
- The preceding jobnet or job ended normally or with a warning, and the jobnet or job you attempt to rerun is a recovery jobnet or recovery job
- The jobnet or job you attempt to rerun is a judgment job or a subordinate unit in a judgment job
- When the job or jobnet that you attempted to re-execute has the recovery attribute, and the preceding job or jobnet ended normally or with a warning
- When the job or jobnet that you attempted to re-execute is a judgment job or a subordinate unit of a judgment job
- When the upper-level jobnet of the job or jobnet that you attempted to re-execute does not have a schedule

In the following cases, attempting to execute a job or jobnet does not result in an error but they will not execute:

- When execution of the job to be re-executed is canceled
- When the jobnet that you attempted to re-execute does not have a schedule, even though its upper-level jobnet does have a schedule

(3) The job preceding a judgment job ended abnormally

When the job preceding a judgment job ends abnormally, you can rerun jobnets or jobs from the root jobnet as follows.

- **From abnormally ended job**
- **From after abnormally ended job[#]**

- **From abnormally ended jobnet**
- **From top**
- **From any job or jobnet** (excludes dependent jobs and jobnets)
- **Only jobs that ended with warning**

Procedure for rerunning a jobnet from the job after the job that ended abnormally

When a job preceding a judgment job ends abnormally, if you rerun the jobnet from the next job, the status of the job that ended abnormally changes to **Ended normally**. The process continues from the next judgment job.

When the judgment job makes a judgment using the return code of the preceding job, it uses the return code from when the preceding job ended abnormally.

(4) Re-executing the dependent job of a judgment job

The execution of the dependent job of a judgment job depends on the execution result of the judgment job. You might want to re-execute the dependent job if it terminates abnormally, but the method for re-executing a dependent job differs from that for executing ordinary jobs.

When an ordinary job (PC job or Unix job) has terminated abnormally, you can select and re-execute the job. However, if a dependent job of a judgment job has terminated abnormally, you cannot re-execute it by selecting it or the judgment job. To re-execute a dependent job, select and re-execute the upper-level jobnet.

(5) Notes about rerunning jobnets or jobs

Take note of the following when you rerun a jobnet or job.

- Processes from before the rerun have still not ended

When you rerun a jobnet from a specific job, the rerun job may catch up with a job that started before the rerun and still has active processes. In this case, start the rerun job after the preceding job ends.

- Result information for a job before the rerun

When you rerun a job, the result information for the job (execution start time, execution end time, status, code) is updated to reflect the results of the rerun. However, the information in the standard output file and the standard error output file is retained.

- Changing the status of a jobnet or job that precedes the rerun jobnet

If the status of the preceding jobnet or job was **Ended abnormally**, **Killed**, **Failed to start**, **Unknown end status**, or **Not executed + Ended**, on rerunning the status is forcibly changed to **Ended normally**, **Ended with warning**, or **Bypassed**. The following table details these changes.

Table 4-6: Status change of units

No.	Status of preceding unit	Does the status apply?#1			After status of preceding unit is changed	
		Jobnet		Job	Set skipped jobs to warning?	
		Root	Nested		Yes	No
1	Not sched. to exe.	Y	Y	Y	Not sched. to exe.	Not sched. to exe.
2	Waiting for start time	Y	Y	N	Waiting for start time	Waiting for start time
3	Waiting for prev. to end	N	Y	Y	Waiting for prev. to end	Waiting for prev. to end
4	Being held	Y	Y	Y	Being held	Being held
5	Waiting to execute	N	N	Y	Waiting to execute	Waiting to execute
6	Now queuing	N	N	Y	Now queuing	Now queuing
7	Not executed + Ended [-W]	N	Y	Y	Ended with warning	Bypassed
8	Bypassed	N	Y	Y	Bypassed	Bypassed
9	Now running	Y	Y	Y	Now running	Now running
10	Running + Warning	Y	Y	N	Running + Warning	Running + Warning
11	Running + Abend	Y	Y	N	Running + Abend	Running + Abend
12	Ended normally	Y	Y	Y	Ended normally	Ended normally
13	Ended with warning	Y	Y	Y	Ended with warning	Ended with warning
14	Ended abnormally [-WR]	Y	Y	Y	Ended with warning	Ended normally
15	Skipped so not exe.	Y	Y	N	Ended with warning	Bypassed
16	Invalid exe. seq.	Y	Y	N	Ended with warning	Ended normally
17	Interrupted	Y	Y	N	V5: Ended with warning #2 V6: Interrupted #2	V5: Ended with warning #2 V6: Interrupted #2
18	Killed [-WR]	Y	Y	Y	Ended with warning	Ended normally
19	Failed to start [-WR]	N	N	Y	Ended with warning	Ended normally
20	Unknown end status [-WR]	N	N	Y	Ended with warning	Ended normally

No.	Status of preceding unit	Does the status apply?#1			After status of preceding unit is changed	
		Jobnet		Job	Set skipped jobs to warning?	
		Root	Nested		Yes	No
21	Shutdown	Y	Y	Y	N/P	N/P
22	Normal end + False	N	N	Y	Normal end + False	Normal end + False
23	Wait for start cond.	Y	Y#3	N	N/P	N/P
24	Now monitoring	Y	Y#3	N	N/P	N/P
25	Unmonitored + Ended	Y	Y#3	N	N/P	N/P
26	Monitor terminated	Y	Y#3	N	N/P	N/P
27	Interrupted monitoring	Y	Y#3	Y#3	N/P	N/P
28	Monitor-end normal	Y	Y#3	N	N/P	N/P
29	Start delay	Y	Y#3	N	#4	#4
30	End delay	Y	Y	Y	#4	#4
31	Nested jobnet delayed start	Y	Y	N	#4	#4
32	Nested jobnet delayed end	Y	Y	N	#4	#4

Legend:

Root: Root jobnet

Nested: Nested jobnet

Y: Applies

N: Does not apply

N/P: Rerun is not possible (an error message is output)

Note

Numbers 1 to 6 indicate the status when the upper-level jobnet is terminated before the preceding unit is rerun.

#1

Indicates whether each type of unit can enter the indicated status.

#2

Partial rerunning of an interrupted unit depends on the setting (V5 or V6) of the environment parameter `VRSHIFT_INTRERUN`. It is set to V6 by default. The table shows the status for each setting.

#3

Since this status only applies to a start condition (`. CONDITION`) or an event icon in a start condition, the unit cannot be a preceding unit.

#4

Delay is additional information relating to the unit status. Therefore, the delay status does not change even if the succeeding unit is rerun.

■ When rerunning the succeeding unit of a unit that has terminated abnormally

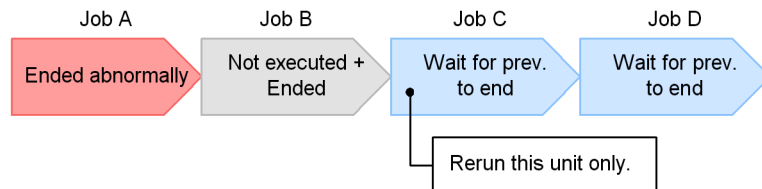
If a unit has terminated abnormally, all the succeeding units other than recovery units move to the *Not executed + Ended* status. The status is changed unit by unit in the order in which the units are linked with relation lines. Accordingly, when you rerun the succeeding unit of the abnormally terminated unit, the status of the succeeding unit might not move from the *Waiting for prev. to end* status to the *Not executed + Ended* status, depending on the timing. Before you attempt to rerun the succeeding unit of the abnormally terminated unit, wait until all the succeeding units are placed in the *Not executed + Ended* status. You must be especially careful when you rerun the succeeding unit from a recovery job.

If you rerun the succeeding unit of an abnormally terminated unit when the succeeding unit is in the *Wait for prev. to end* status, the following problems might occur:

- An attempt to rerun the unit fails

This problem occurs when the unit to be rerun is still in the *Wait for prev. to end* status.

Figure 4-32: When the unit in the Wait for prev. to end status is rerun

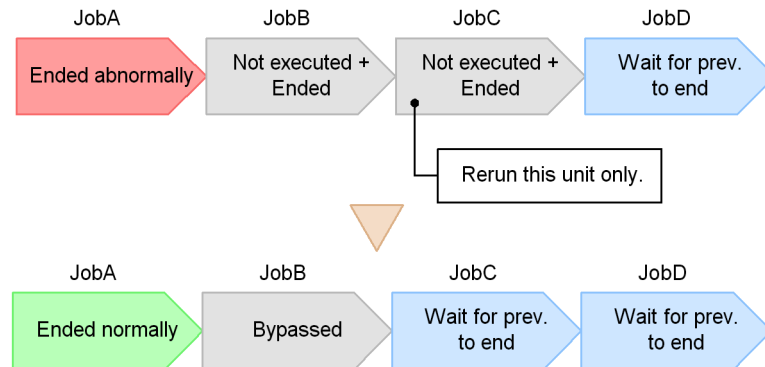


Job C is in the *Wait for prev. to end* status and therefore cannot be rerun. Wait until Job C moves to the *Not executed + Ended* status, and then rerun it.

- An attempt is made to rerun only the specified unit, but succeeding units are also run

This problem occurs when you rerun a unit in the *Not executed + Ended* status while another unit is in the *Wait for prev. to end* status, resulting in no preceding units in the *Ended abnormally* status.

Figure 4-33: When the succeeding job has a unit in the Wait for prev. to end status



When Job C in the *Not executed + Ended* status is rerun, Job C moves to the *Wait for prev. to end* status, Job A moves to the *Ended normally* status, and Job B moves to the *Bypassed* status. Because none of the preceding units for Job D are in the *Ended abnormally* status, Job D does not move to the *Not executed + Ended* status, and will therefore run after Job C is rerun.

- When concurrent execution is disabled

If you rerun a jobnet for which concurrent execution is disabled while another generation of the jobnet is running, the status of the re-executed generation changes to *Waiting for start time* and it is executed when the other generation finishes.

In the same circumstances but with concurrent execution enabled, the jobnet is re-executed immediately.

- When **Schedule skip** is set as the schedule option

If you rerun a jobnet that has **Schedule skip** set as its schedule option, and the next scheduled start time arrives while the rerun is in progress, the next schedule will not be skipped but will be executed after the rerun finishes. The **Schedule skip** option applies only when the next start time arrives and the preceding generation was executed at its scheduled start time. For details about skipping a schedule, see 3.3.3(2) *Schedule option*.

- When rerunning a root jobnet or preceding unit as the result of a judgment job

Take care when rerunning a root jobnet or preceding unit as the result of a judgment job. For details, see 2.4.3 *Dynamically changing a process depending on the result of a preceding job (example of defining a jobnet that uses a judgment job)* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

(6) Notes on resuming processing while a recovery job is running

If a job has terminated abnormally, JP1/AJS3 starts a sequence of recovery jobs or jobnets. However, if you resume the processing from the job after the abnormally terminated one, the rest of the recovery jobs or jobnets will be placed in the *Not executed + Ended* status and will not be executed. In this case, resume processing after all recovery processes have been performed.

If you want to resume the subsequent jobs even when an error is detected, consider whether you can use a judgment job to perform recovery with a dependent job when the termination code of the judgment job has exceeded a threshold.

(7) Notes on re-executing a jobnet with start conditions

When monitoring of start conditions for a root jobnet is completed and the jobnet status changes from *Now monitoring* to any of the following statuses, the status of the lower-level units becomes *Not executed + Ended*.

- Unmonitored + Ended
- Monitor terminated
- Monitor-end normal

Because the *Not executed + Ended* status is treated as an abnormal termination, be careful when you re-execute a generation of a jobnet that has any of the following statuses:

4.5.12 Changing the status of a job

You can change the status of a job or jobnet connector.

Jobs that are queued, running or has ended can be changed to another status. For queueless jobs, you can change the status even when the jobs are waiting for execution. For event jobs and OR jobs, you cannot use a threshold to change the status. For judgment jobs, you cannot change the status.

For a jobnet connector, you can change any status, other than *Not scheduled to execute* or *Wait for previous to end*, to *Ended normally*, *Ended with warning*, or *Ended abnormally*.

You can use this operation when you want to rerun a job manually, or to correct a discrepancy between the actual status and the status determined by JP1/AJS3 from the

execution results in a job-specific log, so that JP1/AJS3 reports the correct status.

When you change the status of a job or jobnet connector, the status of the upper-level jobnet also changes.

When you change the status of a currently running job to ended, the job continues running, but the system also starts executing succeeding units in line with the new job status.

If you change the status of an event job from Now running to the Ended status, execution of the succeeding job or jobnet starts just as if the event job had terminated normally. In this case, the information to be passed from the event job to the succeeding job or jobnet is not set even when the event job terminates normally. The monitoring of the event job terminates when the status is changed.

You can perform this operation using either the GUI, or the `ajschgstat` command. For details on using the GUI, see *9.12 Changing the status of jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajschgstat` command, see the description of `ajschgstat` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

4.5.13 Methods of specifying generations when executing commands

In JP1/AJS3, when you use a command to manipulate jobnet registration information (temporarily change the execution plan, rerun, kill, interrupt, or change the status of a jobnet), you can use the following methods to specify the target jobnet:

- Automatically determine the target generation
- Use execution ID to specify
- Use registration number to specify

The following subsection describes the automatic determination of the target generation, and the use of registration numbers. For details on execution IDs, see *4.2 Managing jobnet generations*.

You can also use the GUI for these operations. The operation is the same as when you specify a jobnet using an execution ID.

(1) *Determining the generation of a jobnet automatically*

This subsection describes the procedure for using a command to determine the generation of a jobnet automatically.

(a) **Automatic judgment of target generation**

When you register a jobnet for daily execution, normally JP1/AJS3 assigns an execution ID to each generation of the jobnet. Although assigned execution IDs are

unique within a scheduler service, identifying the execution ID of the target generation can be difficult when an operation such as releasing held status is performed as part of a batch process.

As a remedy in such situations, you can omit the execution ID when you use a command. JP1/AJS3 will automatically search and determine the target generation. This is called *automatic judgment of target generation*.

(b) Priority in automatic judgment operations

When you perform an operation without specifying the target generation, the generation is selected by automatic judgment. Automatic judgment is based on the status of the root jobnet generations, in accordance with the priorities listed in the following table. The judgment is based on the root jobnet status even if the operation is to be performed on a job or a nested jobnet.

Table 4-7: Priority levels for automatic judgment to find a target generation

Operation	Priority			
	Highest	High	Low	Lowest
Temporarily change plan	Now rerunning ^{#1}	Now running ^{#2}	Scheduled next	Previous completed ^{#4}
Change job status, rerun	Now rerunning ^{#1}	Now running ^{#2}	Previous completed ^{#3}	Scheduled next ^{#4}
Interrupt, kill	Now rerunning ^{#1}	Now running ^{#2}	Previous completed ^{#4}	Scheduled next ^{#4}

#1

Now rerunning refers to any of the following statuses resulting from re-execution:

- Now running
- Running + Warning
- Running + Abend
- Wait for start time
- Being held

#2

Now running refers to any of the following statuses, unless the status results from re-execution:

- Now running
- Running + Warning

- Running + Abend
- Being held
- Now monitoring

#3

Previous completed refers to the generation whose termination time is closest to the current time. For a jobnet with a start condition, however, this applies only to a monitoring generation.

#4

Operations cannot be performed on generations in either of these statuses and might result in an error if attempted.

If there are multiple *Now running* generations and none in *Now rerunning* status, or if there are multiple *Now rerunning* generations, the criteria used in automatic judgment depend on whether a start condition applies. The judgment criteria for selecting the target generation of an operation are as follows:

When a start condition does not apply

The generation whose scheduled start time is closest to the current time is selected as the target.

When a start condition applies

- If there is a monitoring generation in *Now monitoring* status, and an execution generation in *Now running* status, the monitoring generation is selected as the target.
- If there is a monitoring generation that has finished monitoring, and an execution generation in *Now running* status, the first execution generation triggered when the start condition is satisfied is selected as the target.
- If there are multiple monitoring generations (including any that have finished monitoring), the monitoring generation whose start time is closest to the current time takes precedence. This generation is selected if its status is *Now monitoring*. If its status is other than *Now monitoring*, the first execution generation triggered when the start condition is satisfied is selected.
- If a jobnet with a start condition has a monitoring generation in *Now monitoring* status, and an execution generation in *Now rerunning* status, the rerunning generation is selected as the target.

An unintended generation could be selected as the target of an operation based on automatic judgment if the status of the root jobnet changes. This could occur, for

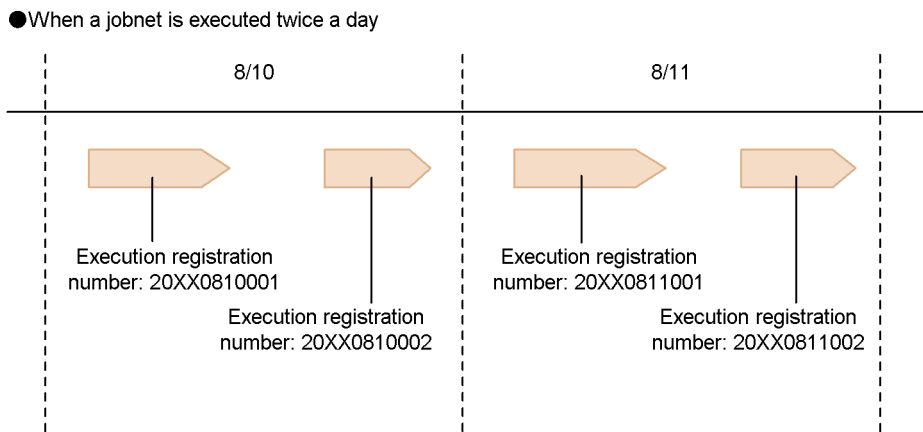
example, when the scheduled start time of the next execution schedule arrives or when the active generation ends normally. If a jobnet is executed repeatedly on a daily basis, or if the next execution schedule is due to start very soon, the status of the root jobnet might have changed by the time a command operation is executed. In such cases, execute the `ajsshow` command to obtain the execution ID of the generation you want to target, and operate on the execution registration information, specifying the execution ID you obtained. For the `ajsshow` command syntax, see *ajsshow* in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(2) Execution registration numbers

An execution registration number is a serial number assigned to each generation of a jobnet. This number is assigned in order according to the execution start time of the generation. The format of an execution registration number is *YYYYMMDDNNN* (*YYYY*: year of execution, *MM*: month of execution, *DD*: Day of execution, *NNN*: order of execution on execution day).

The following example shows how execution registration numbers are assigned when a jobnet is executed twice in one day.

Figure 4-34: Assigning execution registration numbers



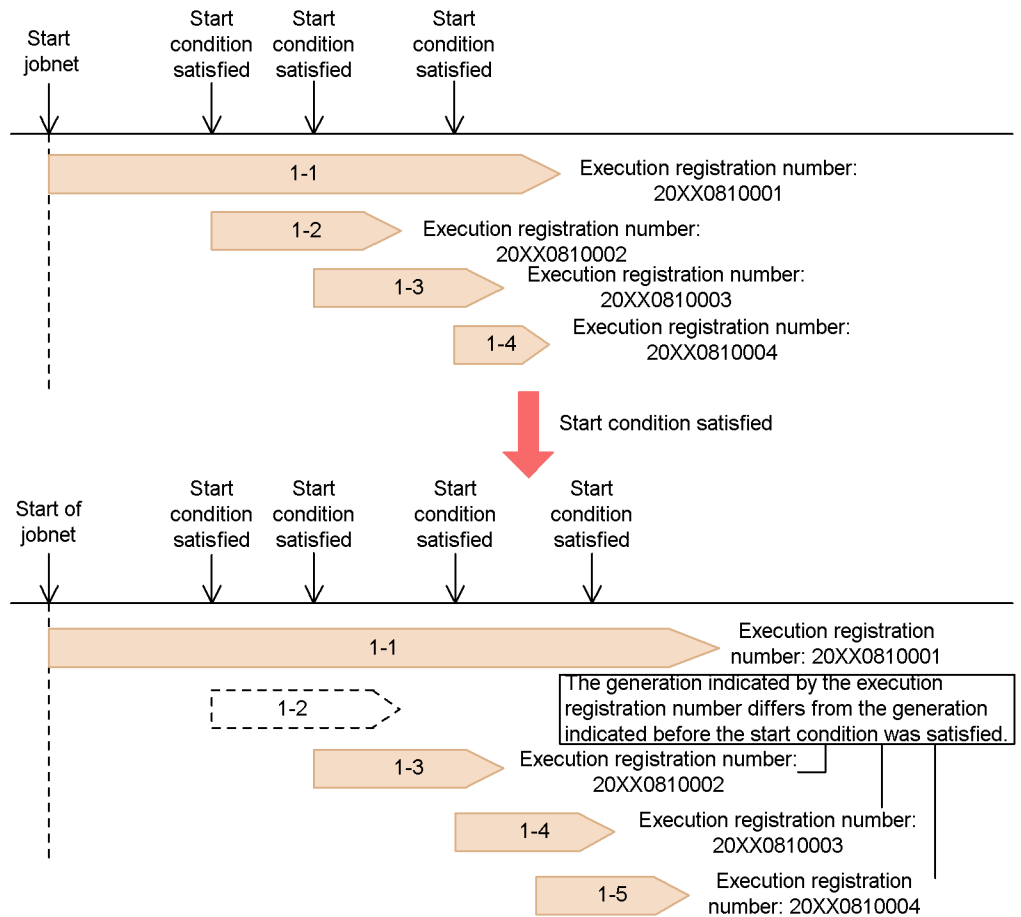
Assume that a jobnet has an execution date of August 10 20XX. Based on this information, the execution registration number of the generation that is executed first will be 20XX0810001. The generation that is executed second will have the execution registration number 20XX0810002.

In this manner, execution registration numbers in the format *YYYYMMDD001*, *YYYYMMDD002* are assigned in order of execution to each generation of a jobnet that is executed within one day. It is therefore much easier to specify a generation using an execution registration number, than by other means such as an execution ID.

Execution registration numbers are assigned according to the generations that exist at the time. Therefore, the same numbers can refer to different generations at different times, and depending on the timing of the command, you may not specify the generation you intended. The following figure shows an example where an execution registration number refers to different generations before and after a start condition is met.

Figure 4-35: Registration numbers indicate different generations before and after a start condition is satisfied

- Execution date of jobnet: August 10 20XX
- Number of logs to keep: 3



Legend:

: Generation deleted due to the number of logs to keep being exceeded

In the above figure, the execution registration number 20XX08010002 corresponds to generation 1-2 before the start condition is satisfied. However, generation 1-2 is deleted by the generation management process when the start condition is next satisfied. Hence, the execution registration number 20XX08010002 corresponds to generation 1-3 after the start condition is met.

Other circumstances in which the execution registration number can change include:

- When a generation is deleted due to the maximum number of logs to keep being exceeded.
- When a generation is added or moved due to a time change.
- When a schedule is added because a generation is either registered for fixed execution specifying an execution period or date, or registered for immediate execution.
- When a generation is deleted by the execution prohibited function.

In an operation where generations of a jobnet are constantly being updated, we recommend that you use execution IDs instead of execution registration numbers to manipulate jobnets.

4.5.14 Automatically switching a jobnet definition at a specified time

You can automatically switch the definition of a registered jobnet at a specified time.

The functionality for automatically switching a jobnet definition at a specified time is called *releasing a jobnet*. Using this functionality, you define the jobnet in advance, and preset the date and time at which the definition is to apply. The jobnet definition is then switched automatically. The jobnet release function is useful when you want to:

- Change a jobnet definition at a specified time without stopping the jobnet.
- Change the jobnet definition without changing the name of the jobnet.
- Set up a jobnet definition change in advance.

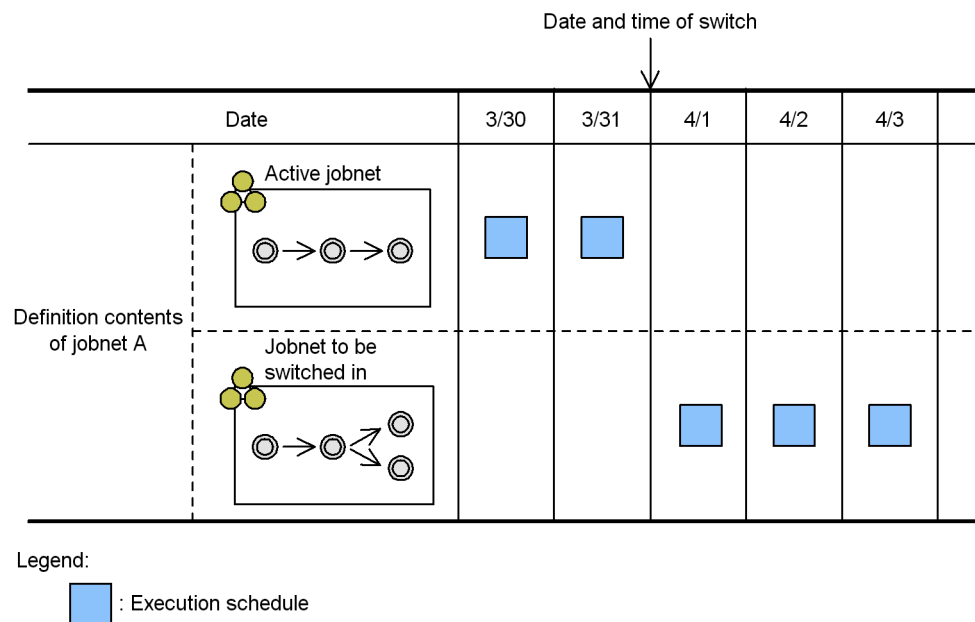
You can perform release operations using JP1/AJS3 - View or the `ajsrelease` command. For the procedure using JP1/AJS3 - View, see *9.14 Switching definitions of a running jobnet according to a plan (jobnet release)* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. For the command operation, see `ajsrelease` in *2. Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(1) Overview of using the jobnet release function

To change a jobnet definition using the release function, you must define both the jobnet you want to replace, and the jobnet you want to replace it with. Run the first jobnet in the usual way by registering it for execution. In the other jobnet, specify what jobnet it is to replace and the timing. When the specified time arrives, the jobnet

definition is switched as shown in the following figure.

Figure 4-36: Overview of using the jobnet release function



The registration of a predefined jobnet to replace an active jobnet at a specified time is referred to as a *release entry*. The predefined jobnet to be switched in is referred to as the *release source jobnet*, and the active jobnet to be replaced is referred to as the *release target jobnet*. The action of switching the jobnet definition at the time specified in the release entry is referred to as *release*.

(2) Process of releasing a jobnet definition

Under the jobnet release function, the jobnet definitions are distinguished by *release ID*. When you register the release source jobnet, in addition to its release ID, you specify the release date and time (for switching jobnet definitions) and the release target jobnet (the active jobnet that is to be replaced).

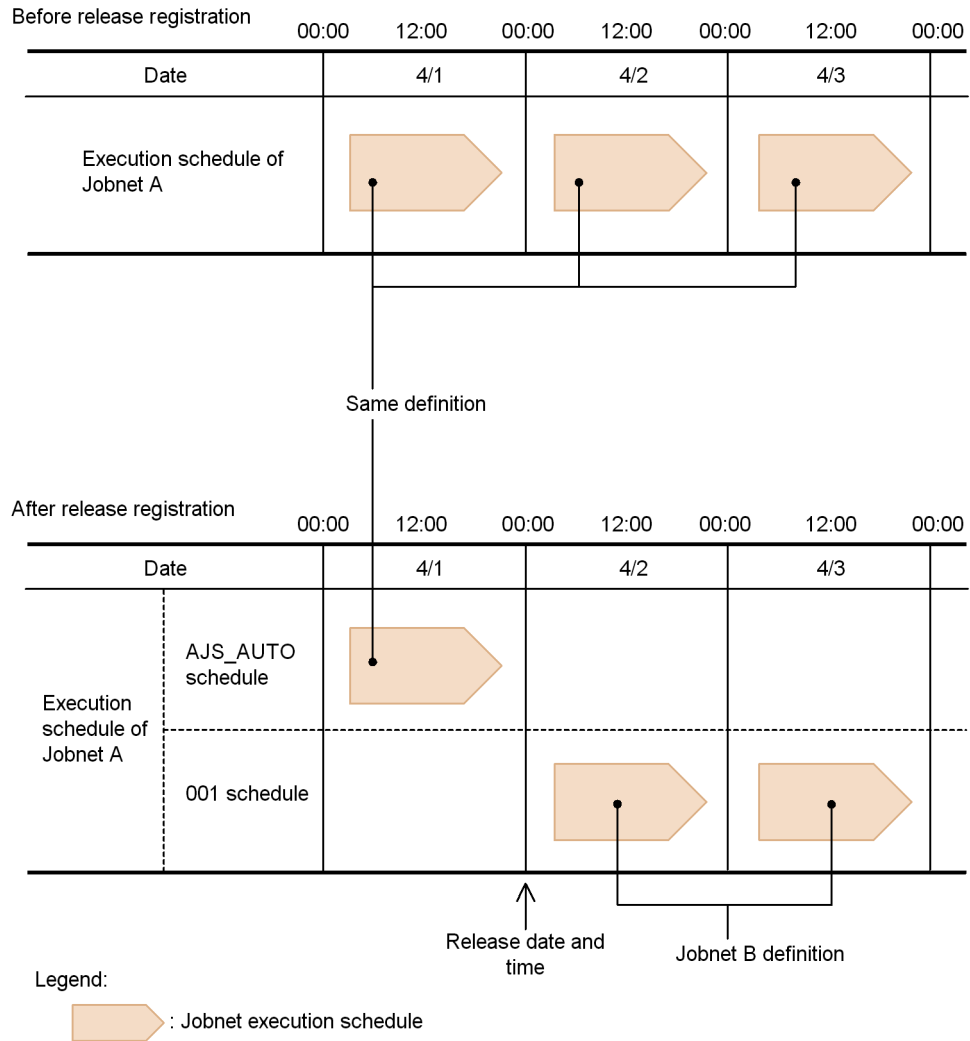
At release entry, the definition of the release source jobnet is copied and associated with the specified release ID. When the specified release time arrives, the jobnet definition associated with the release ID is released and replaces that of the active release target jobnet.

The jobnet definition copied at release entry is managed as one of the definitions of the release target jobnet. Since it is a copy, you can edit or delete the release source jobnet definition without affecting the release target jobnet. However, because the original release source jobnet is the master copy used when changing the definition of the release target jobnet, make a backup copy of the definition as needed. For details about

using the jobnet release function, see 8.3.8 *Using the jobnet release function* in the *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*.

The following figure illustrates how a jobnet definition is switched using the jobnet release function.

Figure 4-37: Example of releasing a jobnet



In this example, the definition of Jobnet B is to be released at 00:00 on 4/2 for Jobnet A. Jobnet B is registered with the release ID 001 and the release time of 00:00 on 4/2, thereby generating the following two release IDs:

AJS_AUTO

The release ID associated with the definition of the active jobnet. This release ID is created automatically at release entry.#

001

The release ID associated with the copied definition of Jobnet B, the release source jobnet.

The AJS_AUTO definition applies until 23:59 on 4/1. At 00:00 on 4/2, the copied definition of Jobnet B associated with release ID 001 is released and replaces the Jobnet A definition. The jobnet name remains as *Jobnet A* after release.

#

Release ID AJS_AUTO is created only at release entry and is not created at any subsequent release entry. For details about the jobnet status and status transitions after release entry, see 8.3.3 *Jobnet status after registration of a jobnet-definition release* in the *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*.

For details about the jobnet release function, see 8.3 *Switching a jobnet definition while the jobnet is registered for execution* in the *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*.

4.5.15 Changing job and jobnet definitions without unregistering the jobnet

You can change the definitions of the jobnets and jobs below a root jobnet that is registered for execution, without canceling the registration of the jobnet.

You can use the *suspend* function to change the definitions below the root jobnet without canceling registration of the root jobnet. Suspending the root jobnet means suppressing unit execution over all generations of the specified root jobnet. Units are not executed once they are suspended. However, any units that are already running continue to be processed.

When you change the lower-level definitions of a registered root jobnet, use the suspend function to prevent malfunctions such as the definition processing being out of sync with the execution control processing. By suspending the root jobnet, you can change definitions by synchronizing them with the execution control processing.

You can perform suspend operations using JP1/AJS3 - View or the *ajssuspend* command. For the procedure using JP1/AJS3 - View, see 9.13 *Changing the lower-level definition of a jobnet registered for execution* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. For the command operation, see *ajssuspend* in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(1) Changing definitions while the root jobnet is registered for execution

You can change the lower-level definitions under a root jobnet while the root jobnet is suspended.

The following conditions are required for you to be able to edit definitions while the root jobnet is registered for execution.

- The root jobnet is suspended
- The target is not being edited exclusively elsewhere

(a) Changes you can make to definitions

The changes you can make to definitions while the root jobnet is registered for execution, and the changes that are unavailable are listed below.

Changes that are available

- Adding a new unit
- Changing an existing definition
- Deleting an existing unit
- Changing the map size
- Adding a start condition

You can also change the definition of units defined under a remote jobnet.

Changes that are unavailable

- Changing the name of an existing unit

However, you can change the name of a unit added during suspension.

- Moving a unit

To move a unit, first copy the source unit and paste it to the destination. Then delete the source unit.

- Deleting a running unit

The following paragraphs describe in detail the types of editing you can perform on lower-level definitions while the root jobnet is registered for execution, and the points to consider when editing a definition.

Adding a new unit

You can add a new unit. The status of the added unit during suspension depends on the status of the jobnet one level above the unit. The following table shows the status of the unit one level above the added unit, and the status of the added unit.

Table 4-8: Status of a unit added during suspension

Status of unit one level above added unit	Status of added unit
Waiting	Not scheduled
Now running	Not scheduled
Ended	Bypassed

Changing an existing definition

You can change an existing definition. However, note the following:

- You cannot change the name of an existing unit.
However, you can change the name of a unit added during suspension.
- You can change the definition of a unit that is running. However, you cannot delete it.
- Even if you change an existing definition, the past execution results remain.
- When you change an existing definition, the jobnet may execute in a different configuration from before. Be aware of this when you rerun the jobnet.
- The dummy schedule for a jobnet registered for planned execution, displayed in the Daily Schedule or Monthly Schedule window, is calculated from the changed definition information even while the jobnet is suspended.

Deleting an existing unit

You can delete an existing unit. However, note the following:

- When you delete an existing unit, past execution results for the unit are also deleted, and you can no longer display them. If you need this past history information, either reference the scheduler log information, or save the information using the `ajsshow` command, before deleting the unit. For the `ajsshow` command syntax, see *ajsshow* in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.
- When you delete an existing unit, the jobnet may execute in a different configuration from before. Be aware of this when you rerun the jobnet.

Changing the map size

You can change the map size during suspension, regardless of the status of the jobnet.

Adding a start condition

When you use a start condition, first you create the start condition object (.CONDITION). You then set the schedule by using schedule rules, or by temporarily changing the execution plan.

However, the schedule is recalculated differently depending on the method of execution registration. When you add a start condition for a jobnet that did not previously have a start condition, the effect is as follows:

- When the jobnet is registered for immediate execution
Even if you add a start condition, it will not be valid.
- When the jobnet is registered for planned execution
The added start condition is valid from the next generation that is scheduled for execution.
- When the jobnet is registered for fixed execution in a specified period
Even if you add a start condition, it will not be valid.
- When the jobnet is registered for fixed execution for a specific number of future generations
The added start condition is valid from the first generation created after the suspension is released.

When you edit the definition of a jobnet that is registered for execution, the changes are reflected in the past execution results, and all running and scheduled generations. For example, if you edit the definition of a jobnet and then display the past history of the jobnet in the Jobnet Monitor window, the status for the edited jobnet is displayed.

(b) Available changes according to unit type and status

The changes you can make to the definition of a unit under a root jobnet that is registered for execution differs according to the type and status of the unit.

The following table shows the different changes you can make for each type of unit.

Table 4-9: Unit types and available editing operations

Edit operation		Unit		
		Job	Jobnet	
Editing a unit	Adding a unit		Y	Y
	Deleting a unit	Running unit	N	N
		Non-running unit	Y	Y
Editing a relation line	Adding a relation line		Y	Y

Edit operation			Unit	
			Job	Jobnet
	Deleting a relation line		Y	Y
Editing unit information	Changing a unit name	New unit	Y	Y
		Existing unit	N	N
	Changing a comment		Y	Y
	Changing the execution host		Y	Y
	Changing a definition		Y	Y
	Changing the transfer file		Y	--
	Changing an attribute	Type	Y	Y
		Other attribute	Y	Y
Other	Changing the map size		--	Y

Legend:

Y : Available

N : Not available

-- : Not applicable

Note that the available editing options differ, depending on the status of the unit you are editing. The following table shows the different editing options available according to the status of the unit.

Table 4-10: Editing options available for units according to status

Edit operation		Unit status		
		Waiting for prev. to end	Running	Ended
Editing a unit	Adding a unit	--	--	--
	Deleting a unit	Y	N	Y
Editing a relation line	Adding a relation line	Y	Y	Y
	Deleting a relation line	Y	Y	Y

4. Executing an Application

Edit operation			Unit status		
			Waiting for prev. to end	Running	Ended
Editing unit information	Changing a unit name	New unit	Y	--	Y
		Existing unit	N	N	N
	Changing a comment		Y	Y	Y
	Changing an execution host		Y	Y	Y
	Changing a definition		Y	Y	Y
	Changing the transfer file		Y	Y	Y
	Changing an attribute	Type	Y	N	Y
		Other attribute	Y	Y	Y
Other	Changing the map size (jobnet)		Y	Y	Y

Legend:

Y : Available

N : Not available

-- : Not applicable

Table 4-11: Editing options available according to start condition status

Edit operation			Status of start condition			
			No start condition	Waiting for prev. to end	Monitoring	Ended
Start condition	Setting a start condition		Y	--	--	--
	Deleting a start condition		--	Y	--	Y
Editing a unit	Adding a unit		Y	Y	--	Y
	Deleting a unit		Y	Y	--	Y
Editing unit information	Changing a unit name	New unit	Y	Y	--	Y

Edit operation		Status of start condition			
		No start condition	Waiting for prev. to end	Monitoring	Ended
	Existing unit	--	N	--	N
	Changing a comment	Y	Y	--	Y
	Changing an execution host	Y	Y	--	Y
	Changing a definition	Y	Y	--	Y
	Changing an attribute	Y	Y	--	Y
Other	Changing the map size	Y	Y	--	Y

Legend:

Y : Available

N : Not available

-- : Not applicable

(2) Procedure for editing definitions

The following describes how to change a definition below a registered root jobnet.

(a) Basic procedure for editing definitions

Follow this basic procedure to change a definition below a root jobnet that is registered for execution.

To change a definition below a root jobnet that is registered for execution:

1. Suspend the root jobnet that is registered for execution.
2. Edit the definition below the root jobnet
3. Release the suspension

To edit a definition that is monitoring for a start condition, you must kill the jobs that are monitoring for a start condition before suspending the root jobnet. Follow this procedure to edit a definition that is monitoring for a start condition.

To edit a definition that is monitoring for a start condition:

1. Kill the jobs that are monitoring for a start condition.
2. Suspend the root jobnet.
3. Edit the definition.

4. Executing an Application

4. Release the suspension.
5. Execute the root jobnet by adding a schedule.

(b) Activating the suspend function

Before you can suspend a job or jobnet, you must activate the suspend function in advance by running the `ajssetup` command. For the command syntax, see *ajssetup* in 2. *Commands Used during Setup* in the manual *Job Management Partner 1/ Automatic Job Management System 3 Command Reference 2*.

To activate the suspend function, execute the command as follows:

```
ajssetup -F scheduler-service-name -m
```

Although you can execute the `ajssetup` command while the scheduler service is running, your settings take effect only after the scheduler service restarts. You should therefore restart the scheduler service after executing the `ajssetup` command. Use the following procedure to activate the suspend function.

To activate the suspend function:

1. Disconnect JP1/AJS3 - View if it is connected.
2. Stop the JP1/AJS3 service.
3. Execute the command `ajssetup -F scheduler-service-name -m`.
4. Restart the JP1/AJS3 service.

Cautionary note

Once you activate the suspend function using the `ajssetup -m` command, the setting is permanent.

(3) Suspend operations

(a) Executing the suspend function

You can suspend a root jobnet using either the JP1/AJS3 - View window or the appropriate command.

Conditions required for suspension

The following conditions must be met before you can suspend a root jobnet:

- The JP1/AJS3 service is running
- There are no generations monitoring for a start condition (the status of the start condition is waiting or ended)
- The user who executes the command has the required permission

Accessing the suspend function from the JP1/AJS3 - View window

In the JP1/AJS3 - View window, from the **Operations** menu choose **Suspension** and then **Suspend**.

Accessing the suspend function using a command

Execute the `ajssuspend` command, specifying the `-s` option.

When you execute the suspend function, you can specify whether to suspend the jobnet if a unit is executing.

(b) Checking the suspension status

To check whether the root jobnet is suspended, use the JP1/AJS3 - View window or the appropriate command.

Using the JP1/AJS3 - View window

An icon indicating the suspension status is displayed in the list area of the JP1/AJS3 - View window.

Using a command

Execute the `ajsshow` command with the `-i` option, and specify the 2-byte format indicator `%SP`.

The command format is as follows.

```
ajsshow -i %SP
```

For details on the `ajsshow` command, see the description of `ajsshow` in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(c) Releasing the suspension

You can release a suspended root jobnet using the JP1/AJS3 - View window, or an appropriate command.

Conditions for releasing suspension

The following conditions must be met before you can release a suspended root jobnet:

- The JP1/AJS3 service is running.
- The user has the appropriate permission to perform the operation.
- The target is not being edited exclusively elsewhere

Releasing suspension using the JP1/AJS3 - View window

In the JP1/AJS3 - View window, from the **Operations** menu choose **Release**

Suspension.

Releasing suspension using a command

Execute the `ajssuspend` command, specifying the `-C` option.

If you cold-start the JP1/AJS3 service, suspension is released automatically, and execution registration is also canceled automatically.

(d) Behavior of added units after suspension is released

When you release the suspension, you can specify the behavior of units added under a running jobnet. However, this does not apply to units added under a remote jobnet.

You can specify one of the following three options. The name of the option, and the behavior of the unit after suspension is released are described in the paragraphs below.

Execute

This option executes the added units.

When you release the suspension, units added directly below a running jobnet enter the **Waiting for prev. to end** status. These units are executed as soon as the previous unit ends. If all preceding units have already ended normally when you release the suspension, then the added units are executed immediately.

This option is assumed if you execute the command without specifying an option.

Do not execute (stop)

This option cancels execution of the added units.

When you release the suspension, execution of the units that were added directly under a running jobnet is canceled. The units then have the status **Bypassed**.

Hold

This option temporarily changes the status of the added units to **Held**.

When you release the suspension, the units that were added directly under a running jobnet temporarily change to **Held** status.

The status of an added unit when you release the suspension with one of these options specified, depends on the status of the unit one level above the added unit. The following table shows the relationship between the status of the upper-level unit and the status of the added unit.

Table 4-12: Status of added unit when suspension is released with specified option

Status of unit one level above added unit	Option		
	Execute or no option specified	Do not execute (stop)	Hold
Running	Waiting for previous unit to end (changes to Now running when the previous unit ends normally)	Bypassed	Waiting for previous unit to end (changes to Being held when the previous unit ends normally)
Waiting	Waiting for previous unit to end	Waiting for previous unit to end	Waiting for previous unit to end
Ended	Bypassed	Bypassed	Bypassed

Cautionary note

If a unit that was added during suspension is bypassed, the plan change for that unit is set to Execution prohibited. If you need to execute the unit (by rerunning it, for example), you can add an execution schedule by changing the time (only applies if the unit is a jobnet), or releasing the changes made by the change plan.

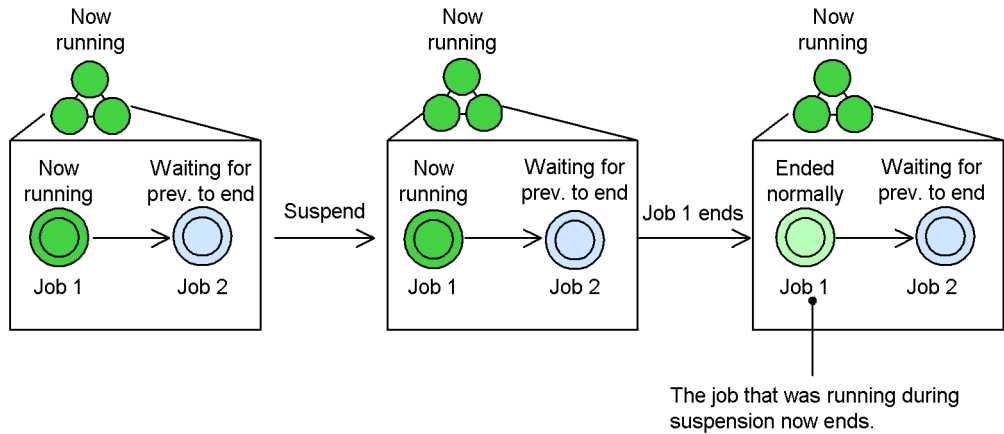
(4) Status change of suspended jobs and jobnets

(a) Status change of suspended jobs

The status of a suspended job changes as follows:

- A job that is already running when the root jobnet is suspended continues to run. When the timeout period of the job expires, the job no longer executes. When the job finishes executing, its status changes to Ended. Delay monitoring also continues. When the delay time arrives, the job enters *Delayed* status, but upper-level units are not affected. The status of upper-level units changes when suspension is released. The following figure shows the suspension of a job in *Now running* status.

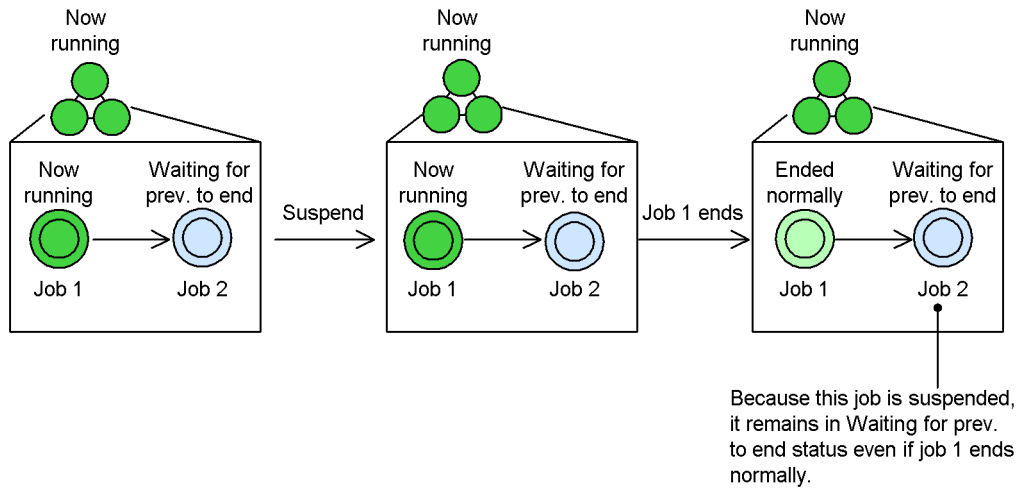
Figure 4-38: Status change of suspended job 1 (in "Now running" status)



- Units under the suspended root jobnet will not be executed again.

The following figure shows the suspension of a job in *Waiting for previous to end* status.

Figure 4-39: Status change of suspended job 2 (in "Waiting for previous to end" status)

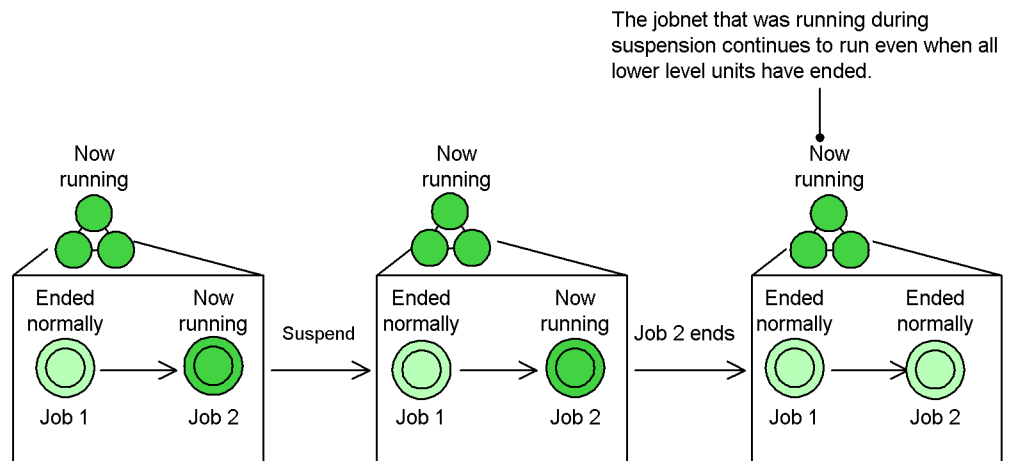


(b) Status change in suspended jobnets (excluding remote jobnets)

- A jobnet (excluding remote jobnets) that has the Now running status when the root jobnet is suspended retains that status even if all lower-level units have ended. In addition, delays and timeout periods (skipping) are not monitored.

The following figure shows the suspension of a jobnet in *Now running* status.

Figure 4-40: Status change of suspended jobnet (in "Now running" status)



(c) Status change in suspended remote jobnets

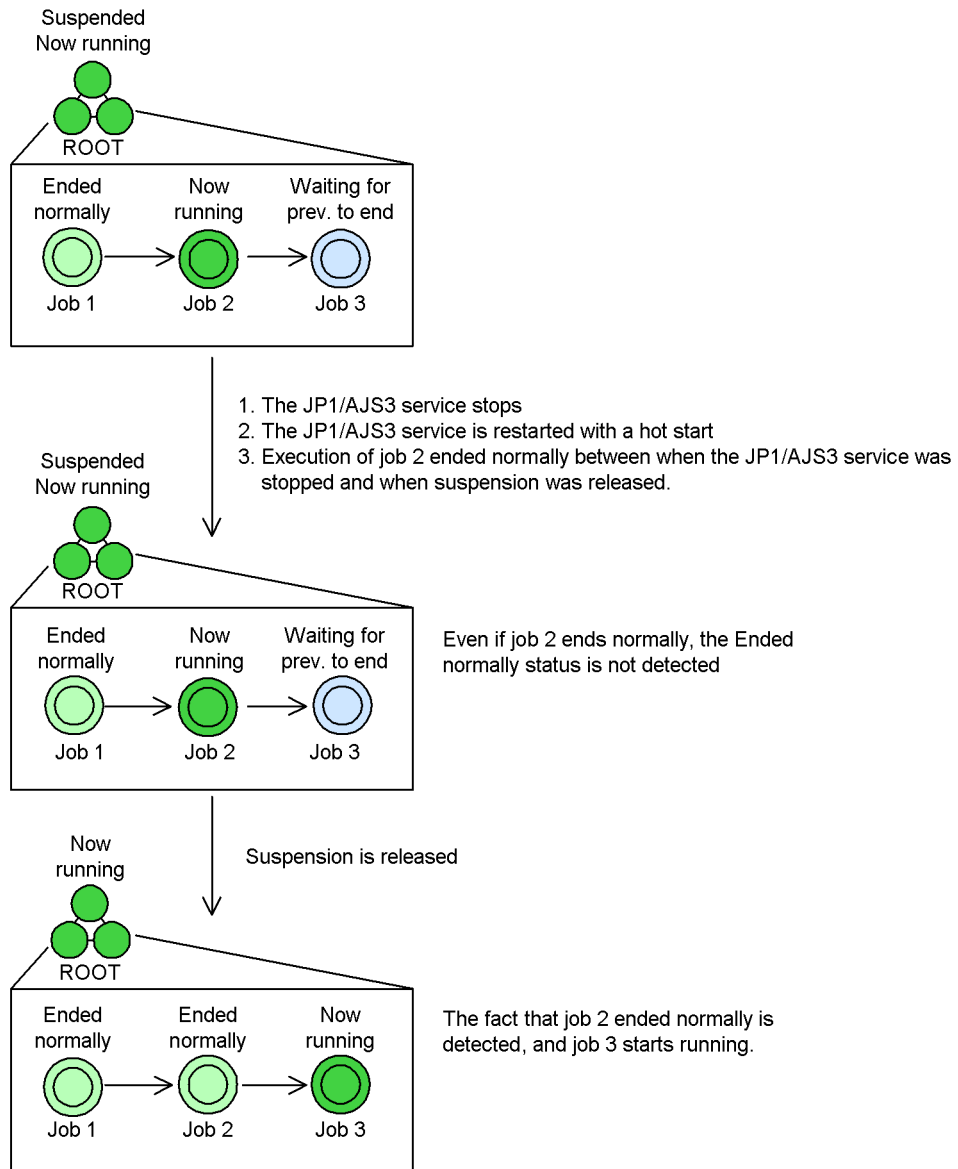
When you suspend the root jobnet, only the status of a remote jobnet that is running changes. The statuses of succeeding jobs and upper-level units do not change.

(d) Status change when the JP1/AJS3 service is hot-started during suspension

If the JP1/AJS3 service stops while the root jobnet is suspended and is then hot-started, jobs with the *Now running* status continue to execute. However, the status of such jobs does not change to ended when the jobs finish executing. The status of the jobs only changes to ended when you release the suspension.

The following figure shows the status change when the JP1/AJS3 service is hot-started during suspension.

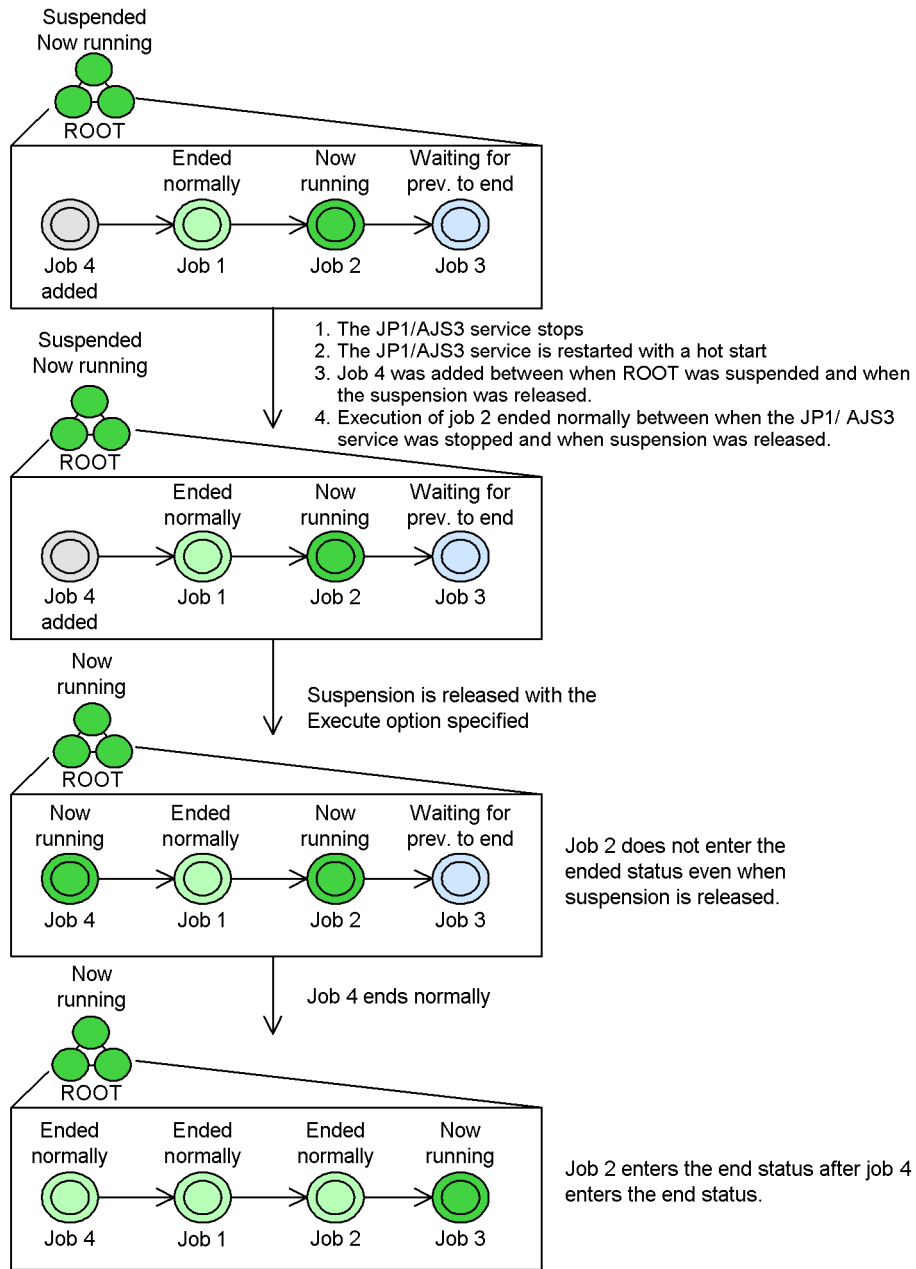
Figure 4-41: Status change when the JP1/AJS3 service is hot-started during suspension



Assume that you hot-start the JP1/AJS3 service while the root jobnet is suspended. Assume that you then add a unit as a preceding unit to a running unit, and specify the **Execute** option when you release suspension of the root jobnet. The status of the unit that was running changes to ended after the unit you added terminates.

The following figure shows the status change when the JP1/AJS3 service is hot-started during suspension.

Figure 4-42: Status change when the JP1/AJS3 service is hot-started during suspension (Execute option specified)



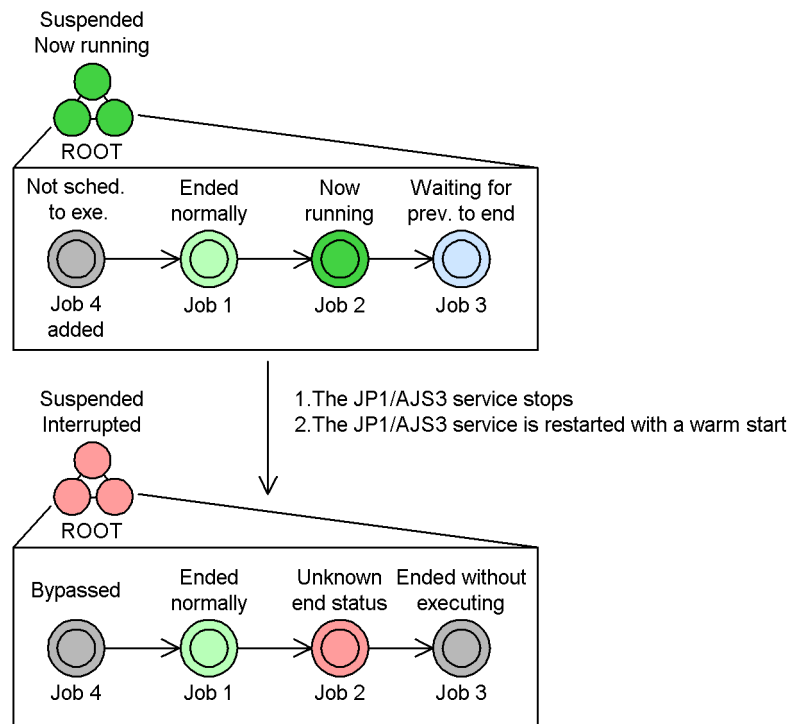
(e) Status change when the JP1/AJS3 service is warm-started during suspension

If the JP1/AJS3 service stops while the root jobnet is suspended and is then warm-started, the root jobnet maintains its suspended status. However, the status of running jobs changes to **Unknown end status**, the status of running jobnets changes to **Interrupted**, and the status of units added below a running jobnet changes to **Bypassed**.

An end delay is not detected if the delay monitor time is reached for a unit during suspension.

The following figure shows the status change when the JP1/AJS3 service is warm-started during suspension.

Figure 4-43: Status change when the JP1/AJS3 service is warm-started during suspension



(f) Notes on restarting the JP1/AJS3 service with a cold-start during suspension

If the JP1/AJS3 service stops while the root jobnet is suspended, and you restart it with a cold-start, the unit records deleted during the suspension remain in the database as

invalid records. In such a case, execute the following command on all root jobnets to delete the invalid records.

```
ajssuspend -U -R -T /
```

(5) Scheduling after suspension is released

(a) Allocating a schedule when the suspension is released

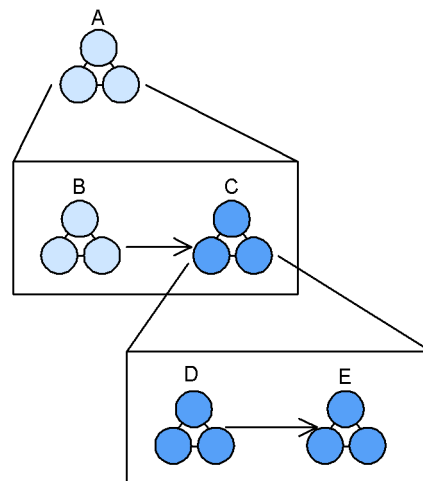
The allocation of a schedule to a unit after the suspension is released is described below, according to the category of execution registration that applies to the unit.

When the root jobnet is registered for immediate execution or fixed execution with a specified date

As is normal with immediate execution registration and fixed execution registration with a specified date, the date specified in the fixed execution registration time is allocated as the scheduled start time. However, if the upper-level jobnet does not have an execution schedule (execution is prohibited), then the scheduled start time of the added jobnet becomes **None**, since although the scheduled start time of the added jobnet is allocated as above, the upper-level jobnet does not have a scheduled start time.

The following example shows an example of suspending the root jobnet and adding units.

Figure 4-44: Example of adding units to a root jobnet that is registered for immediate execution



Note: The darker units are the units added when the root jobnet was suspended.

The following table shows how the schedule is allocated after releasing suspension.

Table 4-13: Schedule after releasing suspension (for immediate execution registration and fixed execution registration with a specified date)

Unit	11/30	12/1	12/2	12/3
A	--	12:00	--	--
B	--	12:00	--	--
C	--	12:00	--	--
D	--	12:00	--	--
E	--	12:00	--	--

Legend:

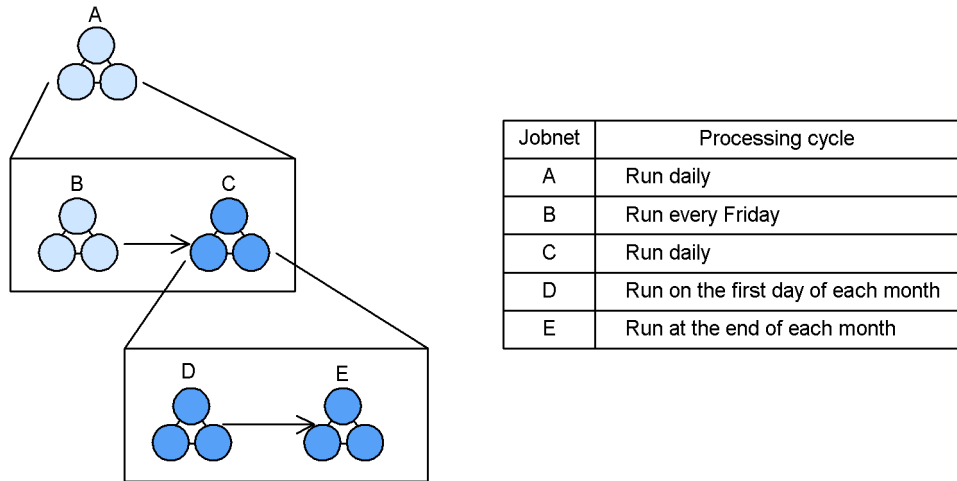
-- : Does not apply

When the root jobnet is registered for planned execution

Schedules are recalculated for all jobnets in waiting generations whose root jobnet has not been rerun. If the root jobnet for a certain generation is running or rerunning, then schedules are recalculated only for the added jobnets in that generation.

The following figure shows an example of adding units after suspending the root jobnet that is registered for planned execution.

Figure 4-45: Adding units to a root jobnet registered for planned execution



Note: The darker units are the units added when the root jobnet was suspended.

Imagine the current date is December 1 (Friday), and you change the configuration of the application before it starts on December 1 (Friday). The application for November 30 (Thursday) is rerunning, and jobnet B is currently running. Based on this information, the following schedule is allocated after the suspension is released.

Table 4-14: Schedule after suspension is released (when registered for planned execution)

Jobnet	11/24 (Fri)	11/25 (Sat)	11/26 (Sun)	11/27 (Mon)	11/28 (Tue)	11/29 (Wed)	11/30 (Thu)	12/1 (Fri)	12/2 (Sat)
A	EN	EN	EN	EN	EN	EN	NR	(WS)	WS
B	BP	BP	BP	BP	BP	BP	NR	(WP)	NS
C	BP	BP	BP	BP	BP	BP	(WP)	(WP)	WP
D	BP	BP	BP	BP	BP	BP	(NS)	(WP)	NS
E	EN	BP	BP	BP	BP	BP	(WP)	(NS)	NS

Legend:

EN : Ended normally

BP : Bypassed

- NR : Now running
- WS : Waiting for start time
- WP : Waiting for prev. to end
- NS : Not sched. to exe.
- () : Recalculated schedule

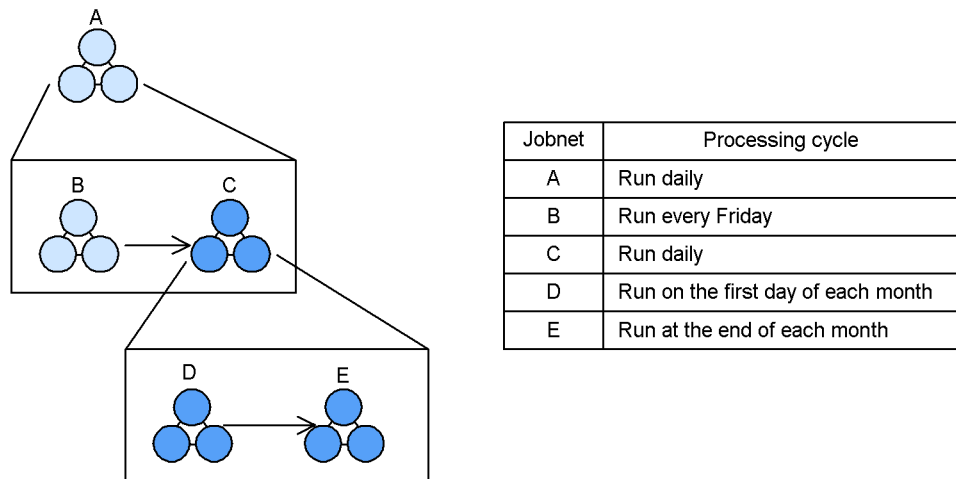
The schedule for November 30 is recalculated for jobnet C, and for the jobnets below jobnet C (D and E). Therefore, even if an exclusive schedule is defined for jobnet B and jobnet C, it is invalid. On December 12, the exclusive schedule is valid, since the schedule is recalculated from the uppermost jobnet.

When the root jobnet is registered for fixed execution

The schedule is recalculated for jobnets that were added to a jobnet whose root jobnet is either waiting or running.

The following figure shows an example of adding units after suspending a root jobnet that is registered for fixed execution.

Figure 4-46: Adding units to a root jobnet registered for fixed execution



Note: The darker units are the units added when the root jobnet was suspended.

Imagine the current date is December 1 (Friday), and you change the configuration of the application before it starts on December 1 (Friday). The application for November 30 (Thursday) is rerunning, and jobnet B is currently running. Based on this information, the following schedule is allocated after the suspension is released. Assume that the root jobnet is registered for fixed

execution until December 6.

Table 4-15: Schedule after suspension is released (when registered for fixed execution)

Jobnet	11/29 (Wed)	11/30 (Thu)	12/1 (Fri)	12/2 (Sat)	12/3 (Sun)	12/4 (Mon)	12/5 (Tue)	12/6 (Wed)
A	EN	NR	WS	WS	WS	WS	WS	WS
B	BP	NR	WP	NS	NS	NS	NS	NS
C	BP	(WP)	(WP)	(WP)	(WP)	(WP)	(WP)	(WP)
D	BP	(NS)	(WP)	(NS)	(NS)	(NS)	(NS)	(NS)
E	BP	(WP)	(NS)	(NS)	(NS)	(NS)	(NS)	(NS)

Legend:

EN : Ended normally

BP : Bypassed

NR : Now running

WS : Waiting for start time

WP : Waiting for prev. to end

NS : Not sched. to exe.

() : Recalculated schedule

When a jobnet is registered for fixed execution over a specified period, the schedules of existing generations are not recalculated when the suspension is released. Exclusive schedules defined for the hierarchical level that contains jobnets added within the fixed period (the hierarchical level of jobnets B and C) are invalid. When a jobnet is registered for fixed execution with a specific number of future generations, the exclusive schedule is valid for generations created after the suspension is released.

(b) Execution schedules after suspension is released

When you register a jobnet for planned execution or fixed execution with a specific number of future generations, JP1/AJS3 assigns a new generation each time the jobnet is executed. Therefore, when you display past results and future plans using the Monthly Schedule or Daily Schedule window, or the `ajsshows` command, the execution schedules shown below are created by schedule simulation.

- When the jobnet is scheduled for planned execution:

Execution schedules are created for the generation after next, and succeeding

generations.

- When the jobnet is scheduled for fixed execution with a specific number of future generations:

Execution schedules created for generations after the specified number of future generations.

Therefore, if the suspension is not released before the next scheduled execution of the root jobnet, the execution schedule of the jobnet may be affected by events like the skipping process that occurs when suspension is released. For this reason, try to ensure that the suspend operation does not last for more than one generation.

The following table shows an example of a jobnet that was registered for planned execution on June 12.

Table 4-16: Jobnet registered for planned execution on June 12

	6/10	6/11	6/12	6/13	6/14	6/15	6/16	6/17	6/18	6/19	6/20
Jobnet	EN	EN	EN	SN	SS	SS	SS	SS	SS	SS	SS

Legend:

EN : Ended normally

SN : Scheduled next

SS : Schedule simulation

The JP1/AJS3 database stores information for the above jobnet that was registered on June 12, about the generation scheduled to execute on June 13. Unless you suspend this jobnet when it finishes executing on June 12, and release the suspension only after June 14, the schedule for June 13 remains. It will then be subjected to the skipping process after the suspension is released. In this case, a Skipped so not executed generation may be created, or an execution schedule may be skipped.

(6) Inheriting macro variables when adding an event job

Macro variables are passed on to a succeeding unit when the status of the succeeding unit changes from Waiting to execute to Now running.

The following paragraphs and figures show how macro variables are inherited when you change the lower-level definitions of the root jobnet while it is registered for execution.

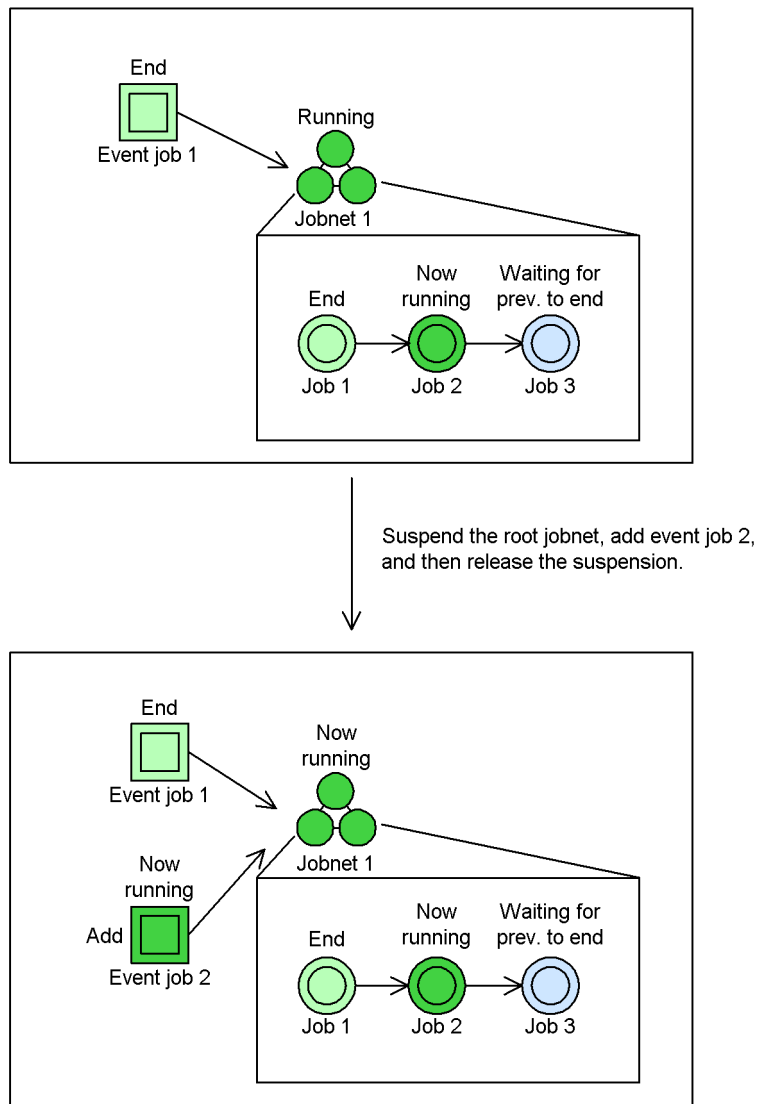
The way in which macro variables are inherited depends on the status of the unit that succeeds the added event job.

- When the succeeding unit is running

The following figure shows macro variable inheritance according to the status of

the succeeding unit.

Figure 4-47: Adding an event job that precedes a unit that is running

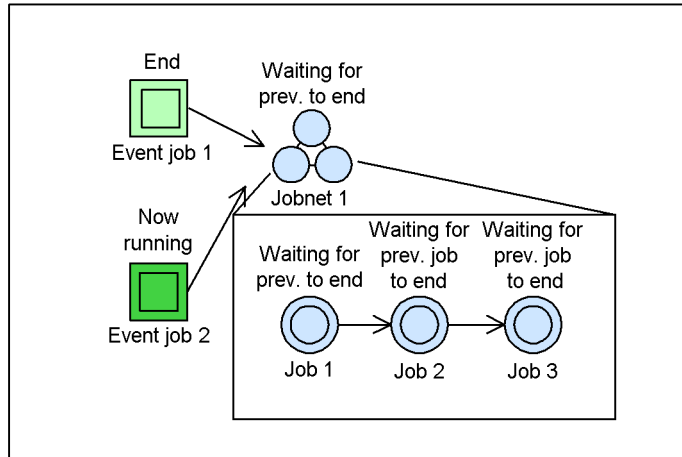


When job 2 ends, job 3 waits for event job 2 to end. The value of the macro variable that jobnet 1 inherits is created when jobnet 1 changes to Now running status. In Figure 4-45, the value of the macro variable that is created before suspending the root jobnet continues to be used under jobnet 1.

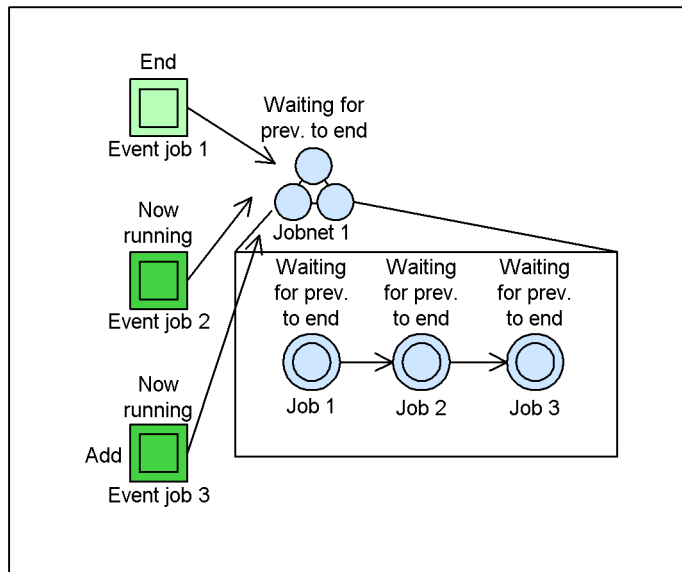
- When the succeeding unit is waiting for the previous unit to end

The following figure shows macro variable inheritance according to the status of the succeeding unit.

Figure 4-48: Adding an event job that precedes a unit that is waiting for the previous unit to end



Suspend the root jobnet, add event job 3, and then release the suspension.



4. Executing an Application

When event job 2 and event job 3 end, jobnet 1 is executed. The value of the macro variable that jobnet 1 inherits is created when jobnet 1 changes to Now running status. In Figure 4-46, the value of the macro variable that jobnet 1 inherits is obtained by merging the execution results of event job 1, the execution results of event job 2, and the execution results of event job 3 that was executed after the suspension was released.

Chapter

5. Job Execution Environments

This chapter describes how jobs are executed in JP1/AJS3 and the restrictions that can be placed on transferring jobs for execution.

- 5.1 How jobs are executed
- 5.2 Restricting job transfer
- 5.3 Job execution environments and system operation

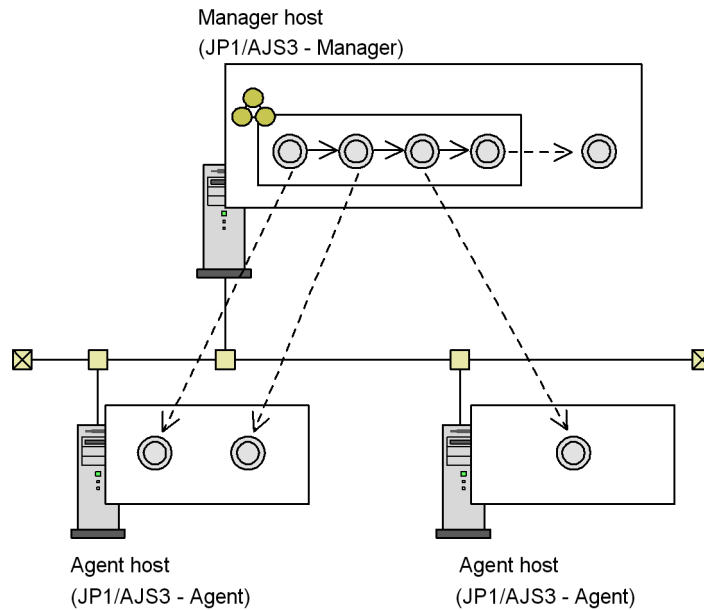
5.1 How jobs are executed

This section describes how jobs are executed in JP1/AJS3.

The environment for executing Unix jobs, PC jobs, event jobs, action jobs, and custom jobs in JP1/AJS3 consists of a manager host and agent hosts. Jobs are transferred from the manager host to an agent host for execution.

The following figure shows the flow of JP1/AJS3 job execution.

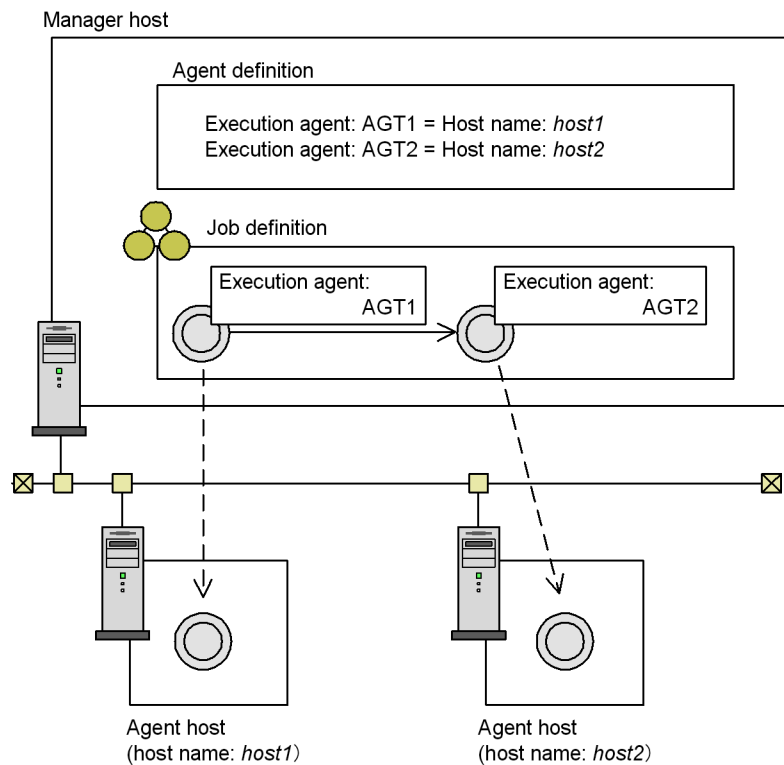
Figure 5-1: Flow of JP1/AJS3 job execution



To execute jobs using agent hosts, you must define the execution agents at the manager host. Then, in each job definition, specify the name of the agent that is to execute the job. Jobs are sent in turn to the specified agent host for execution.

The following figure shows the job execution process.

Figure 5-2: Process of job execution



You can distribute the processing load by using an *execution agent group* consisting of multiple execution agents.

5.1.1 Execution agents

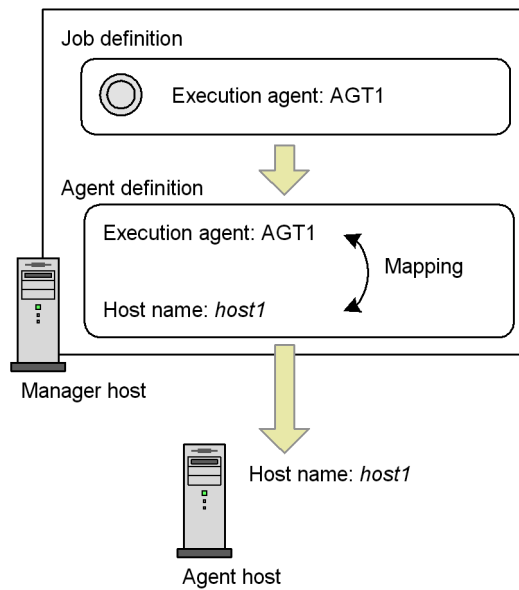
An *execution agent* is a logical name defined in JP1/AJS3 as a job execution host. JP1/AJS3 determines where each job is to be executed by mapping the execution agent name with the physical host name of an agent host that executes jobs. The job is then sent to that agent host.

Supplementary note

Queueless jobs do not use execution agents. To define a queueless job, specify the execution host in **Target agent** in the detailed definition of a job.

The following figure illustrates how execution agents are used.

Figure 5-3: Overview of execution agents



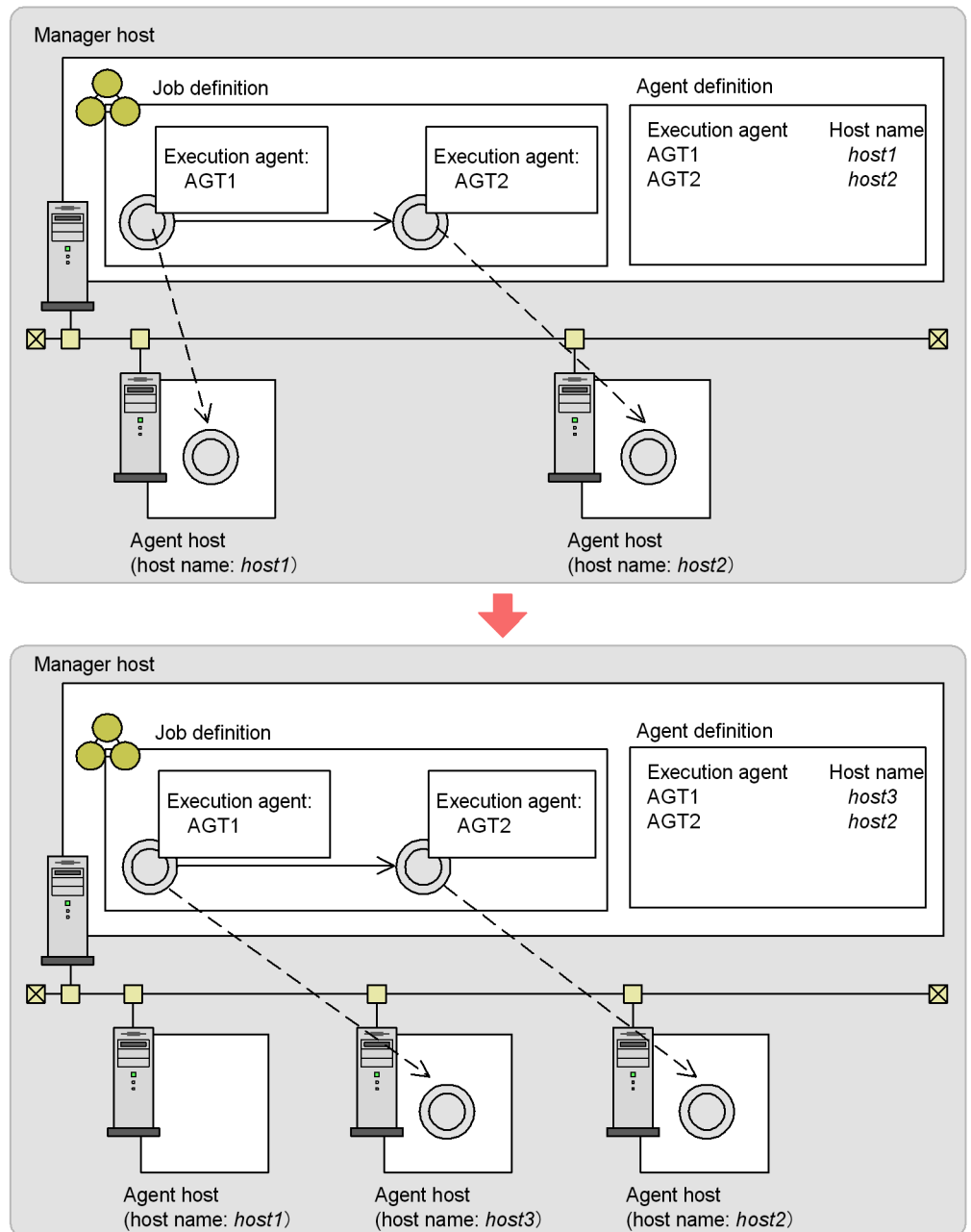
You can specify an execution agent for any of the following units:

- Jobnet
- Nested jobnet
- PC job
- Unix job
- Event job
- Action job
- Custom job

Note that this applies only when **Standard** is selected for **Exec. Service** in the detailed definition of a job.

You can change an agent host name, or use a different agent host, without changing the job definition. On the manager host, you simply change the physical host name that the execution agent is mapped to.

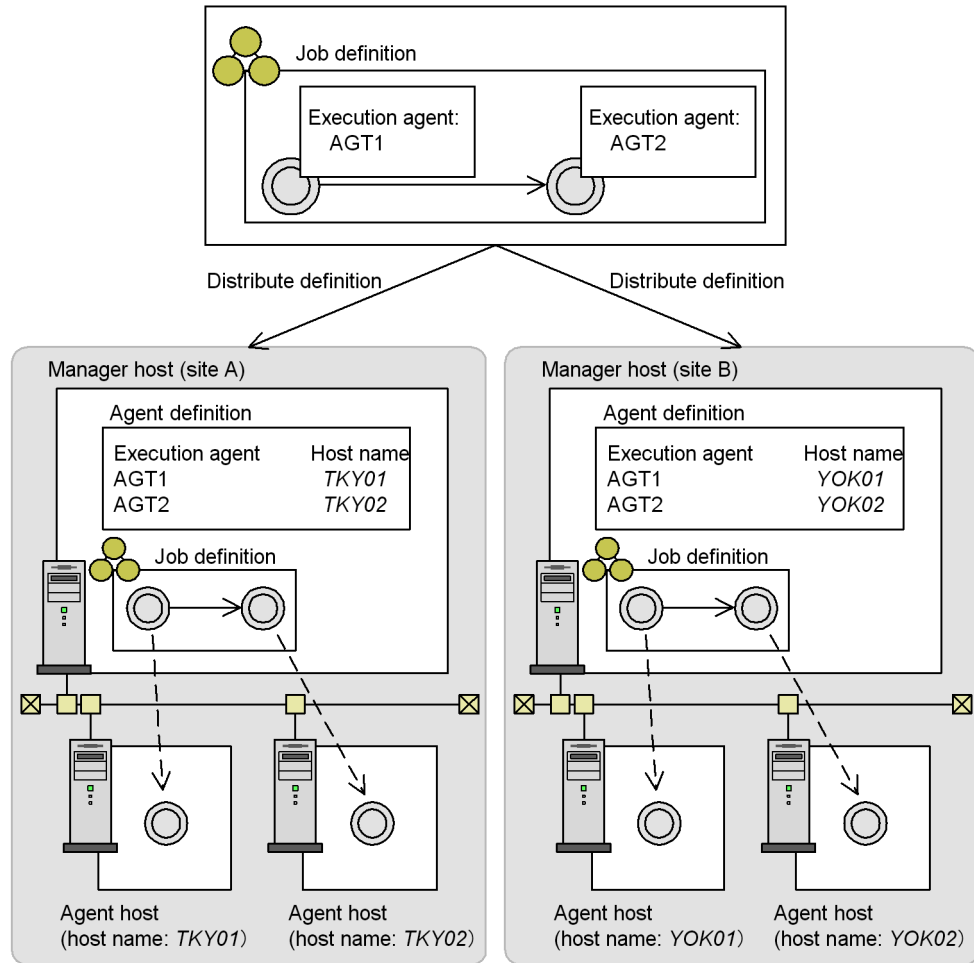
Figure 5-4: Using execution agents: Example 1



If an application needs to be executed at more than one site, you simply define the

execution agents separately at each manager host, without having to change the job definition.

Figure 5-5: Using execution agents: Example 2



To map an execution agent with the host name of an agent host, use the `ajsagtadd` command. To map a previously defined execution agent with a different agent host, use the `ajsagtal` command.

5.1.2 Executing jobs concurrently

You can restrict the number of jobs that can be executed concurrently in line with resource availability at the job (Unix jobs, PC jobs, event jobs, action jobs, and custom jobs other than QUEUE jobs) execution host. To do so, set the maximum number of concurrently executable jobs when you define the execution agent. This limits the

number of jobs that the agent host can run at the same time.

The maximum number of concurrently executable jobs is specified by period, in units of 30 minutes. You can set different limits for different time periods, helping to distribute the processing load.

When the maximum number of concurrently executable jobs is reached, subsequent jobs are made to wait in *Now queuing* status until the active jobs finish.

For event jobs, you cannot specify the maximum number of concurrently executable jobs.

For notes about concurrent execution, see 2.5.4 *Maximum number of concurrently executable jobs* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

(1) Specifying a maximum number of concurrently executable jobs

To specify a maximum number of concurrently executable jobs, set the `-c` option of the `ajsagtadd` command when you add an execution agent.

If you add an execution agent with the `ajsagtadd` command, but do not specify a maximum number of concurrently executable jobs, the default setting of `00:00-00:00=5` (maximum of five concurrent jobs, 24 hours a day) is assumed. You can change the setting using the `ajsagtalt` command. For the command syntax and an example of specifying a maximum number of concurrently executable jobs, see `ajsagtalt` in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(2) Load distribution based on maximum number of concurrently executable jobs

By grouping multiple execution agents in an execution agent group, you can distribute the processing load among the agents, based on the *agent usage rates* (ratio of active jobs to the maximum number of concurrently executable jobs). For details, see 5.3.9 *Distributing jobs among execution hosts (agent hosts)*.

Cautionary note

The maximum number of concurrently executable jobs you specify applies to each execution agent. If different execution agents are mapped to the same execution host, the sum of the maximum number of concurrently executable jobs specified for these execution agents is set for the jobs that will be executed on that execution host.

As an example, take the addition of an execution agent which has the local host name and whose maximum number of concurrently executable jobs is set to `00:00-00:00=10` (a maximum of 10 concurrent jobs, 24 hours a day). In this case, because the maximum number of concurrently executable jobs for the default execution agent is `00:00-00:00=5` (a maximum of five concurrent jobs,

24 hours a day), a maximum of 15 jobs could be executed concurrently.

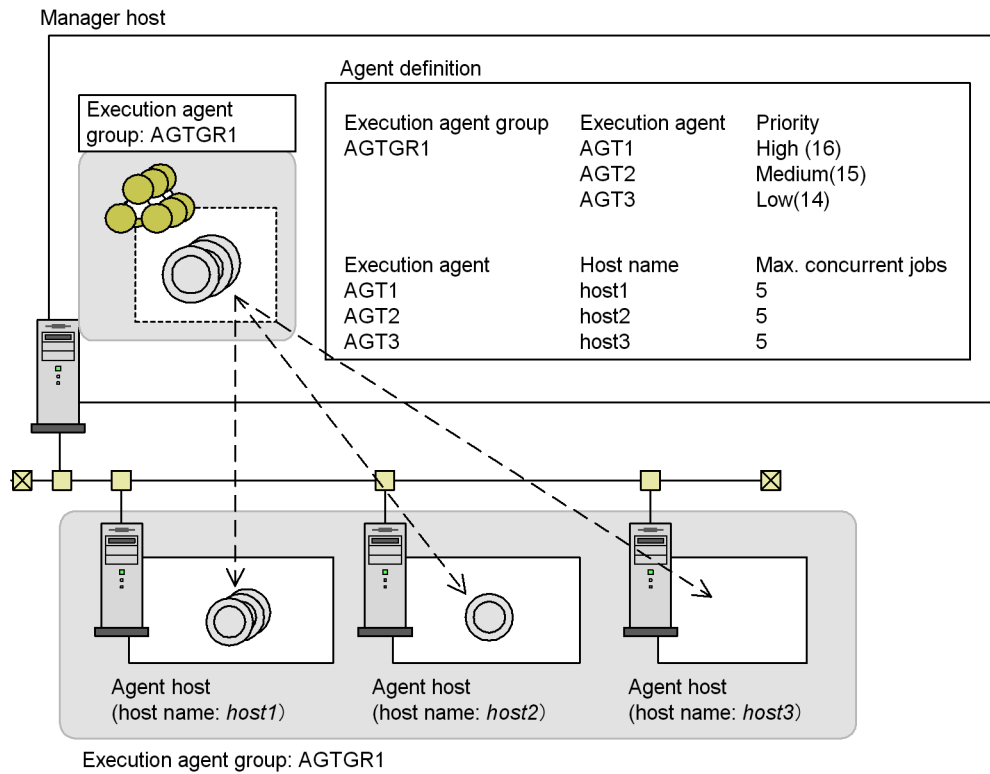
When you specify the maximum number of concurrently executable jobs, specify a value appropriate for the execution host.

5.1.3 Execution agent group

An execution agent group allows job processing to be distributed among multiple execution agents. Based on the priorities you set for the agents in the group, JP1/AJS3 determines which agent host to send each job to.

The following figure illustrates how jobs are executed using an execution agent group.

Figure 5-6: Overview of job execution using an execution agent group



You can specify an execution agent group for any of the following units:

- Jobnet[#]
- Nested jobnet[#]
- PC job

- Unix job
- Action job
- Custom job

Note that this applies only when **Standard** is selected for **Exec. Service** in the detailed definition of a job.

#

Event jobs do not support operations that use execution agent groups. If an event job without a specified execution agent is included in a root jobnet or nested jobnet for which an execution agent group is specified, JP1/AJS3 will attempt to use the agent group specified for the jobnet as the execution agent for the event job. If an execution agent with the same name as the agent group exists, the event job will be executed by that execution agent. If there is no execution agent with the same name as the agent group, an error occurs and the following message is output to the integrated trace log: KAVT0403-E The specified agent is not defined in the job execution environment. (host=*agent-host-name*, maintenance-information). Therefore, if you want to specify an execution agent group for a root jobnet or nested jobnet, make sure that an execution agent is explicitly specified for the event job in the jobnet.

5.1.4 Default execution agent

Agent information in which the local host is defined as the agent host is set by default in each manager host. This agent is referred to as the *default execution agent*.

The following table lists the information defined for the default execution agent.

Table 5-1: Information defined for the default execution agent

Item	Setting
execution agent	@SYSTEM
Execution host	Manager host
Maximum number of concurrently executable jobs	5 (00:00-00:00=5)
Job transfer restriction status	Effective
Comment	None set

Use the default execution agent to execute a job on the local host. To specify the default execution agent as the execution agent for a unit, either specify @SYSTEM for the execution agent in the unit definition or omit the execution agent specification. Even after migration of the unit definition to another manager host, the default execution agent will allow you to execute jobs on the destination host without changing the unit definition, improving unit definition portability.

Cautionary note

If you want to execute jobs by specifying the local host name for the execution agent in the unit definition, an execution agent with the same name as the local host name must separately be added in advance to the default execution agent. Use the `ajsagtadd` command to add the execution agent. For details about the `ajsagtadd` command, see *ajsagtadd* in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

If you add an execution agent that has the same name as the local host name, be careful in the handling of the maximum number of concurrently executable jobs defined for the execution agent. The maximum number of concurrently executable jobs specified for an execution agent applies to that specific execution agent. As an example, take the addition of an execution agent which has the local host name and whose maximum number of concurrently executable jobs is set to `00:00-00:00=10` (a maximum of 10 concurrent jobs, 24 hours a day). In this case, because the maximum number of concurrently executable jobs for the default execution agent is `00:00-00:00=5` (a maximum of five concurrent jobs, 24 hours a day), a maximum of 15 jobs could be executed concurrently.

You can change the information defined for the default execution agent by using the commands for working with execution agents. For details, see 1.5.5 *Commands used to operate execution agents* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*. Note that the execution agent commands include functionality that does not apply to the default execution agent. The following table shows which operations can be targeted to the default execution agent.

Table 5-2: Applicability of execution agent commands to the default execution agent

Command	Purpose	Applicability
ajsagtadd	Define the default execution agent as a connected execution agent when adding an execution agent group.	Y
	Add a default execution agent.	N
ajsagtalt	Change the execution host.	N
	Change the maximum number of concurrently executable jobs.	Y
	Change the job transfer restriction status.	Y
	Add the default execution agent as a connected execution agent in an execution agent group.	Y
	Delete the default execution agent from the connected execution agents when connected to an execution agent group.	Y

Command	Purpose	Applicability
	Change the priority of the default execution agent when connected to an execution agent group.	Y
ajsagt del	Delete the default execution agent.	N
ajsagtshow	Display the job execution status or other information about the default execution agent.	Y
ajsagtprint	Output definition information about the default execution agent.	Y

Legend:

Y: Applies to the default execution agent.

N: Does not apply to the default execution agent.



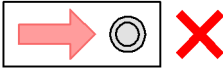

5.2 Restricting job transfer

You can restrict the reception and distribution of a job for each execution agent in either of two ways: by having the system issue an error when the agent receives a job, or by holding distribution. These two actions are described below. Event jobs can still be monitored when job transfer is restricted.

- Issuing an error on job receipt
This action stops jobs from being received and ends received jobs abnormally.
- Holding distribution
This action holds job transfer until the held status is released.

The following table describes the operations that occur for each job transfer restriction status at an execution agent.

Table 5-3: Operations for each job transfer restriction status at an execution agent

Status	Status returned by the ajsagshow command	Meaning
Effective 	Ef	<ul style="list-style-type: none"> • Jobs can be received. • Jobs can be sent to the agent host.
Ineffective 	In	<ul style="list-style-type: none"> • Jobs cannot be received (they are placed in <i>Failed to start</i> status when executed). • Jobs can be sent to the agent host.
Hold 	Hd	<ul style="list-style-type: none"> • Jobs can be received. • Jobs cannot be sent to the agent host (they remain on JP1/AJS3 - Manager).
Blockade 	B1	<ul style="list-style-type: none"> • Jobs cannot be received (they are placed in <i>Failed to start</i> status when executed). • Jobs cannot be sent to the agent host (they remain on JP1/AJS3 - Manager).

The following table describes the job status transitions that occur for each job transfer restriction status at an execution agent.

Table 5-4: Job status transitions for each job transfer restriction status at an execution agent

Job transfer status	Job status transition	Event job status transition
Effective	<i>Now queuing</i> → <i>Now running</i> → <i>Ended</i>	<i>Now queuing</i> → <i>Now running</i> → <i>Ended</i> (Transfer of event jobs cannot be restricted.)
Ineffective	<i>Failed to start</i> (<i>Now queuing</i> jobs have the same status as <i>Effective</i> .)	
Hold	<i>Now queuing</i>	
Blockade	<i>Failed to start</i> (<i>Now queuing</i> jobs have the same status as <i>Hold</i> .)	

5.3 Job execution environments and system operation

5.3.1 User account for job execution

In JP1/AJS3, jobs are executed under the OS user account of the agent host. To use this account, user mapping must be defined on the agent host where the job is to be executed. When a job is executed, the JP1 user who registered the jobnet for execution is converted to the OS user according to the user mapping definitions. The job is then executed under the OS user's permissions. The resources referenced or updated by the job process are dependent on the OS security control.

The user account is referenced each time a job is executed. Any changes to the mapping definitions affect jobs executed after the changes are made.

In the Windows version of JP1/AJS3, when a job is executed under a user account other than the account from which the JP1/AJS3 service was started, JP1/AJS3 acquires the user information needed to start the job process. To acquire information about a user, an *access token* is required. The access token contains user information such as the security groups to which the user belongs and the user's access permissions.

Using Win32 API functions, JP1/AJS3 obtains an access token each time it starts a job, and releases the access token when the job completes execution. When an error occurs in one of these functions, the job is placed in *Failed to start* status if the access token could not be acquired, or in *Ended abnormally* status if the access token could not be released.

You can keep and reuse the information contained in an access token if the domain name, user name, and password remain the same.

By reusing access tokens, you can minimize the number of times tokens are acquired and released, thus avoiding the temporary errors caused by the Win32 API functions. This results in fewer jobs ending abnormally. For details, see *6.2.17 Reusing access tokens for job execution* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1*.

5.3.2 OS user environment used for job execution

The OS user environment used to execute jobs is platform-dependent.

When a job is executed on a Windows agent host, the system environment variable normally takes precedence over the user profile information. However, you can enable the user profile information by changing the environment setting for job execution control. For details, see *6.2.16 Executing a job that requires a user profile* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1*.

When a job is executed on a UNIX agent host, the local login script of the OS user mapped from the JP1 user is read. The OS user's login shell is set as the execution shell

of the job, unless otherwise specified in the first line of the script file or command statement.

The OS user environment is read each time a job is executed. Any changes to the definitions affect jobs executed after the changes are made.

5.3.3 Access permission to executable files and script files

The setting of the access permission for accessing executable files and script files for a job differs depending on whether the host (agent) on which the job is to be executed is a Windows or a UNIX host.

Windows

For a Windows execution host (agent), you do not need to set access permission for the executable files for the OS user who executes the job. Set the access permission for the executable files only in the JP1/AJS3 service account.

UNIX

For a UNIX execution host (agent), set the access permission for the script file for the OS user who executes the job.

For details about access permissions to files used for UNIX jobs, also see 5.3.4 *Checking file access permission at job execution (Unix jobs)*.

5.3.4 Checking file access permission at job execution (Unix jobs)

For Unix jobs, you can specify the scope of the file access permission check performed when a job is executed. You can choose to check the file permissions only, or the access control list (ACL) and secondary group settings as well.

An access permission check is performed on the following files:

- Script file
- Environment variable file
- Standard input file
- Standard output file
- Standard error output
- Transfer destination file

You can set any of the following three methods of checking access permissions:

- Check only the file permission setting for the file in question.
- For a script file, check the ACL and secondary group settings in addition to the file permission. For all other types of files, check the file permission only.
- For all files, check the file permission and the ACL and secondary group settings.

The default method is to check file permissions only.

The check method is set separately for each agent host. For details on the setting procedure, see *14.2.18 Enabling the file access permission check for the ACL and secondary group settings during job execution* in the *Job Management Partner 1/ Automatic Job Management System 3 Configuration Guide 1*.

The following table describes the relationships between the type of user accessing the file, the file category, and the check method when the ACL and secondary group settings are checked in addition to the file permission settings.

Table 5-5: File access permission check by user category and file category

User category		File category		
		Files provided by JP1/AJS3 ^{#1}	User files ^{#2}	
			No ACL	With ACL
Superuser		No check needed	No check needed	No check needed
Others	No secondary groups	File permission	File permission	ACL
	With secondary groups	File permission	File permission and secondary groups	ACL and secondary groups

Legend:

ACL: Access control list

#1

Files and directories provided by JP1/AJS3 products.

#2

Files and directories for user resources, specified by the user when executing a job or command.

Cautionary notes

- Access permissions to files and directories provided by JP1/AJS3 products are not checked.
- If the agent host is running JP1/AJS version 08-10 or earlier, only the file access permissions are checked. The ACL and secondary group settings are not checked.
- If you change the file access permission check method, jobs that were executable under the previous check method might fail to start. When setting the check method, consider whether the file permission settings differ from those in the ACL and secondary group settings, and make sure that the check

will not impact on job startup.

5.3.5 HP-UX OS user groups

When an OS user who executes jobs belongs to more than one group, and you want to enable access to all those groups, you must set `/etc/logingroup`; otherwise, only the group ID defined in `/etc/passwd` can be used.

To enable access to multiple groups, copy the `/etc/group` group definition into `/etc/logingroup`, or make a symbolic link between `/etc/logingroup` and `/etc/group`. For details, see your OS documentation.

Cautionary note

To enable access to multiple groups when the process of a JP1/AJS3 service accesses a job result file or temporary file, perform the setting described in *14.2.18 Enabling the file access permission check for the ACL and secondary group settings during job execution* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1*.

If you do not perform this setting, access from a secondary group based on the entries in `/etc/logingroup` is possible only for a job process. Entries in `/etc/logingroup` do not apply to a JP1/AJS3 service process accessing a job result file or temporary file. If a JP1/AJS3 service process is unable to access a result file, change the permission of the directory containing the result file to a permission that will allow access from a secondary group. Alternatively, change the group that owns the directory or the specified result file from the secondary group to the primary group of the user executing the job. In the case of a temporary file, change the permission of the work directory to a permission that will allow access from a secondary group.

For details, see *2.7.2(2) Executing a standard job, action job, or custom job results in an abnormal end* in the manual *Job Management Partner 1/Automatic Job Management System 3 Troubleshooting*.

5.3.6 Login shell at job startup (Unix jobs and action jobs)

When a job is started, JP1/AJS3 executes the login shell (contents of the `/etc/passwd` file) of the OS user at the agent host. If no login shell is defined, `/bin/sh` is executed. JP1/AJS3 supports the login shells `sh`, `csh`, and `ksh`. See the following precautions on using shells other than these three.

(1) Precautions for Unix jobs

If you use a login shell other than `sh`, `csh`, or `ksh`, the login script might not be executed under certain OSs. Consequently, since the environment variable settings for the user executing the job are not applied, jobs might terminate abnormally or return unexpected results.

To avoid such problems, specify the environment variable settings required for

executing a job script. To check whether the environment variables needed for job execution are set properly, create a shell script that writes the execution results returned by the `env` command to a file, and then look at the contents of the file that is output when you execute this script as a job.

(2) Common precautions for Unix jobs and action jobs

Some shells, such as `bash`, do not initialize signal processing to the default status. As a result, jobs might terminate abnormally or return unexpected execution results. In this case, use the `sh`, `csh`, or `ksh` shell instead.

Signal processing can be initialized to the default status by changing a setting as described in *14.2.17 Applying SIG_DFL (setting for receiving the SIGCHLD signal) to a child process started from JP1/AJS3* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1*. If you are performing a new installation of JP1/AJS version 08-00 or later, there is no need to change the setting. In these versions, jobs are executed with the default setting (`SIG_DFL`), which allows child processes to receive SIGCHLD signals.

5.3.7 Value of umask set for the standard output file and the standard error output file (UNIX only)

When the execution host (agent) is a UNIX system, by default the standard output file and standard error output file are created using the `umask` value of the user who started the JP1/AJS3 service. To allow registered users to set their own `umask` value for these files, create a job execution environment settings file for each user.

The procedure for using a job execution environment settings file and the file specifications are as follows:

Using a job execution environment settings file

On the agent host, create the following file for each JP1 user:

- File name


```
/etc/opt/jp1ajs2/conf/profiles/JP1-user-name/  
jajsJobProfile
```
- File contents


```
umask=3-digit-octal-number
```

Specification of a job execution environment settings file

The `umask` value specified in a job execution environment settings file affects only the standard output file and standard error output file. Files created by job processes are not affected. When standard output or standard error output is performed for an existing file, the access permission of the existing file is kept unchanged regardless of whether the output data overwrites existing data or is added to the file.

5.3.8 Monitoring the status of registered jobs

The manager host monitors the agent hosts and polls the jobs that are registered for execution. In JP1/AJS3, the status of each job is automatically reported by the agent host to the manager host. If the manager host fails or if a communication error occurs between the manager host and the agent host that is executing a job, the job status might not be reported correctly. JP1/AJS3 performs monitoring to enable recovery from such errors.

(1) *Monitoring active jobs*

The manager host polls active jobs at five-minute intervals. As each job finishes, the manager host places it in *Ended* status when notified by the agent host that the job has ended. If the manager host fails to receive notification due to a temporary communication error or other fault, it detects that the job has ended by polling its status. If polling fails due to a communication or other error, and the status of the active job cannot be verified for 12 to 30 minutes (depending on the monitoring interval set for the agent host and the execution start time of the job), the manager host changes its status. In a configuration where jobs are executed by multiple agent hosts, the manager host checks the status of active jobs on a host-by-host basis. This means that the frequency of communication increases in proportion to the number of agent hosts that are being managed for failures.

If the status of a job defined in a jobnet cannot be verified, the manager host places it in *Killed* status and sets its return code to -1. If the job was executed by the `jpqjobsub` command, its status changes to the status specified in the `-rs` option. For details, see `jpqjobsub` in 3. *Commands Used for Special Operation* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2*.

At this time, the following message appears in the integrated trace log:

```
KAVU4534-W No response was received from the agent (agent-host-name), so the
status of job (job-number) was changed to recovered (status).
```

The manager host monitors the status of standard jobs (other than QUEUE jobs executed from another system), action jobs, and custom jobs.

(2) *Monitoring execution hosts (agents)*

If the manager host gets no response from the agent host when it attempts to register a job for execution, it recognizes that an error has occurred on the agent host or that the agent host has stopped. On detection of failed or stopped status, the manager host continues to poll the agent host at five-minute intervals. While the agent host remains in failed or stopped status, jobs are queued and wait for the agent host to recover. When the manager host detects that the agent host has recovered (from failed or stopped status), it resumes registering jobs for execution on that host. However, if the execution host has not recovered within 10-15 minutes of the initial registration failure (depending on the monitoring interval set for the agent host and the time of the job

registration request), the queued jobs are placed in *Failed to start* status. Because the manager host checks the status of the agent hosts individually, the frequency of communication increases in proportion to the number of agent hosts that are being managed for failures.

At this time, the following message appears in the integrated trace log:

```
KAVU4593-W An executable agent does not exist.
```

Applicable jobs here are standard jobs (jobs other than QUEUE jobs running in another system), action jobs, and custom jobs.

(3) **Monitoring jobs on another system**

To check the status of a job registered for execution on another system (JP1/NQSEEXEC or JP1/OJE, for example), the manager host polls the job at five-minute intervals. If no response is received for approximately one hour or longer, the following error message is output to the integrated trace log and the job is placed in abnormally terminated status:

```
KAVU6218-W In the job status notification process, the job information was not acquired because the error occurred during TCP/IP communication. But the job might have ended normally. (manager-descriptor, job-number)
```

The other system might not support the functionality for notifying the manager whenever a status change occurs in a job. In this case, it might take as long as five minutes for the manager host to acquire the job status through polling. See the relevant system documentation to find out whether the other system supports the status notification functionality.

The manager host does not perform five-minute polling of a submit job that was registered by the `jpqjobsub` command for execution on another system. To check the job status, use the `jpqjobget` command.

Note on linking with JP1/NQSEEXEC

Notification of job termination is not performed when a job execution request is sent from JP1/AJS3 to JP1/NQSEEXEC version 05-20 or earlier. In this case, JP1/AJS3 polls the job executed by the linked JP1/NQSEEXEC at five-minute intervals. Consequently, it might take as long as five minutes for the job's status to change after completion. Be aware of how this will affect the execution monitoring interval if there are succeeding jobs.

The JP1/AJS3 notification functionality is supported in JP1/NQSEEXEC version 06-00 and later. When execution by JP1/NQSEEXEC finishes, the job's status is immediately reported to JP1/AJS3.

If you are using JP1/NQSEEXEC version 05-20 or earlier, the length of time taken to detect job termination could have a major impact on operations. We

recommend that you upgrade to JP1/NQSEXEC 06-00 or later, or migrate to JP1/AJS3.

(4) Error detection and recovery time at a job execution host

If a communication error or failure occurs on the agent host executing a job (standard job, action job, or custom job), JP1/AJS3 does not immediately assume that the job has terminated abnormally. Instead, the manager host waits for a certain amount of time and then retries, waiting for the system error or communication error on the agent host to be corrected. This grace time prevents disruption of job processing should a temporary and recoverable error occur.

For some applications, you might want errors to be promptly detected and speedily corrected, rather than waiting for recovery. Errors can be rapidly detected if you reduce the TCP/IP communication time or recovery wait time. For details on how to reduce the time until an error is detected, see *6.2.12 Changing the wait time for recovery when an agent has failed* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or *14.2.12 Changing the wait time for recovery when an agent has failed* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

The time taken to detect an error on the agent host differs at job transfer and at job execution. This is explained next.

(a) Error detection and recovery time at job transfer

The manager host uses TCP/IP communication to send jobs to an agent host. If the agent host has not started or if a network error occurs, a TCP/IP connection error results. However, because of the retry interval allowed, it might take as long as five minutes for the manager host to detect the error. The agent host where the connection error occurred is assumed to have failed. The manager host does not attempt to transfer subsequent jobs via TCP/IP while the agent host remains in failed status.

If the agent host is in failed status, all jobs wait for recovery for the specified recovery wait time (default 10 minutes). During this time, jobs are placed in *Now queuing* status (or *Waiting to execute* status in the case of a submit job). If the agent host has not recovered from the error when the wait time elapses, all jobs are placed in *Failed to start* status. The length of time from when a job is registered for execution until it enters *Failed to start* status depends on whether TCP/IP communication is performed, as follows:

- Job transferred to the agent host before error detection (TCP/IP communication is performed)

Communication time on the TCP/IP connection (approx. 5 minutes maximum)^{#1}
 + recovery wait time at the agent host (10 minutes) = 15 minutes 10 seconds maximum

- Job waiting for transfer to the agent host after error detection (TCP/IP communication is not performed)

Recovery wait time at the agent host (10 minutes)

(b) Error detection and recovery time at job execution

On receiving notification from the agent host that a job has started execution, the manager host changes its status to *Now running* and starts checking the job status by polling the agent host at the set interval (default 300 seconds (5 minutes)). At this time, TCP/IP communication is used to pass information between processes. If the agent host has not started or if a network error occurs, a TCP/IP connection error results. However, because of the retry interval allowed, it might take as long as 310 seconds (5 minutes 10 seconds) for the manager host to detect the error.^{#1}

If the connection error is detected within the recovery wait time set for the agent host (default 10 minutes), the manager host resumes polling. If the connection error remains undetected until after the wait time elapses, the manager host changes the job status to *Killed*.^{#2} Therefore, it takes roughly 12 to 30 minutes in total for a job error to be detected from the time it actually occurred on the agent host.^{#3}

#1

With TCP/IP connection, retry is performed by default. Thus, the following time settings apply from the time a connection error occurs until connection timeout:

- Timeout value for TCP/IP connection
Default 90 seconds
- Retry count for TCP/IP connection
Default 2 retries
- Retry interval for TCP/IP connection
Default 20 seconds

Even if a connection error occurs immediately, two retries are attempted at the default 20-second interval. Therefore, the communication time could be from roughly 40 seconds to as long as 310 seconds (5 minutes 10 seconds). For details on setting a retry interval and retry count for TCP/IP connection, see 6.2.8 *Changing the interval and number of retry attempts when a TCP/IP connection error occurs* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or 14.2.8 *Changing the interval and number of retry attempts when a TCP/IP connection error occurs* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

#2

If the job was executed by the `jpqjobsub` command, its status changes to the status specified in the `-rs` option. For details, see `jpqjobsub` in *3. Commands Used for Special Operation* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2*.

#3

With the default settings, the total time is calculated as follows:

Approximate total time to detect an error =
 (agent monitoring interval x 2 retries)
 + (communication time x 3 times)
 + time from error occurrence until the job status is first verified

When monitoring active jobs, the manager host checks the status of one job with each poll. If multiple jobs are being executed on an agent host, one job will be killed on detection of an error. The manager host then starts polling the next job. The time taken to detect the error from the time a poll starts is the same for every job.

For example, suppose three jobs are running and it takes 20 minutes to detect an error in one job. To detect the error and kill all three jobs, it would take at least 60 minutes.

Depending on the system, rather than waiting for recovery after a communication error in this way, it is sometimes better to kill all jobs currently running on the agent host, enabling immediate detection of the error and rapid recovery. For details about implementing this setup, see *6.2.20 Placing all running jobs in an end status when a communication error occurs* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or *14.2.19 Placing all running jobs in an end status when a communication error occurs* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

5.3.9 Distributing jobs among execution hosts (agent hosts)

(1) Distributing jobs among execution agents in an execution agent group

You can connect multiple execution agents with assigned priorities within an execution agent group. Jobs can then be distributed to the execution agents in accordance with the specified priorities. This functionality applies to Unix jobs, PC jobs, action jobs, and custom jobs. Queueless jobs cannot be distributed among execution agents.

Specify the execution agent priorities to suit your system operation. If you assign different priorities, each job will be sent for execution to the execution agent that has

the highest priority. When that agent's usage rate[#] reaches 100%, the next job is sent to the agent that has the next highest priority. For example, if the execution agent with the highest priority is already fully occupied or is in failed or stopped state and cannot run the job, the execution agent with the next highest priority can run it instead. Setting different priorities is useful for this type of system operation.

If you set the same priority for each execution agent, each job will be sent to the execution agent that has the lowest agent usage rate. If no particular execution agent takes precedence, jobs are distributed evenly among the execution agents. Setting identical priorities is useful for this type of load-balancing system.

#

Agent usage rate is the ratio of active jobs to the maximum number of concurrently executable jobs at an execution agent. It is given by the following equation, rounded to four decimal points:

Agent usage rate = *number of active jobs* / *maximum number of concurrently executable jobs*

Cautionary note

When two or more execution agents have the same priority and usage rate, the execution agent with the highest agent ID[#] is selected by default to execute the job. For example, if the jobs are short and quickly completed, the execution agent's usage rate hardly increases and differs only slightly from that of other execution agents.

Consequently, because jobs are always assigned to the execution agent having the highest agent ID, that agent executes more jobs than the others, resulting in asymmetric system usage. In such situations, you can distribute jobs more uniformly if, instead of the agent ID method, you set a job distribution method that prioritizes the execution agent that has not been used for the longest time.

For details, see 6.2.9 *Setting the method for determining the agent host to which a job will be delivered* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or see 14.2.9 *Setting the method for determining the agent host to which a job will be delivered* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

#

Agent IDs are assigned internally as each execution agent is registered, in ascending order of the unused IDs. When execution agent hosts are displayed by the `ajsagtshow` command, they are listed in ascending order of their agent IDs.

For the `ajsagtshow` command syntax, see *ajsagtshow* in 2. *Commands* in

the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(2) Error detection and job distribution to execution hosts (agent hosts)

In the following two cases, there may be a delay of up to 15 minutes in distributing jobs to execution hosts that are operating normally. During this period, the jobs remain in *Now queuing* status:

- When a failure occurs simultaneously on three or more agent hosts
- When three or more agent hosts have stopped

In such cases, restart the affected execution agent hosts.

5.3.10 Killing jobs

JP1/AJS3 kills an active job on receipt of a request to forcibly terminate a job or to forcibly stop the JP1/AJS3 services.

(1) Windows execution host

Forcibly terminating an active job kills the following processes on a Windows execution agent host:

- Process started directly from JP1/AJS3 (job)
- Process started by the `jp1exec` command (command process)

To kill these processes, the Windows OS calls the Win32API `TerminateProcess` function. For the function specifications, see your Windows documentation. A child process started from a user program cannot be killed.

(2) UNIX execution host

Forcibly terminating an active job kills the following processes on a UNIX execution agent host:

- Process started directly from JP1/AJS3 (job)
- Process started from a user program (excluding a process whose process group is modified by a `setpgrp` system call or other means)

To kill these processes, the UNIX OS sends a SIGKILL signal to the process group. For details about process groups and SIGKILL, see the documentation relating to UNIX process control.

(3) Killing a job created by JP1/Script

Job processes created by JP1/Script (process having the extension `.spt`) are killed from JP1/AJS3 using `SPTHTerminate`, a script control interface of JP1/Script. For details about `SPTHTerminate`, see the manual *Job Management Partner 1/Script (for Windows(R))*. This interface also terminates child processes started from the JP1/Script

process (.spt).

Which child processes are killed depends on whether they were started from the JP1/Script process (.spt) or from a different process, as described in the following table:

Table 5-6: Range of killed processes

Pattern	Process call sequence	Range of terminated processes when the JP1/AJS3 process (no. 1 in the call sequence) is killed
Pattern 1	<ol style="list-style-type: none"> 1. JP1/AJS3 process 2. JP1/Script process 3. Process other than the JP1/Script process 	All processes up to no. 3
Pattern 2	<ol style="list-style-type: none"> 1. JP1/AJS3 process 2. JP1/Script process 3. JP1/Script process 4. Process other than the JP1/Script process 	All processes up to no. 4
Pattern 3	<ol style="list-style-type: none"> 1. JP1/AJS3 process 2. JP1/Script process 3. Process other than the JP1/Script process 4. Process other than the JP1/Script process 	All processes up to no. 3 (no. 4 is not terminated)

If the JP1/Script process has not ended within 30 seconds of the forced termination request from JP1/AJS3, it is killed by calling the Win32API `TerminateProcess` function. As `TerminateProcess` does not kill child processes, only the JP1/Script process (no. 2 in the above patterns) is terminated.

5.3.11 Stopping JP1/AJS3 services and job execution control

When stopping JP1/AJS3 services, you can either kill active jobs or wait for them to terminate. The default stop method is to kill active jobs.

Kill active jobs and stop JP1/AJS3 services

In this method, job execution control kills active jobs and then stops the JP1/AJS3 services. Jobs running on other agent hosts are not affected.

Stop JP1/AJS3 services after active jobs terminate

In this method, job execution control waits until all jobs including those running on other agent hosts have terminated, and then stops the JP1/AJS3 services.

For details on how to stop JP1/AJS3 services, see *7.1.2 Stopping JP1/AJS3 services manually* in the *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*. For details on forced termination by job execution control, see *5.3.10 Killing jobs*.

You can temporarily change the method used to stop JP1/AJS3 services by specifying

the required option in the `jajs_spmc_stop` command arguments. For details, see *7.2.2 Temporarily changing the end mode of JP1/AJS3* in the *Job Management Partner 1/Automatic Job Management System 3 Administration Guide*.

5.3.12 Group ID for job execution (UNIX only)

When JP1/AJS3 services are started automatically, jobs executed from a user account that has root privileges inherit the status of the root account from which the JP1/AJS3 services were started. That is, jobs executed from the root account are executed under the group ID of the root user account from which JP1/AJS3 services were started automatically, not the group ID set when the executing user logged in as the root user.

Chapter

6. Monitoring Applications

Once a jobnet has been registered for execution according to a defined schedule, you can monitor the execution status and check the execution results of the jobnet or job.

This chapter describes how to monitor a jobnet or job that has been registered for execution.

6.1 Monitoring execution status and checking execution results

6.1 Monitoring execution status and checking execution results

Once you have registered jobnets for execution according to a defined schedule, you can monitor the execution status and check the execution results of any jobnet.

You can use JP1/AJS3 - View or various commands to check the execution status and execution results for a jobnet, or for each of the jobs defined in a jobnet. For details on the commands you can use to check the execution status and execution results, see the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

The different execution status levels you may encounter are explained below, followed by an explanation of how to monitor jobs and jobnets using JP1/AJS3 - View.

6.1.1 Status levels of jobnets, jobs, and jobnet connectors

The following table lists the statuses that apply to jobnets, jobs, and jobnet connectors.

Table 6-1: Status levels of jobnets, jobs, and jobnet connectors

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connector	
	Root	Nested			
Not sched. to exe. (no plan)	Y	Y	Y	Y	No execution schedule is defined.
Wait for start time ^{#1} (time-wait)	Y	Y	N	N	Waiting for the execution start time to arrive. ^{#2} If synchronized with a jobnet connector, the jobnet is waiting for the start time of the jobnet connector.
Wait for prev. to end (term-wait)	N	Y	Y	Y	Waiting for the preceding job or jobnet to end. Even if no preceding unit exists, the unit will be placed in this status if any of the following conditions applies: <ul style="list-style-type: none"> • The upper-level jobnet is in wait state. • The scheduler service is suspended. • The execution process for the unit has not started even though the start condition of the upper-level jobnet has been satisfied.

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connect or	
	Root	Nested			
Being held ^{#1} (holding)	Y	Y	Y	N	The job or jobnet is being held.
Waiting to execute (exec-wait)	N	N	Y	N	Waiting for the job's submit process to complete. This status applies after a job has been submitted by scheduler control, until it is queued by job execution control.
Not executed + Ended (not scheduled) ^{#3} (unexec)	N	Y ^{#4, #5}	Y ^{#4, #5}	Y	The job or jobnet was not executed because the preceding unit ended abnormally.
-- (unexec-W)	N	Y	Y	N	The job or jobnet ended with a warning, and the attempted rerun was unsuccessful.
Bypassed ^{#1} (bypass)	N	Y	Y	Y	The job or jobnet was not executed because the schedules of the nested jobnet and the upper-level jobnet did not match, or execution was prohibited, for example.
Now running ^{#1} (running)	Y	Y	Y	Y	The job or jobnet is being executed. The jobnet connector is waiting for the connection-destination jobnet to end.
Now queuing ^{#1} (queuing)	N	N	Y	N	The start condition has been satisfied and the job is waiting to be executed.
Running + Abend ^{#1} (AB-cont)	Y	Y	N	Y	Jobs or nested jobnets running concurrently in the same jobnet ended abnormally. An error occurred in the connection-destination jobnet of the jobnet connector.
Running + Warning ^{#1} (WA-cont)	Y	Y	N	Y	Jobs or nested jobnets running concurrently in the same jobnet ended with a warning. The connection-destination jobnet of the jobnet connector ended with a warning.

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connector	
	Root	Nested			
Ended normally ^{#1} (normal)	Y	Y	Y	Y	All jobs and jobnets ended normally. The connection-destination jobnet of the jobnet connector ended normally.
Normal end + False (normal-false)	N	N	Y	N	The judgment job ended normally, although the result of the preceding unit did not match the judgment condition.
Ended with warning ^{#1} (warning)	Y	Y	Y	Y	The job ended with a warning. The jobnet contains a job that ended with a warning. The connection-destination jobnet of the jobnet connector ended with a warning.
Ended abnormally ^{#1} (abnormal)	Y	Y	Y	Y	The job ended abnormally. The jobnet contains a job that ended abnormally. The jobnet connector could not connect to the connection-destination jobnet due to an invalid definition.
-- (abnormal-WR)	Y	Y	Y	N	The job or jobnet ended with a warning, and the subsequent rerun ended abnormally.
Skipped so not exe. ^{#3} (exec-deffer)	Y	Y	N	N	The jobnet was not executed because, for example, the JP1/AJS3 - Manager service was not active at the job's scheduled start time, or the base time passed while the root jobnet was being held. ^{#4,#6} A jobnet enters <i>Skipped so not exe.</i> status in the following cases: <ul style="list-style-type: none"> The jobnet's timeout period elapses while the root jobnet is in <i>Wait for start time</i>, <i>Being held</i>, or <i>Wait for start cond.</i> status.^{#7} Schedule skip is selected in Schedule option, and the start time of the next generation arrives before termination of the preceding generation^{#8}.

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connect or	
	Root	Nested			
					<ul style="list-style-type: none"> • Schedule skip is selected in Schedule option, and a jobnet that is running in immediate execution mode is again registered for immediate execution. • Multi-schedule is selected in Schedule option, but concurrent execution is not enabled for the jobnet. The current generation fails to end, and the root jobnet's timeout period expires before the next generation starts. • Concurrent execution is not enabled for the jobnet. The start condition is satisfied while a generation is running, but the root jobnet's timeout period expires while the next generation is waiting for the current generation to end. • Skip that day's schedule (<code>plan</code>) is set in Jobnet skip (<code>OVERSCHEDULE</code>) in the scheduler service settings. A run is scheduled for that day, and the scheduler service has been started in warm-start mode. • Skip exceeded schedule (<code>skip</code>) is set in Jobnet skip (<code>OVERSCHEDULE</code>) in the scheduler service settings, or Execute from next time was selected in Planned time passed when daemon starts when registering the jobnet. The scheduler service has been started in warm-start mode, and there is a resident schedule that was due to run before the scheduler service started.
-- (invalid-seq)	Y	Y	N	N	The execution sequence is in a loop and there is job or jobnet that cannot be executed.

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connect or	
	Root	Nested			
Interrupted ^{#1} (interrupt)	Y	Y	N	N	<p>Execution of the jobnet has been interrupted. There are no abnormally terminated^{#9} units in the jobnet, but there is a unit that has not been processed (in <i>Not executed + Ended</i> status).</p> <p>A unit is placed in <i>Not executed + Ended</i> status in the following cases where the preceding unit has not terminated abnormally:^{#7}</p> <ul style="list-style-type: none"> • The jobnet is interrupted from JP1/AJS3 - View or by command. • A queued job is killed and not executed. • A queueless job is not executed due to a communication error or other problem. • An abnormally terminated^{#7} job in a jobnet is no longer in abnormally terminated status because it has been re-executed or a status change has been performed. • The scheduler service was stopped by executing the <code>ajsstop</code> command with the <code>-j</code> option specified. For the command syntax, see <code>ajsstop</code> in 2. <i>Commands</i> in the manual <i>Job Management Partner 1/ Automatic Job Management System 3 Command Reference 1</i>. <p>Note that if the scheduler service is started in warm-start mode, root jobnets in <i>Now running</i>, <i>Running + Abend</i>, or <i>Running + Warning</i> status are placed in <i>Interrupted</i> status regardless of the status of the subordinate units.</p>

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connector	
	Root	Nested			
Killed ^{#1} (kill)	Y	Y	Y	Y	<p>The job or jobnet was killed for one of the following reasons:</p> <ul style="list-style-type: none"> • An error occurred which prevented the end status of the job or jobnet from being determined (for example, the end result could not be written to the file at job execution). • The JP1/AJS3 - Manager service stopped while a job was being executed. • The system stopped while a job was being executed. • The active job was killed. • Processing was cancelled because the active job exceeded its specified job execution time. <p>In the case of a jobnet connector, the root jobnet was killed during execution of the jobnet connector.</p>
-- (kill-WR)	Y	Y	Y	N	The job or jobnet ended with a warning, and the attempted rerun was killed.
Failed to start ^{#1} (fail)	N	N	Y	N	Process creation failed when the job started, or the environment settings for the process could not be set.
-- (fail-WR)	N	N	Y	N	The job or jobnet ended with a warning, and the attempted rerun failed to start.
Unknown end status ^{#1} (unknown)	N	N	Y	Y	<p>The end status is unknown for either of the following reasons:</p> <ul style="list-style-type: none"> • An error occurred which prevented the end status of the job from being determined. For example, the end result could not be written to the file at job execution. • The JP1/AJS3 - Manager service stopped and the system shut down while the job was being executed. <p>In the case of a jobnet connector, a warm start was performed during execution.</p>

6. Monitoring Applications

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connect or	
	Root	Nested			
-- (unknown-WR)	N	N	Y	N	The job or jobnet ended with a warning, and the end status of the attempted rerun is unknown.
Shutdown (shutdown)	Y	Y	Y	Y	A jobnet was registered for planned execution, but a non-existent unit is specified in the jobnet's exclusive schedule or in the calendar referenced in the jobnet's schedule information.
Wait for start cond. (condition-wait)	Y	Y ^{#10}	N	N	The jobnet is waiting for the start condition to be satisfied. If concurrent execution is disabled and another jobnet is running, the jobnet keeps waiting for the start condition even if it has already been satisfied.
Now monitoring (monitoring)	Y	Y ^{#10}	N	N	The job or jobnet is monitoring for the events defined in the start condition.
Unmonitored + Ended (unexec-monitor)	Y	Y ^{#10}	N	N	The operation time expired before the start condition was satisfied.
Monitor terminated ^{#11} (monitor-close)	Y	Y ^{#10}	N	N	The jobnet that was monitoring for a start condition to be satisfied was interrupted or killed.
Interrupted monitoring ^{#11} (monitor-intrpt)	Y	Y ^{#10}	Y ^{#10}	N	Monitoring by a jobnet with a start condition was interrupted by the process that stops the scheduler service. The scheduler service was stopped either in restricted mode (job control, jobnet control, or schedule control option specified) or by forced termination (killing active jobs first). In this case, the next time the scheduler service starts (hot start or warm start), monitoring is restarted based on the latest definition of the event job specified in the start condition.
Monitor-end normal ^{#11} (monitor-normal)	Y	Y ^{#10}	N	N	The start condition was satisfied within the valid range (specified execution count or monitoring period).
End delay (end-delay)	Y	Y	Y	N	An end delay has occurred.

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connect or	
	Root	Nested			
Start delay (start-delay)	Y	Y	N	N	A start delay has occurred.
Nested jobnet delayed end (nest-end-delay)	Y	Y	N	N	An end delay has occurred in a lower-level unit.
Nested jobnet delayed start (nest-start-delay)	Y	Y	N	N	A start delay has occurred in a nested jobnet.

Legend:

Root : Root jobnet

Nested : Nested jobnet

Y: Displayed.

N: Not displayed.

--: Not applicable

#1

If you execute the `ajsshows` command to display the status, an `-R` mark, indicating *rerun*, is added to a unit that meets one of the following conditions:

- A unit that was rerun in a rerun operation
- A jobnet that contains a unit that was rerun
- A unit operating after the rerun in the root jobnet that contains rerun units

An `-R` mark is not added to a nested jobnet that is already in the *Now running* status when the rerun starts or to units included in that nested jobnet. If you rerun only specific units by specifying the **Only this unit** option, the `-R` mark is not added to succeeding units because they are not included in the rerun.

For the `ajsshows` command syntax, see `ajsshows` in 2. *Commands* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

#2

This status includes cases in which a jobnet generation whose start time has arrived or a re-executed jobnet generation is waiting for another jobnet generation

that is being executed when concurrent execution for the jobnet is disabled. It also includes cases in which the execution suppression function is suppressing the start of jobnet execution.

#3

For the *Not executed + Ended* and *Skipped so not exe.* status levels, you can distinguish between units that are *scheduled* and *not scheduled* in JP1/AJS3 - View by displaying them with different colored icons. This is not possible when the status name is only displayed in text format, or in the command output.

A jobnet becomes *scheduled* when an execution schedule (scheduled start time) has been set but the jobnet has not yet been executed. If no execution schedule has been set, the status of the job is *not scheduled*.

A job becomes *scheduled* when an execution schedule (scheduled start time) has been set for an upper-level jobnet, and the job has not been prohibited. In all other cases the job has *not scheduled* status.

You can set different colors for the *scheduled* and *not scheduled* statuses using the **General** page in the Preferences dialog box. For details, see 15.3.36 *Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide* and 15.3.38 *Select Color dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

#4

All nest jobnets or remote jobnets that are subordinate to a jobnet that has been placed in the *Skipped so not exe.* status are placed in the *Skipped so not exe.* status regardless of whether they have been scheduled. The jobs in those jobnets are placed in the *Not executed + Ended* status.

#5

The jobnet is placed in the *Not executed + Ended* status even when the upper-level jobnet has been placed in the *Skipped so not exe.* status.

#6

The next execution schedule is also placed in the *Skipped so not exe.* status. If successive generations are placed in this status, only the generation whose start time is closest to the current time remains.

#7

A jobnet with a timeout period of one day is still placed in *Skipped so not exe.* status if it is registered for execution immediately before the base time and the base time arrives before the jobnet is executed.

#8

When the jobnet has a start condition, this refers to an execution generation in

Wait for start cond. status, not to the generation in *Now monitoring* status.

#9

Abnormal termination shown here indicates one of the following abnormal termination statuses (*Not executed* + *Ended* is not included), which indicates that the unit was not executed:

- Ended abnormally
- Interrupted
- Invalid exe. seq.
- Killed
- Failed to start
- Unknown end status
- Skipped so not exe.

#10

This status might only be displayed for a start condition (.CONDITION) or for a jobnet for which a start condition is defined. The *Interrupted monitoring* status might also be displayed for an event job defined in the start condition.

#11

The status of units that belong to a root jobnet is *Not executed* + *Ended*.

Supplementary notes

- JP1/AJS3 - View includes a function for displaying a colored icon (yellow by default) for waiting units for which a hold attribute has been set. The color of the icon indicates the hold plan for the unit. Waiting units are those with statuses such as *Waiting for start time*, *Waiting for prev. to end*, *Waiting to execute* and *Wait for start cond.* You can access the function for displaying a hold plan in the Preferences dialog box. For details about accessing this function, see the description of the **General** page in *15.3.36 Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. You must be running JP1/AJS - Manager and JP1/AJS - View version 06-71 or later to use this function.
- At termination of a preceding unit that belongs to a *Wait for prev. to end* jobnet for which the hold attribute is set, the succeeding jobnet moves from *Wait for prev. to end* to *Wait for start time* status, and then to *Being held* status.
- In JP1/AJS3 - View or JP1/AJS2 - View version 08-50 or later, you can set a

filter so that *Waiting to execute* units for which the hold attribute is set will be included in listings of *Being held* units in the Monthly Schedule window or Daily Schedule window. To include *Waiting to execute* units that have the hold attribute in listings of *Being held* units, in the Filter Settings dialog box select the **Being held status includes hold plan** check box. For details, see *15.7.11 Filter Settings dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

- The scheduled end time for a jobnet or job is displayed together with the execution status. The scheduled end time is calculated based on the average execution time of the job or jobnet in the past (the average processing time of the job or jobnet when it ended normally). For details, see *4.4.2(2) Execution simulation*. The actual time taken for a job or jobnet to execute depends on system conditions such as the CPU utilization at the time. The scheduled start time and the scheduled end time are therefore intended only as guidelines.

6.1.2 Using JP1/AJS3 - View to monitor jobs and jobnets

This section describes how to use JP1/AJS3 - View to monitor jobs and jobnets. For details on how to monitor jobs in the different windows of JP1/AJS3 - View, see *8. Checking the Execution Schedule and Execution Status of a Jobnet and Jobs* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

In JP1/AJS3, you can use the following screens to check the execution status and execution results of a job or jobnet.

- JP1/AJS3 - View window
- Jobnet Monitor window
- Daily Schedule window
- Monthly Schedule window

The use of each window to monitor jobs or jobnets is described below.

(1) Monitoring via the JP1/AJS3 - View windows

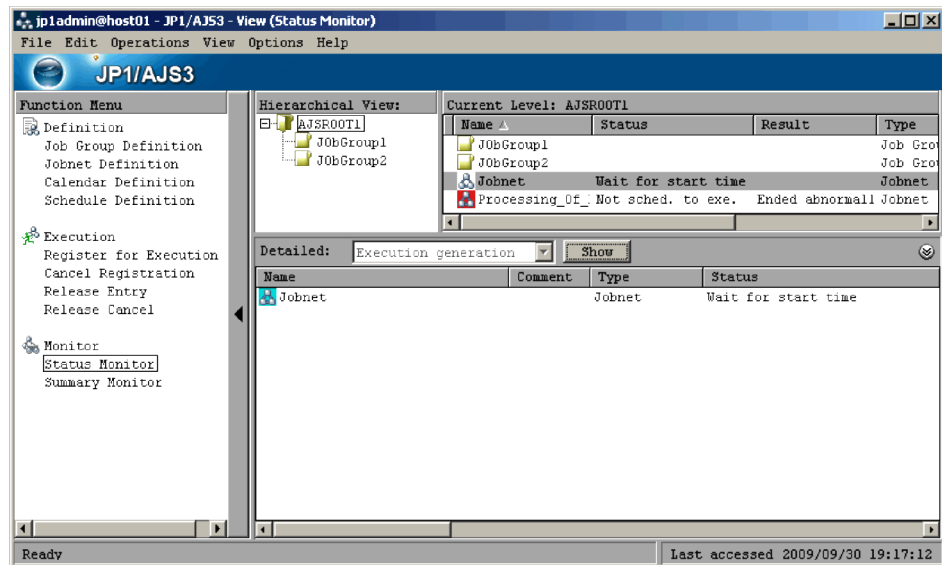
From the JP1/AJS3 - View window (Main window), you can monitor the execution status of root jobnets and job groups and check their execution results. From the JP1/AJS3 - View window (Summary Monitor window), you can monitor the progress of a root jobnet and check the execution status of the root jobnet and lower-level jobs. These two windows are described next.

(a) JP1/AJS3 - View window (Main window)

In the JP1/AJS3 - View window (Main window), you can view a list of execution information including the status and start time of root jobnets and job groups.

The JP1/AJS3 - View window (Main window) is shown below.

Figure 6-1: JP1/AJS3 - View window (Main window)



To check the execution status or execution result of a unit, in the **Function Menu** choose **Status Monitor**. You can then view information about job groups and registered jobnets in the list area and detailed information area.

■ Monitoring root jobnets in the list area

To monitor job groups in the list area, select an upper-level job group in tree area. The root jobnets defined directly under the selected job group appear in the list area.

To monitor root jobnets in the list area, select the upper-level job group in the tree area. The root jobnets defined directly under the selected job group appear in the list area.

Which generation is shown in the **Status** field depends on the value set in the VIEWSTATUSRANGE environment setting parameter in the JP1/AJS3 - Manager that the user is logged in to. The parameter settings, and the priorities on which the generation determination is based, are as follows:

- VIEWSTATUSRANGE=all (default)
 1. The execution generation[#] whose start time or rerun time is closest to the current time
 2. A scheduled generation[#] that has been re-executed and is in wait state
 3. A scheduled generation[#] due to execute in the next run, or a generation in *Shutdown* status
- VIEWSTATUSRANGE=today

1. The execution generation[#] whose start time or rerun time is closest to the current time
2. A scheduled generation[#] that has been re-executed and is in wait state
3. A scheduled generation[#] set for today and due to execute in the next run, or a generation in *Shutdown* status
4. Of the result generations[#] that completed execution today, the result generation whose end time is closest to the current time

If there is no generation that meets the above criteria, **Not sched. to exe.** appears in the **Status** field.

#

The listed statuses are categorized as being for an execution generation, scheduled generation, or result generation. The following table lists execution statuses for each generation category.

Table 6-2: Execution statuses and generation categories

No.	Category	Displayed status levels
1	Execution generation	<ul style="list-style-type: none"> • Now running • Running + Warning • Running + Abend • Now monitoring • Wait for start cond.
2	Scheduled generation	<ul style="list-style-type: none"> • Wait for start time • Being held
3	Result generation	<ul style="list-style-type: none"> • Ended normally • Ended with warning • Ended abnormally • Interrupted • Killed • Invalid exe. seq. • Skipped so not exe. • Monitor-end normal • Unmonitored + Ended • Monitor terminated • Interrupted monitoring

Supplementary note

The color of an icon displayed in the **Name** column indicates information in either the **Status** or **Result** column. In the **Status** and **Result** columns, you can also display color images that represent the status. For details about these settings, see

the description of the **Main** page in 15.3.36 *Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*. For details about the display colors used to represent statuses, see the description of the **General** page in 15.3.36 *Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

■ Monitoring a root jobnet in the detailed information area

To monitor a root jobnet in the detailed information area, select the root jobnet in the list area and then click the **Show** button. Information about the generations of the selected root jobnet appears.

Information about the following generations is displayed in the detailed information area:

- Today's generations^{#1}
 - Execution generation^{#2}
 - Scheduled generation whose start time falls within today^{#2}
 - Result generation whose end time falls within today^{#2}
- Generation of the previous day or earlier
Of previous generations not included in today's generations, the generation with the most recent end time.
- Generation of the next day or later
Of next generations not included in today's generations, the generation with the earliest start time.
- Generation in *Shutdown* status

#1

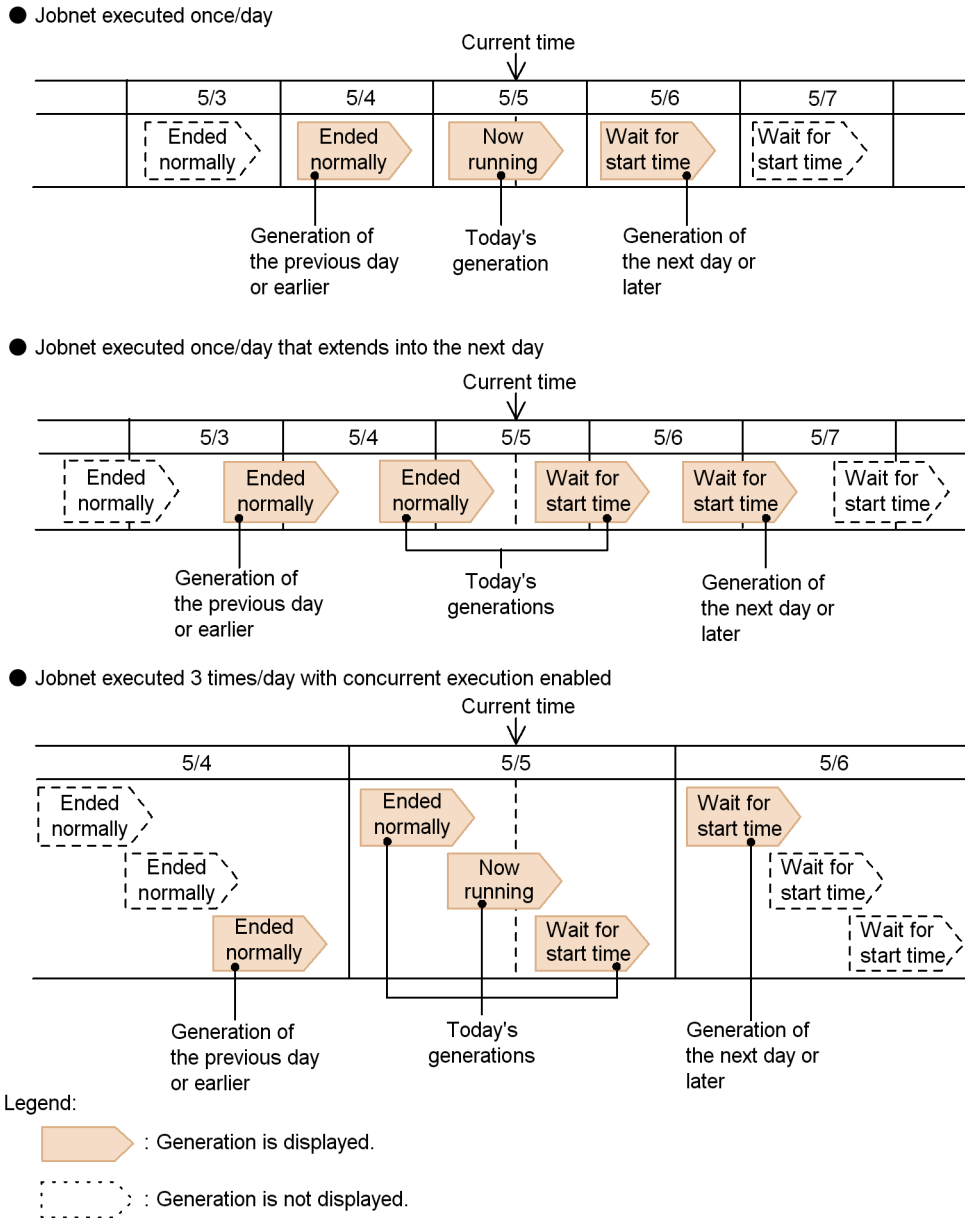
Today means the current day in the local time of the connected scheduler service.

#2

The execution generation, scheduled generation, and result generation for today are categorized into separate sets of execution statuses. For execution statuses and generation categories, see *Table 6-2*.

The following figure shows the generations displayed in the detailed information area.

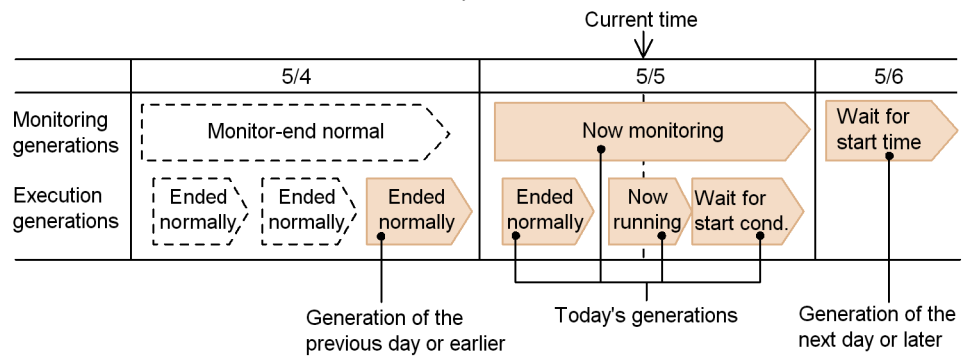
Figure 6-2: Jobnet generations displayed in the detailed information area



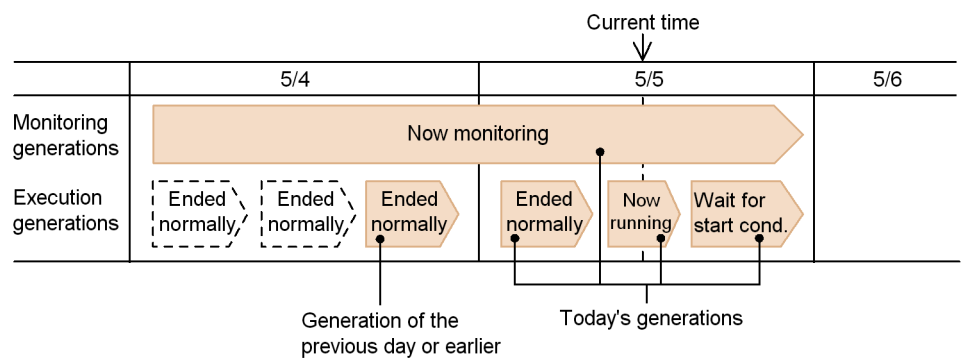
The following figure shows the generations displayed for a jobnet that has a start condition.

Figure 6-3: Jobnet generations displayed in the detailed information area (jobnet with a start condition)



- Jobnet with a start condition, executed once/day



- Jobnet with a start condition, monitoring for events over 2 or more days



Legend:

-  : Generation is displayed.
-  : Generation is not displayed.

Supplementary note

- Generations in *Not registered* and *Not sched. to exe.* statuses are not displayed.
- A rerun generation placed in *Wait for start time* and *Being held* status is categorized as being in *Now running* status.
- An execution generation of a jobnet with a start condition that has been placed in *Being held* status is categorized as being in *Now running* status.

■ Checking a root jobnet in the Monitor Details - [Jobnet] dialog box

In the Monitor Details - [Jobnet] dialog box, in addition to the jobnet status, you can

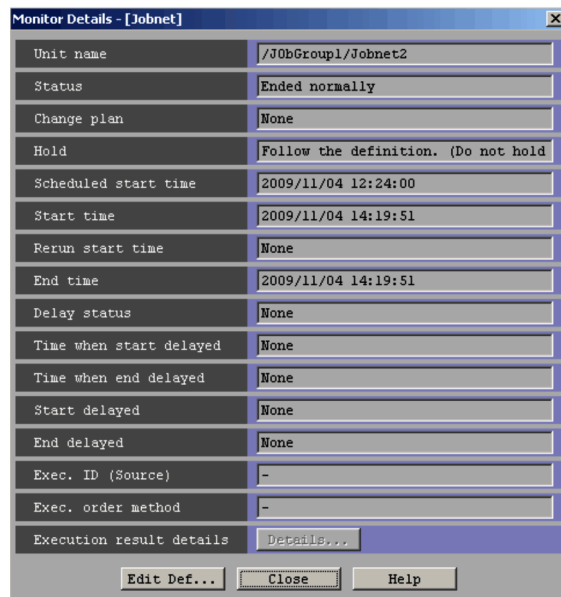
check information such as the unit's start time, end time, and delay status, and execution result details.

To check a root jobnet in the Monitor Details - [Jobnet] dialog box, select the root jobnet in the list area and choose **View, Detailed Information**, and then choose **Status, Result**, or **Next Schedule**. The generations displayed in the Monitor Details - [Jobnet] dialog box differ depending on the menu item you choose as described below.

- When **Status** is chosen:
The generations with the status displayed in the **Status** column in the list area are displayed.
- When **Result** is chosen:
The generations with the status displayed in the **Result** column in the list area are displayed.
- When **Next Schedule** is chosen:
The next generations due to run.

The Monitor Details - [Jobnet] dialog box is shown below.

Figure 6-4: Monitor Details - [Jobnet] dialog box



■ Monitoring the delay status of a unit in the delay list area and detailed information area

You can also check the delay status of a unit in the list area and the detailed information

area. The delay status appears in parentheses after the character string indicating the status name (for example, **Ended normally (delayed)**).

The delay status displayed for a job group or root jobnet in the list area or detailed information area of the JP1/AJS3 - View window (Main window) is determined by the status priority, as listed in the following table.

Table 6-3: Priority of delay status levels

Priority	Delay status	Delay display for root jobnets and job groups
1	Root jobnet end delay	(delayed)
2	Nested jobnet or job end delay	(nested jobnet delayed)
3	Root jobnet start delay	(delayed)
4	Nested jobnet start delay	(nested jobnet delayed)
5	Not delayed	Not displayed

From the Jobnet Monitor window you can monitor and check the execution status and execution results of the jobs and nested jobnets defined in the root jobnet.

Supplementary note

In the list area and detailed information area, the icons for delayed units are displayed in a color that represents its delay status. However, if an abend or warning status exists, by default, the abend or warning display has precedence. To give the delay status color higher priority than the abend and warning colors, you need to set the priority of delay display color. For details about delay color priority, see the description of the **General** page in 15.3.36 *Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

■ Monitoring job groups in the list area

To monitor job groups in the list area, select a scheduler service (default AJSROOT1) or an upper-level job group in tree area. The job groups defined directly under the selected unit appear in the list area.

In the list area, you can display the status of a job group that corresponds to the status of a jobnet defined in the job group. To display the status of a job group in the list area, you need to specify the setting. For details, see the description of the **Main** page in 15.3.36 *Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

The status displayed for a job group or root jobnet in the list area or detailed information area is that of the job or jobnet with the highest priority among those defined under the job group or root jobnet. The different priorities of status levels that

the job group or root jobnet can acquire from lower-level units are shown in the following table.

Table 6-4: Priority of status levels acquired by job groups and root jobnets

Priority	Job group status	Route jobnet status
1	Shutdown	Shutdown
2	Running + Abend	Running + Abend
3	Ended abnormally	Ended abnormally Interrupted Killed Invalid exe. seq.
4	Running + Warning	Running + Warning
5	Ended with warning	Ended with warning
6	Now running	Wait for start time Being held Wait for start cond. Now running Now monitoring
7	Ended normally	Ended normally Skipped so not exe. Unmonitored + Ended Monitor-end normal Monitor terminated Interrupted monitoring
8	Not sched. to exe.	Not sched. to exe.
9	None	Not registered

(b) JP1/AJS3 - View window (Summary Monitor window)

In the JP1/AJS3 - View window (Summary Monitor window), you can monitor the

progress of an application, including the numbers of scheduled units and completed units on the specified day, and their progress percentages. From the number of delayed units, scheduled end times, and other information, you can predict when the application will finish.

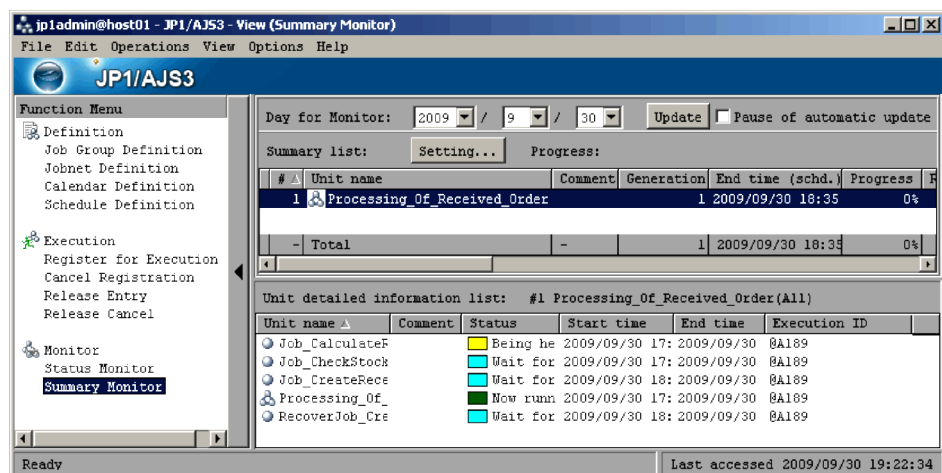
Progress field

The purpose of the **Progress** field is to show how far a normally running unit has progressed toward completion. When an error occurs during execution of a unit, the number of scheduled units and the progress percentage might decline during the course of monitoring. Because units not running normally are counted as abnormally terminated units, you need to investigate the cause of the error using the Jobnet Monitor window or by some other means. Take appropriate action, and then rerun the affected units and resume monitoring.

The units displayed in this window are set on a root jobnet basis.

The JP1/AJS3 - View window (Summary Monitor window) is shown below.

Figure 6-5: JP1/AJS3 - View window (Summary Monitor window)



To check unit progress in the JP1/AJS3 - View window (Summary Monitor window), in the **Function Menu** choose **Summary Monitor**. The JP1/AJS3 - View window (Summary Monitor window) opens.

Monitoring a unit in the Summary list

In the Summary list, you can monitor the number of units due to run on a monitor day, the scheduled end time of the root jobnet, the progress expressed as a percentage, and the statuses of the units included in the jobnet.

To monitor a root jobnet in the summary list, click the **Setting** button. Then, in the displayed Monitoring Unit Select dialog box, set the jobnet you want to monitor.

Monitoring units in the Unit detailed information list

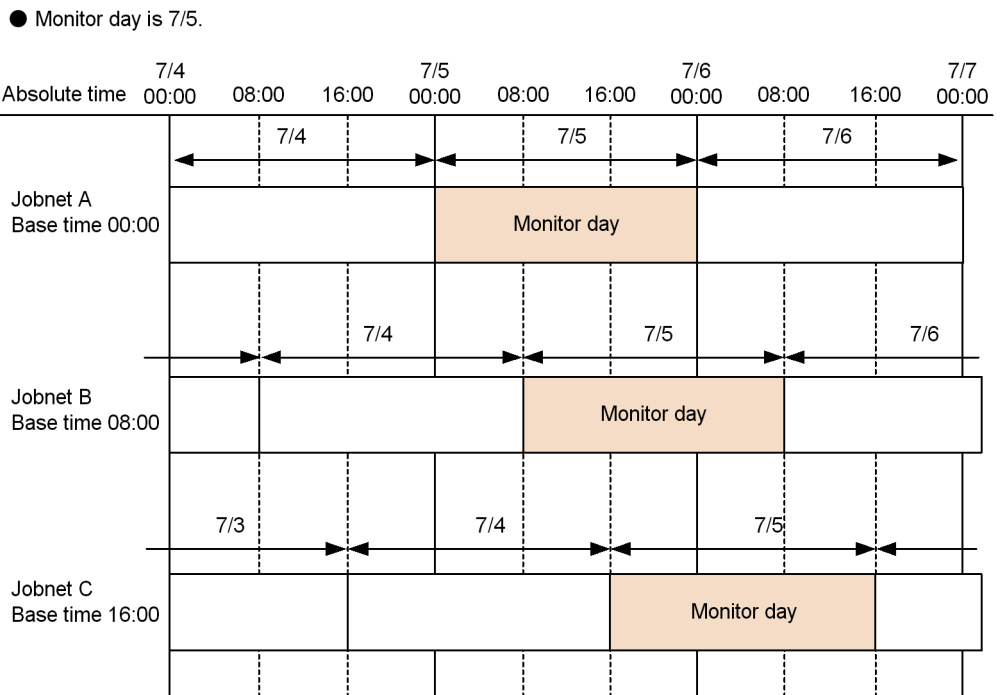
In the Unit detailed information list, you can monitor the execution status of a root jobnet and lower-level jobs.

To monitor units in the Unit detailed information list, select a unit in the Summary list, and then choose **View, Unit Detailed Information List**, and *execution status*. Of the selected jobnet and its lower-level units, those units with the status you select in *execution status* are listed in the window.

■ Monitor days and monitored generations

As the period to be monitored, you can specify the date on any one day. The span of one day is from the base time on that day to the next base time. When jobnets have different base times, their monitor days will cover different periods. The range of a monitor day is illustrated below.

Figure 6-6: Range of a monitor day



In this example, 7/5 is specified as the monitor day.

Because the base time of jobnet A is 00:00, the monitor day extends from 00:00 to 23:59 on 7/5 in absolute time.

Because the base time of jobnet B is 08:00, the monitor day extends from 08:00 on

7/5 to 07:59 on 7/6 in absolute time.

Because the base time of jobnet C is 16:00, the monitor day extends from 16:00 on 7/5 to 15:59 on 7/6 in absolute time.

A generation is monitored only if it meets one of the following conditions:

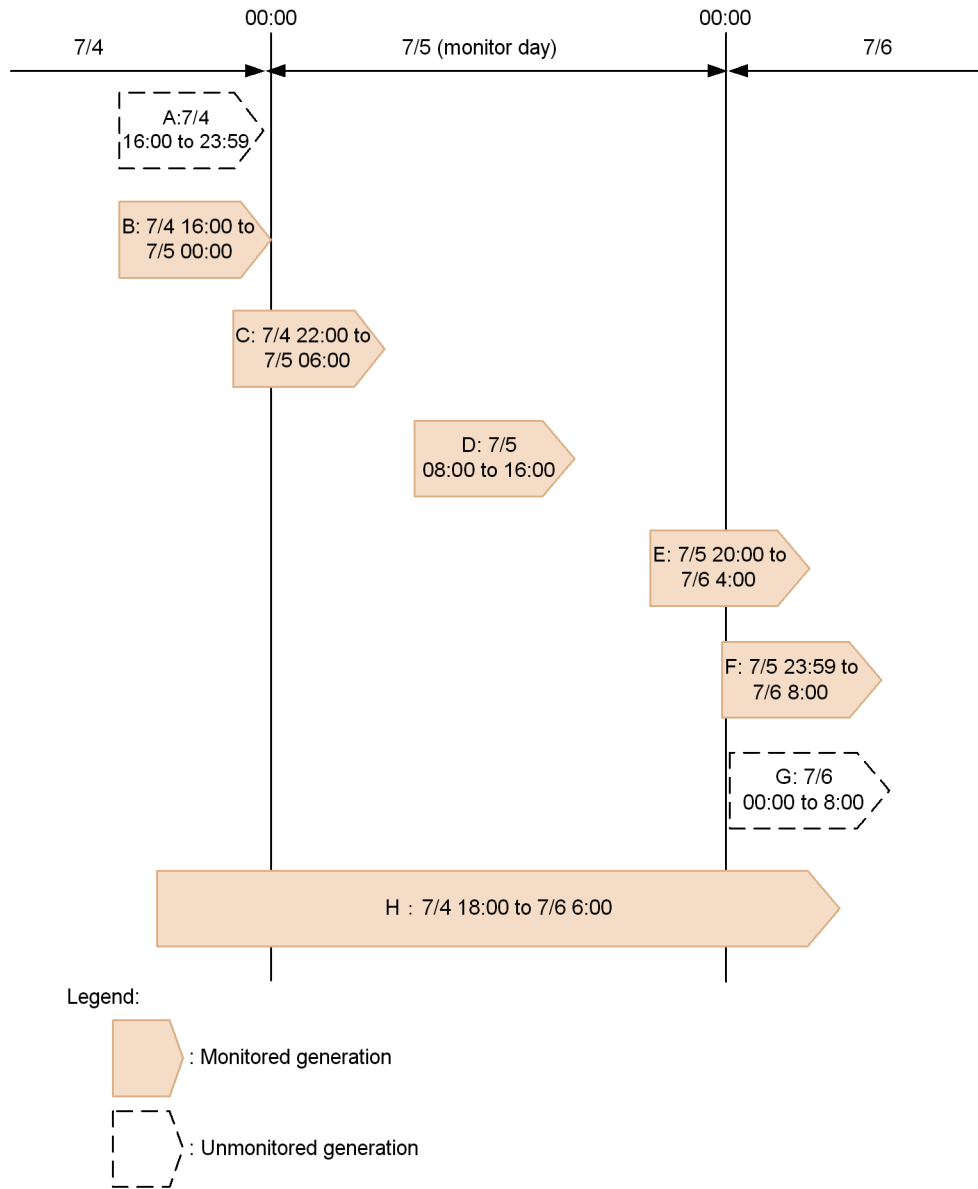
- The start time or scheduled start time falls within the specified day.
- The end time or scheduled end time falls within the specified day.
- The generation is in *Now running* status throughout the specified day.

If you specify today or a future date, dummy schedules are also monitored.

The following figure shows how JP1/AJS3 determines the generations to be monitored.

Figure 6-7: Determination of monitored generations

● Monitor day is 7/5, and the base time is 00:00.



In this example, 7/5 is specified as the monitor day.

Generations A and G are not monitored because their start and end times fall outside

the specified day.

Generations B and C are monitored because their end times fall within the specified day.

Generation D is monitored because its start and end times fall within the specified day.

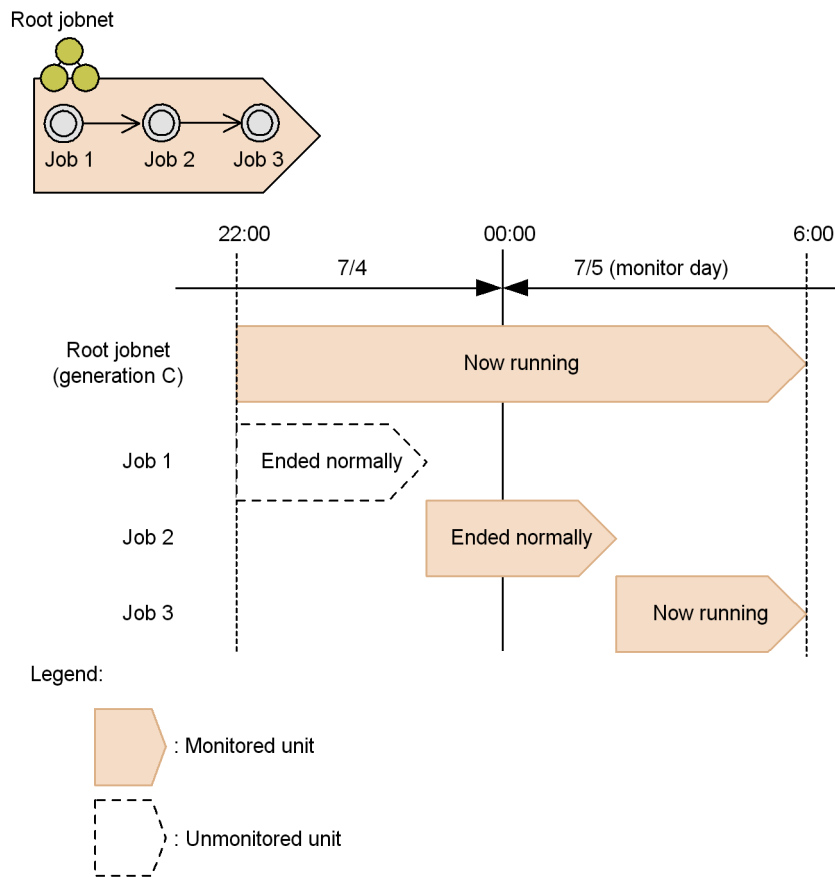
Generations E and F are monitored because their start times fall within the specified day.

Generation H is monitored because it is in *Now running* status throughout the monitor day.

When a generation is partially outside the monitor day, as in generations C and E, any lower-level units that fall completely outside the monitor day are not monitored.

This is illustrated below, taking the example of generation C in *Figure 6-7*.

Figure 6-8: Determination of monitored units (where the end time falls within the monitor day)



Root jobnet (generation C) is monitored because its end time falls within the monitor day.

Job 1 is not monitored because its start and end times fall outside the monitor day.

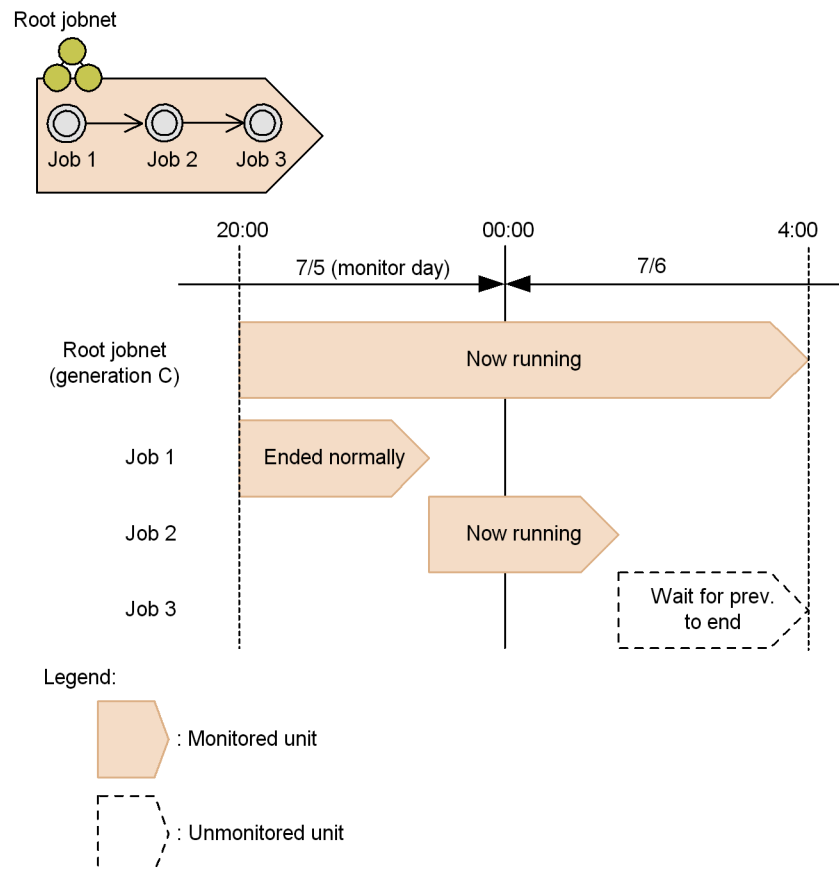
Job 2 is monitored because its end time falls within the monitor day.

Job 3 is monitored because its start and end times fall within the monitor day.

This means that if there is a delay of some sort, and an execution schedule that was due to end on the previous day extends into the monitor day, its status can still be checked as a schedule belonging to the monitor day.

This is illustrated below, taking the example of generation E in *Figure 6-7*.

Figure 6-9: Determination of monitored units (where the start time falls within the monitor day)



Job 1 is monitored because its start and end times fall within the monitor day.

Job 2 is monitored because its start time falls within the monitor day.

Job 3 is not monitored because its start and end times fall outside the monitor day.

This means that if there is a delay of some sort, and an execution schedule that was due to end on the monitor day extends into the next day, its status can still be checked from its scheduled end time.

Supplementary note

- Whether a rerun jobnet is monitored is determined from its original start time, not from its rerun start time.
- The span of the monitor day is 24 hours from the base time, regardless of

whether the 24-hour schedule or 48-hour schedule is set in JP1/AJS3 - Manager.

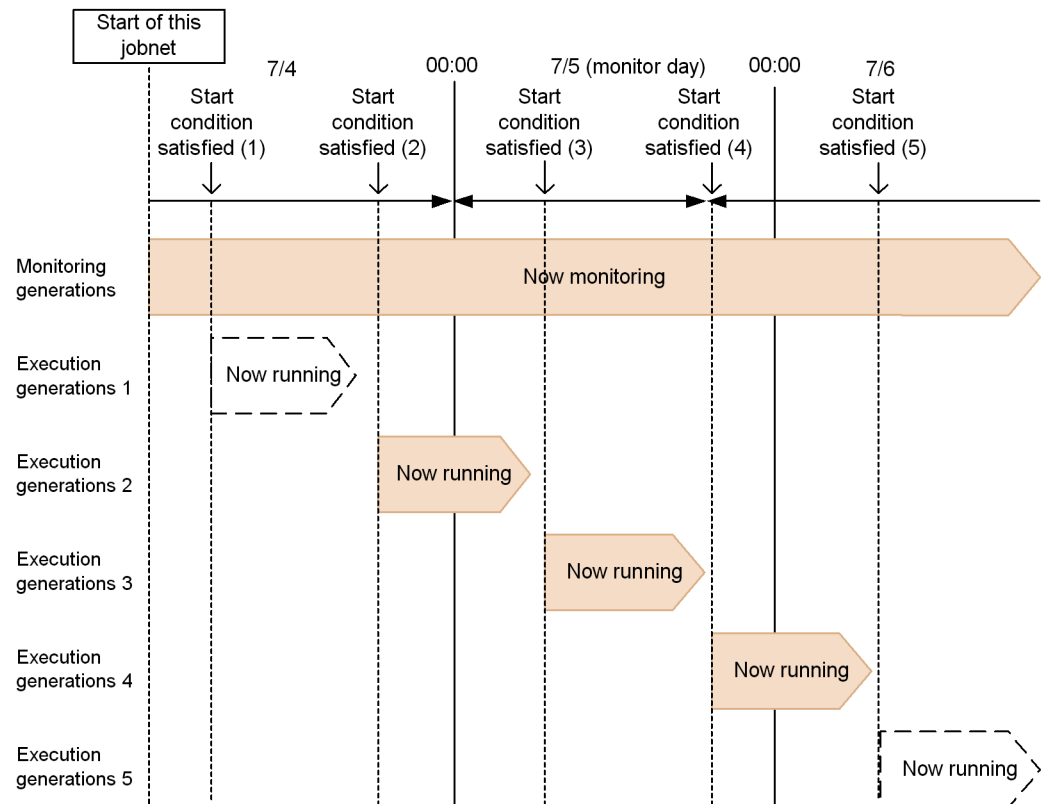
Monitoring a jobnet that has a start condition

For a jobnet that has a start condition, both the monitoring generations and execution generations are monitored.

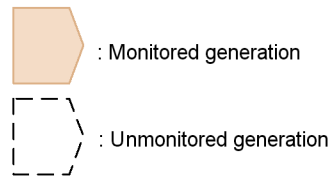
The following figure shows how the monitored generations are determined in this case.

Figure 6-10: Determination of monitored execution generations when the jobnet has a start condition

- Monitor day is 7/5, and the base time is 00:00.



Legend:



In this example, 7/5 is specified as the monitor day.

The monitoring generation is monitored because it is in *Now monitoring* status throughout the monitor day.

Execution generations 1 and 5 are not monitored because their start and end times fall outside the monitor day.

Execution generation 2 is monitored because its end time falls within the monitor day.

Execution generation 3 is monitored because its start and end times fall within the monitor day.

Execution generation 4 is monitored because its start time falls within the monitor day.

For an execution generation, whether a unit is monitored is determined in the same way as for a root jobnet, according to whether its start time or end time falls within the monitoring range. For a monitoring generation, however, whether a unit is monitored depends on the status of the monitoring generation, as follows:

Table 6-5: Status of a monitoring generation and monitored units

No.	Monitoring generation status	Monitored units
1	Wait for start time Being held	The root jobnet and all lower-level units are monitored.
2	Now monitoring Monitor-end normal Monitor terminated Unmonitored + Ended Interrupted monitoring Skipped so not exe.	Only the root jobnet is monitored.
3	Not sched. to exe.	No units are monitored.

For a dummy schedule, the root jobnet and all lower-level units are monitored.

■ Calculation of progress percentages

The progress percentage (%) is calculated using the following formula:

$$\text{Progress \%} = (\text{number of ended units}^{\#}) / (\text{number of units in the execution schedules falling on the monitor day}^{\#}) \times 100$$

#

The total includes the root jobnet as well as the jobs in the jobnet.

Cautionary note

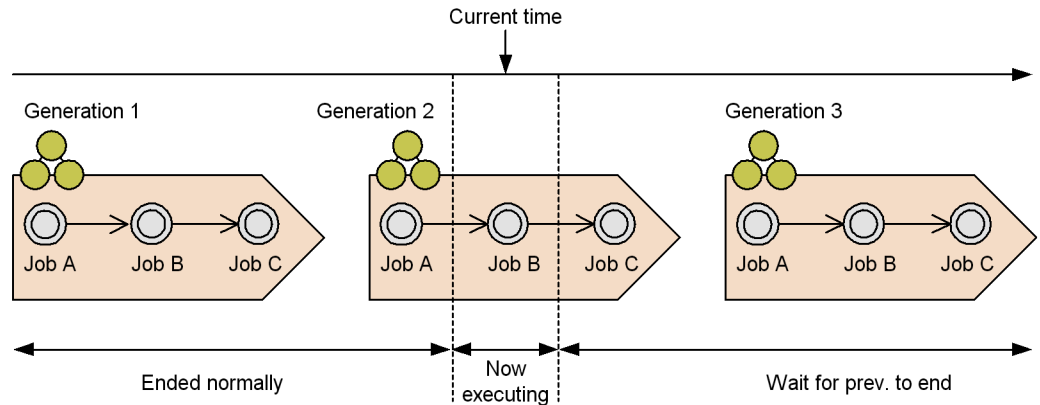
If an unlimited number of executions and an unlimited period are set as the valid range of the start condition, the monitoring generation is always in *Now monitoring* status and its progress percentage never becomes 100%.

The following examples illustrate how a progress percentage is calculated.

Example 1: When a jobnet is executed three times on the monitor day

The jobnet in this example is executed three times on the monitor day.

Figure 6-11: Example of a jobnet executed three times on the monitor day

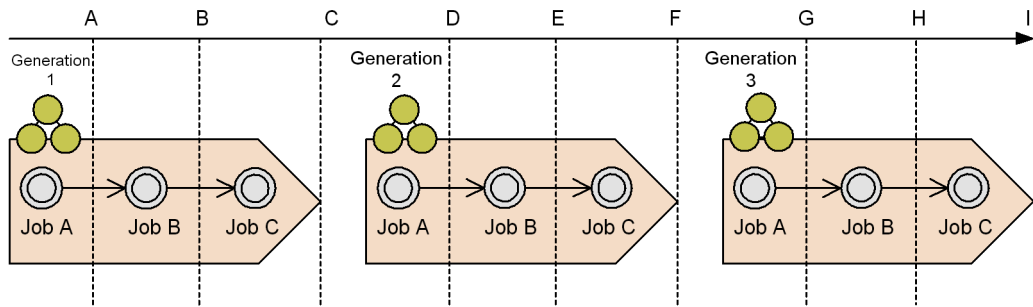


In this example, nine jobs and three jobnets (generations 1 to 3), or a total of 12 units, are scheduled to execute on the monitor day. Currently, jobs up to job A in generation 2 have ended normally, and the succeeding job B is now running. Therefore, four jobs plus one jobnet (generation 1), or a total of five units, are in ended status. Putting these values in above formula, we get a progress percentage of 41%.

Example 2: When the progress percentage declines during the monitor day

The progress percentage calculation is dependent on the number of logs to keep. Depending on the values set for these items, the progress percentage of a jobnet executed multiple times on the monitor day might decline during the course of monitoring.

Figure 6-12: Example of the progress percentage declining during the monitor day



- Number of logs to keep = 3, Future generations = 1

Monitor time	A	B	C	D	E	F	G	H	I
Number of ended units / Number of scheduled units	1/8	2/8	4/8	5/12	6/12	8/12	9/12	10/12	12/12
Progress percentage (%)	13	25	50	42	50	67	75	83	100

- Number of logs to keep = 1, Future generations = 3

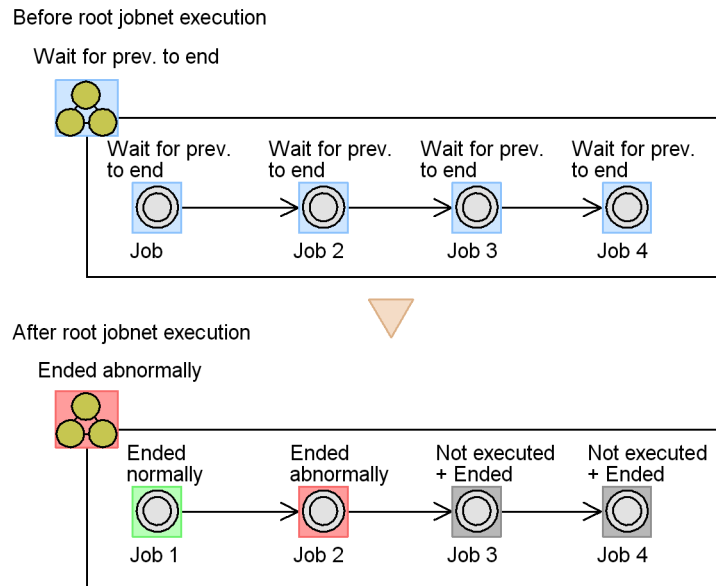
Monitor time	A	B	C	D	E	F	G	H	I
Number of ended units / Number of scheduled units	1/12	2/12	4/12	1/8	2/8	4/8	1/4	2/4	4/4
Progress percentage (%)	8	17	33	13	25	50	25	50	100

In this example, a jobnet is executed three times on the monitor day and the number of logs to keep is set to 1. In the current time frame A-C, the monitored units include the past job in generation 1 and the scheduled generations 2 and 3, making a total of 12 scheduled units to monitor. However, when execution of generation 2 starts, because the number of logs to keep is set to 1, the job in generation 2 becomes the past job, and generation 1 is deleted. Accordingly, in the current time D-F, the monitored units include the scheduled generation 3 and the past job in generation 2, making a total of eight scheduled units to monitor. Similarly, in the current time frame G-I, because the past job in generation 3 is also monitored, there are four scheduled units to monitor. That is, the progress percentage declines between C and D, and between F and G.

Example 3: When a job linked by relation lines ends abnormally

When the jobnet being monitored contains jobs linked by relation lines, the progress percentage depends on the execution results of those jobs. Suppose, for example, we have a jobnet configuration like the following one.

Figure 6-13: Example of a jobnet that contains jobs linked by relation lines



While the jobnet is waiting to execute, the ratio of ended units to scheduled units is 0:5, giving a progress percentage of 0%.

When the jobnet starts, job 2 ends abnormally and the succeeding jobs 3 and 4 enter *Not executed + Ended* status. Now the ratio of ended units to scheduled units is 1:3, giving a progress percentage of 33%.

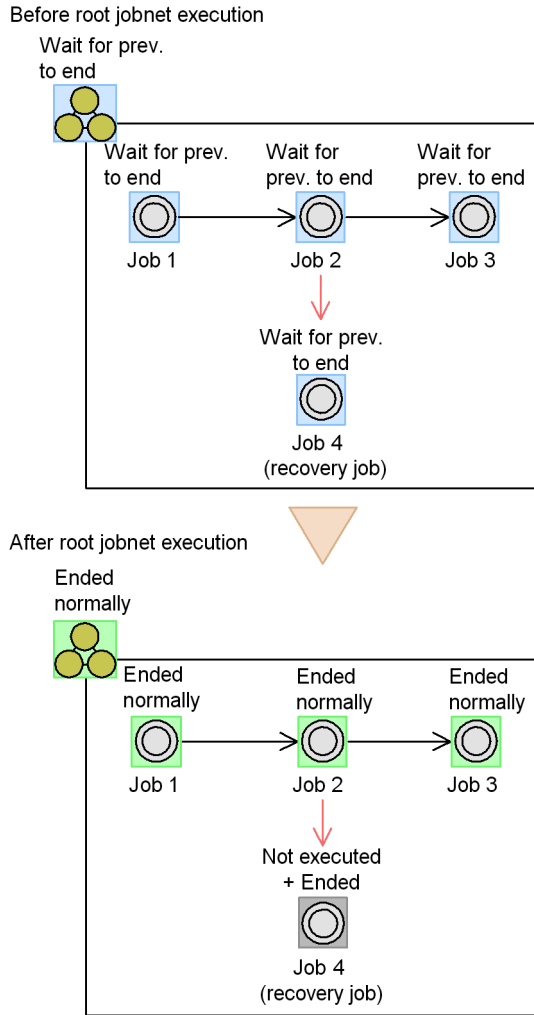
In the Summary list, **2** appears in the **Ended Abnormally** field, indicating how many jobs failed.

Example 4: When the jobnet contains a recovery job or recovery jobnet

When the jobnet being monitored contains a recovery job or recovery jobnet, the progress percentage depends on the execution result of the recovery job or jobnet.

Suppose, for example, we have a jobnet configuration like the following one.

Figure 6-14: Example of a jobnet that contains a recovery job



While the jobnet is waiting to execute, the ratio of ended units to scheduled units is 0:5, giving a progress percentage of 0%.

When the jobnet starts, job 2 ends normally and the jobnet completes execution. Recovery job 4, which follows job 2, is placed in *Not executed + Ended* status. In this case, the ratio of ended units to scheduled units is 4:4, giving a progress percentage of 100%.

As this example shows, recovery jobs and recovery jobnets, which are not executed unless there is an error, are excluded from the monitoring process, so the jobnet can be monitored in the usual manner.

■ Status classes

Unit statuses are classified and displayed as follows:

- **Ended Abnormally**
- **Ended with Warning**
- **Delay**
- **Being held**
- **Waiting to start**
- **Now running**
- **Ended Normally**

The following table lists the statuses in each class.

Table 6-6: Status classes

No.	Class	Statuses
1	Ended Abnormally	<ul style="list-style-type: none"> • Ended abnormally • Invalid exe. seq. • Interrupted • Killed • Failed to start • Unknown end status • Skipped so not exe.^{#1} • Monitor terminated
2	Ended with Warning	Ended with warning
3	Delay	<ul style="list-style-type: none"> • Start delay^{#2} • End delay^{#2}
4	Being held	Being held
5	Waiting to start	<ul style="list-style-type: none"> • Wait for start time • Wait for prev. to end • Waiting to execute • Now queuing • Wait for start cond.
6	Now running	<ul style="list-style-type: none"> • Now running • Running + Warning • Running + Abend • Now monitoring

No.	Class	Statuses
7	Ended Normally	<ul style="list-style-type: none"> • Ended normally • Normal end + False • Unmonitored + Ended • Interrupted monitoring • Monitor-end normal

Note

Units in the following statuses are excluded from the summary in the Summary Monitor window:

- Not registered
- Not sched. to exe.
- Bypassed
- Shutdown
- Not executed + Ended
- Nested jobnet delayed start
- Nested jobnet delayed end

#1

A unit that has no execution schedule, because execution was prohibited or it is a nested jobnet whose schedule does not match that of the upper-level jobnet, enters *Not sched. to exe.* status. If the root jobnet is skipped, the status of the unscheduled unit changes to *Skipped so not exe.* Units in *Skipped so not exe.* status are also excluded from the summary in the Summary Monitor window.

#2

When a start delay and end delay both occur in a unit, it is counted as a single delay. When a lower-level unit is delayed (*Nested jobnet delayed start* or *Nested jobnet delayed end* status), the upper-level unit being monitored is not counted as being in the **Delay** class.

The following table describes the relationship between delay statuses and the **Delay** count.

Table 6-7: Unit delay statuses and the number of units counted as being delayed

Case	Unit delay status				Number added to the Delay count
	Start delay	End delay	Nested jobnet delayed start	Nested jobnet delayed end	
1	N	N	N	N	0
2	N	N	N	Y	0
3	N	N	Y	N	0
4	N	N	Y	Y	0
5	N	Y	N	N	1
6	N	Y	N	Y	1
7	N	Y	Y	N	1
8	N	Y	Y	Y	1
9	Y	N	N	N	1
10	Y	N	N	Y	1
11	Y	N	Y	N	1
12	Y	N	Y	Y	1
13	Y	Y	N	N	1
14	Y	Y	N	Y	1
15	Y	Y	Y	N	1
16	Y	Y	Y	Y	1

Legend:

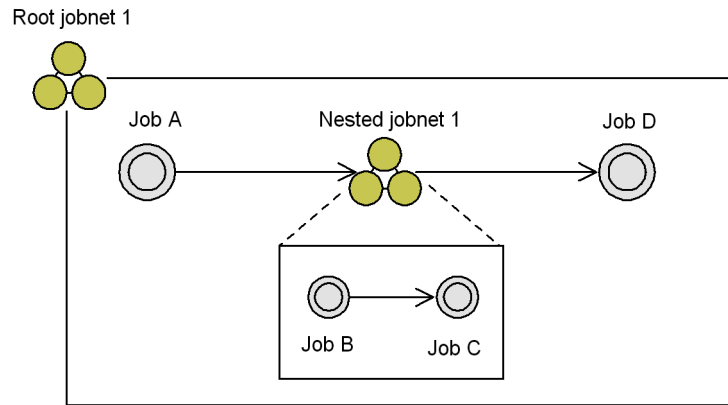
Y: Counted.

N: Not counted.

Note

When the unit being monitored is a job that started on time and has no lower-level units, it is counted as a delayed unit depending on whether an end delay occurred in the job itself (cases 1 and 5 in the above table).

For example, consider the following unit.



Example 1: Only the nested jobnet is delayed (case 2)

Only job B is delayed. Root jobnet 1 and nested jobnet 1 are not delayed and are not counted as delayed units. Only job B is counted in the total, so **1** appears in the **Delay** field.

Example 2: Both the upper-level unit and nested unit are delayed (case 6)

Root jobnet 1, nested jobnet 1, and job B are all delayed. Root jobnet 1 and nested jobnet 1 are counted as delayed units, but the delay in the lower-level units is disregarded. Therefore, the **Delay** count is incremented by 1 for each of root jobnet 1, nested jobnet 1, and job B, adding up to three delayed units in total.

Example 3: Both a start delay and end delay occur (case 13)

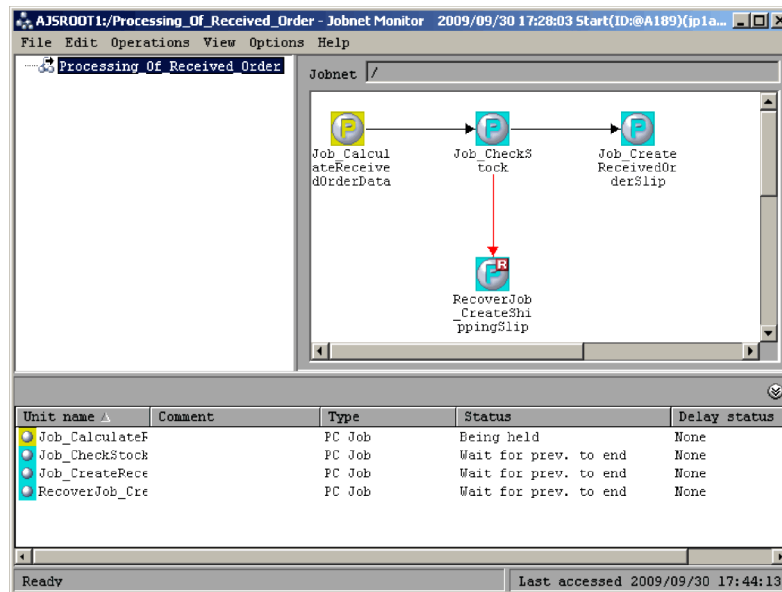
Root jobnet 1 both starts and ends later than scheduled. It is counted as a delayed unit, but the **Delay** count is incremented by 1 only, giving a total of one delayed unit.

(2) Monitoring via the Jobnet Monitor window

In the **Jobnet Monitor** window, you can monitor the execution status and check the execution results for the jobs and nested jobnets defined in a jobnet. This information is displayed in the form of a job flow image (the same image that is displayed when you define a jobnet).

The Jobnet Monitor window is shown below.

Figure 6-15: Jobnet Monitor window



The color of the icon for each unit displayed in the job flow image indicates the status of the unit the icon represents. For details about using different colored icons to show the status of a unit, see the explanation of the **General** page in *15.3.36 Preferences dialog box in the Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

To display the Jobnet Monitor window, from the JP1/AJS3 - View window (Main window) choose **Status**, **Result** or **Scheduled next**. By choosing **Status**, you can check the most recent status for generations whose status is **Being held**, **Now monitoring**, or **Waiting for start time**. By choosing **Result**, you can check the most recent results for generations that have completed execution. By choosing **Scheduled next**, you can check the generations that have statuses like **Waiting for start time** or **Being held** and are next scheduled for execution.

By choosing to display detailed information for a unit, you can check other information in addition to the status of the unit. This information includes the start time, end time and delay status of the unit, as well as detailed execution results. You can also check units' execution status and delay status in the list area.

Supplementary notes

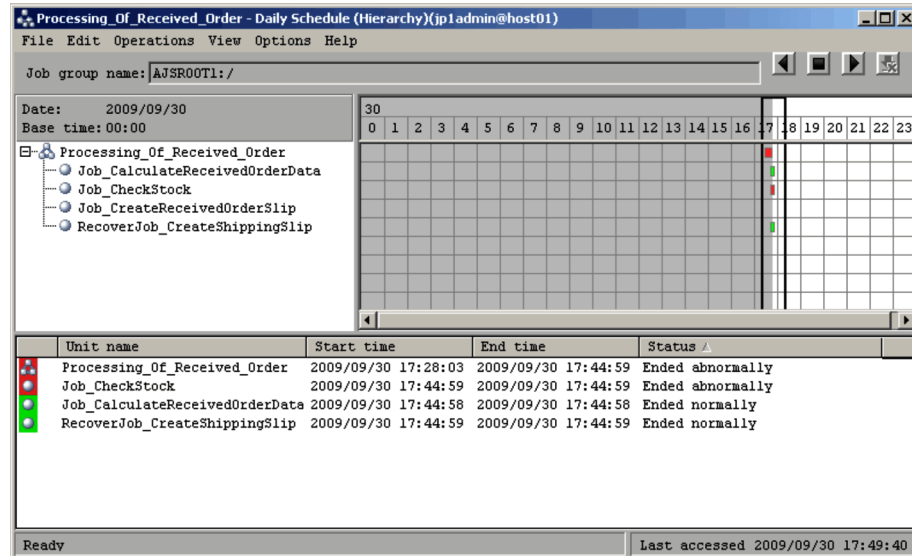
- You can display detailed information for jobnets, remote jobnets and manager jobnets.
- When you specify a manager jobnet, the execution status and results for the jobs referenced by the manager are displayed.

(3) Monitoring via the Daily Schedule window

In the **Daily Schedule** window, you can monitor the execution status and check the execution results for jobs and jobnets on a specific day, and also check the execution schedule for that day.

The Daily Schedule [Hierarchy] window is shown below.

Figure 6-16: Daily Schedule [Hierarchy] window



In the Daily Schedule window, you can check the execution schedule, execution status and execution results of units within a schedule that spans one day. The schedule is displayed against a time axis that is divided into one-hour units. The schedule area shows the schedule for each unit. The gray part of the schedule area indicates the past, and the white part indicates the future. If you select the schedule of a unit that is within the gray part of the schedule area, you can see detailed information about the execution results and execution status of the unit. If you select the schedule of a unit that is within the white part of the schedule area, you can see detailed information on the execution schedule in the execution result list.

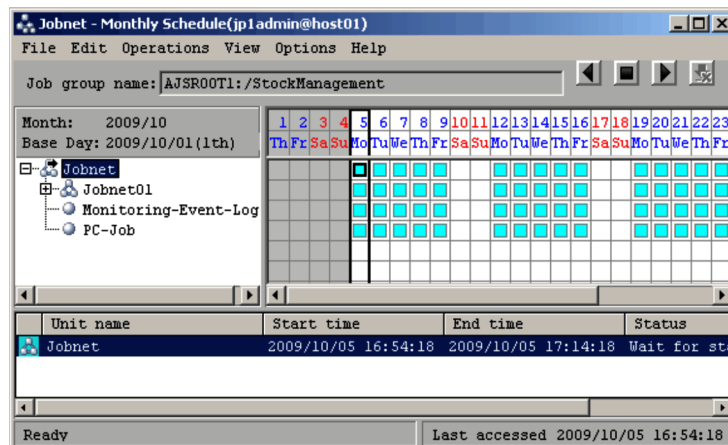
Generations that do not have an execution ID (generations that are scheduled for execution but do not have a defined schedule) are not displayed in the execution result list.

(4) Monitoring via the Monthly Schedule window

In the Monthly Schedule window, you can monitor the execution status and check the execution results for jobs and jobnets for a specific month, and also check the execution schedule for that month.

The Monthly Schedule window is shown below.

Figure 6-17: Monthly Schedule window



In the Monthly Schedule window, you can check the execution schedule, execution status and execution results of units within a schedule that spans one month. The schedule is displayed against a time axis that shows the days and dates for the specific month. The schedule area shows the schedule for each unit. The gray part of the schedule area indicates the past, and the white part indicates the future. If you select the schedule of a unit that is within the gray part of the schedule area, you can see detailed information about the execution results and execution status of the unit. If you select the schedule of a unit that is within the white part of the schedule area, you can see detailed information on the execution schedule in the execution result list.

Generations that do not have an execution ID (generations that are scheduled for execution but do not have a defined schedule) are not displayed in the execution result list.

Chapter

7. Monitoring Applications Using JP1/AJS3 Console

JP1/AJS3 Console is a program for monitoring applications that allows you to set the individual jobnets you want to monitor. JP1/AJS3 Console also allows you to centrally monitor, from a single screen, the status of jobnets that are distributed over multiple hosts or different hierarchies.

This chapter describes the monitoring of applications using JP1/AJS3 Console. For details about the on-screen operations used with JP1/AJS3 Console, see 13. *Centrally Monitoring Applications* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide* and 16. *JP1/AJS3 Console Screens* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

- 7.1 Overview of monitoring applications using JP1/AJS3 Console
- 7.2 Defining the applications to monitor and the monitoring method
- 7.3 Monitoring applications

7.1 Overview of monitoring applications using JP1/AJS3 Console

JP1/AJS3 Console is a monitor program that allows you to monitor the status of a jobnet.

This section provides an overview of the use of JP1/AJS3 Console to monitor applications. This section also describes how monitoring applications using JP1/AJS3 Console differs from monitoring in JP1/AJS3 - View.

7.1.1 Monitoring applications using JP1/AJS3 Console

JP1/AJS3 Console allows you to select which of the jobnets defined in JP1/AJS3 you want to monitor. JP1/AJS3 Console also allows you to centrally monitor, from a single screen, jobnets that are distributed over different manager hosts or different hierarchies (scheduler services, job groups).

The monitoring screen in JP1/AJS3 Console can be set up to display only the jobnets that the particular JP1 user wishes to monitor. To specify a jobnet, you need only specify the name of the host where the jobnet is defined, and the jobnet name (full name). The specified jobnet is displayed on the monitoring screen as an icon (AJS3 unit monitored object). The color of the icon indicates the status of the jobnet.

Examples of defining a monitoring task and monitoring a jobnet are shown in the following figures.

Figure 7-1: Defining a target for monitoring in JP1/AJS3 Console

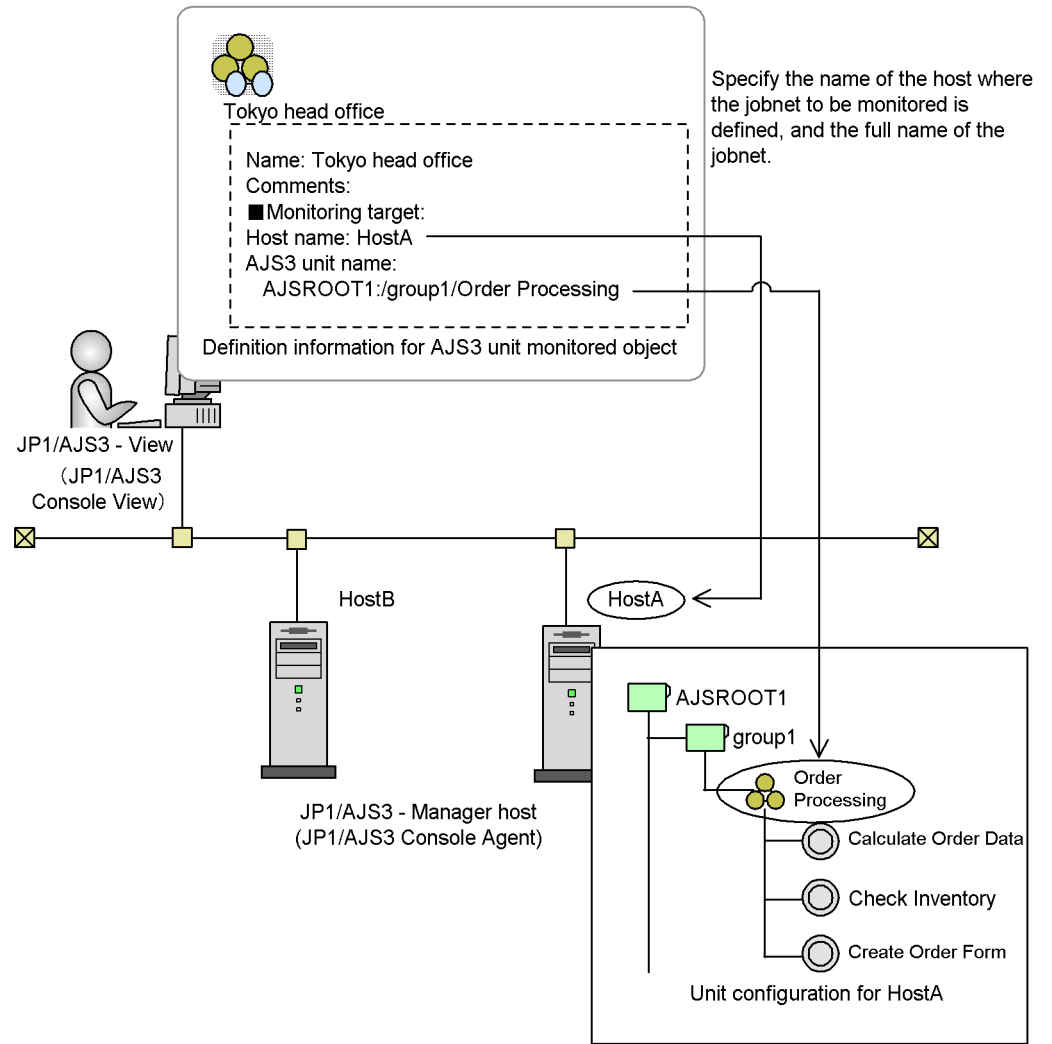
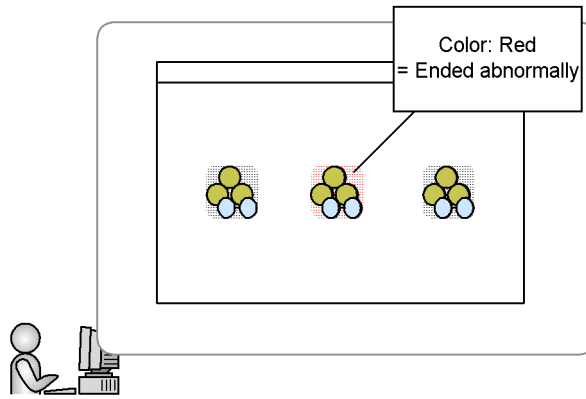


Figure 7-2: Monitoring a jobnet using JP1/AJS3 Console



When the monitoring screen shows that an error has occurred in a jobnet, you can use JP1/AJS3 Console to launch JP1/AJS3 - View on the remote host where the jobnet is defined. This allows you to easily identify where the error occurred and find the cause of the problem.

JP1/AJS3 Console also provides functionality, such as hierarchical management of monitored objects and background settings for monitoring screens, which gives you an enhanced visual representation of the system.

A JP1 user who logs in to JP1/AJS3 Console must fulfill certain conditions before he or she can use JP1/AJS3 Console to monitor applications. The user must:

- have permission to view the jobnet he or she wants to monitor.
- be mapped to an OS user on the host where the jobnet is defined.

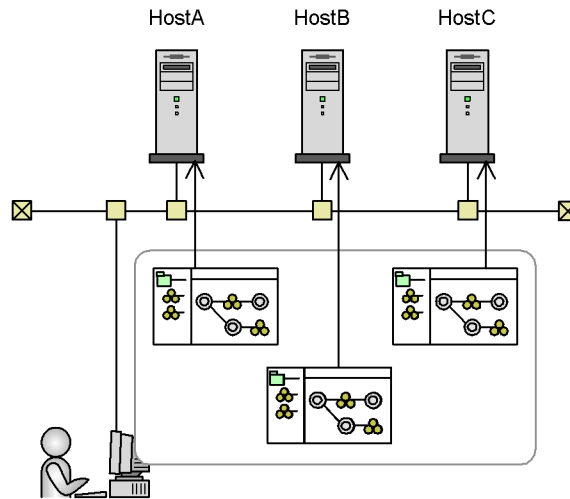
7.1.2 How JP1/AJS3 Console differs from JP1/AJS3 - View

In JP1/AJS3 - View, to monitor jobnets on different manager hosts in the Daily Schedule window or Jobnet Monitor window, you must log in and open a separate window for each manager host. To monitor jobnets that belong to several different hierarchies (scheduler services, job groups), you need to switch the display whenever you want to view a different hierarchical level.

The following figure shows an example of monitoring jobnets using JP1/AJS3 - View.

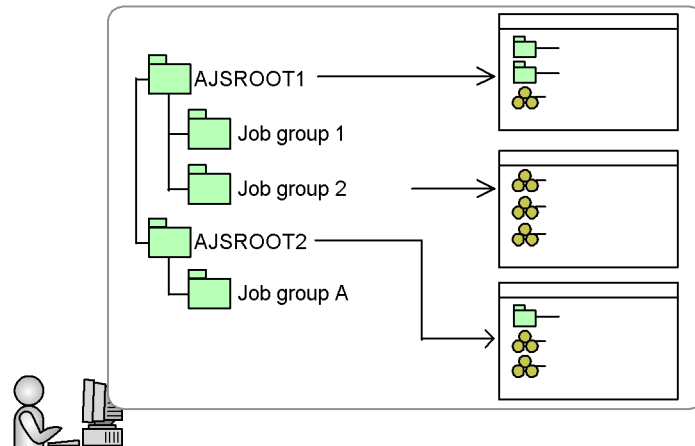
Figure 7-3: Monitoring jobnets using JP1/AJS3 - View

■ Monitoring jobnets on different manager hosts



Monitor jobnets by opening a separate JP1/AJS3 View window for each manager host

■ Monitoring jobnets on different job network hierarchies



Monitor jobnets by switching the display for each job network hierarchy

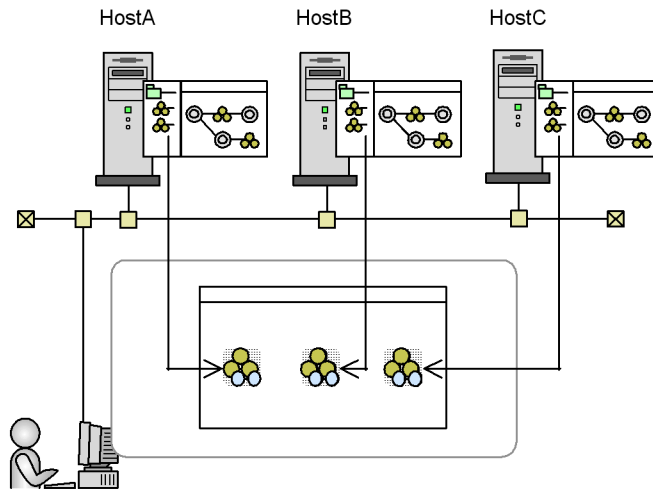
As shown in these examples, in JP1/AJS3 - View you cannot monitor jobnets running in different manager hosts, different scheduler services, or different job groups, except from the Summary Monitor window.

In contrast, with JP1/AJS3 Console you can define and monitor jobnets regardless of which manager host or hierarchy they belong to.

The following figure shows an example of monitoring jobnets using JP1/AJS3 Console.

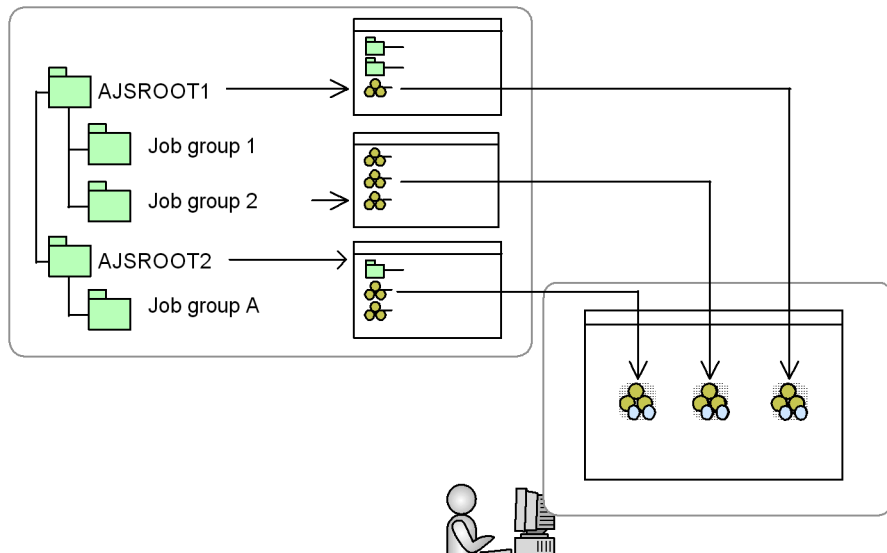
Figure 7-4: Monitoring jobnets using JP1/AJS3 Console

■ Monitoring jobnets on different manager hosts



Monitor jobnets on a single screen, even for different manager hosts

■ Monitoring jobnets on different job network hierarchies



Monitor jobnets on a single screen, without switching the display for each job network hierarchy

With JP1/AJS3 Console, you can collectively monitor jobnets from a single screen, without requiring a separate monitoring screen for each manager host, and without

needing to switch the screen display in order to view different hierarchies.

7.1.3 Components of JP1/AJS3 Console

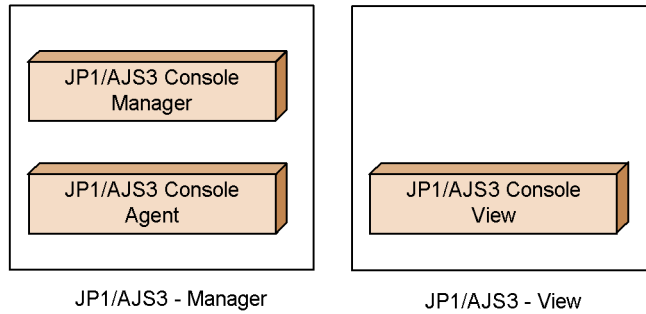
JP1/AJS3 Console is made up of the following three components.

Table 7-1: Components of JP1/AJS3 Console

No.	Component Name	Functions
1	JP1/AJS3 Console Manager	Acquires status information on the jobnets being monitored by JP1/AJS3 Console View, and saves definition information.
2	JP1/AJS3 Console Agent	Periodically monitors the execution status of the jobnets from JP1/AJS3 - Manager specified by JP1/AJS3 Console Manager, and informs JP1/AJS3 Console Manager of any changes in status.
3	JP1/AJS3 Console View	Provides the GUI environment for monitoring applications and defining which applications to monitor.

The following figure shows the program product configuration for JP1/AJS3 Console.

Figure 7-5: Program product configuration for JP1/AJS3 Console



7.2 Defining the applications to monitor and the monitoring method

To monitor an application in JP1/AJS3 Console, you define the jobnet you want to monitor as a monitored object, and set the monitoring method.

This section describes the objects you can use in JP1/AJS3 Console. This section also describes how to set up the screen mode, create an object, and specify the monitoring method in JP1/AJS3 Console.

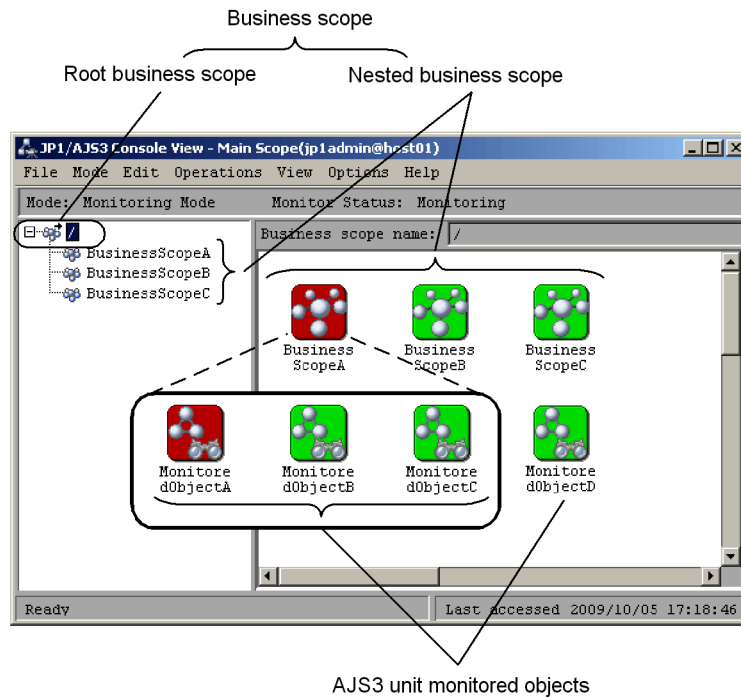
7.2.1 Objects used in JP1/AJS3 Console

With JP1/AJS3 Console, you can monitor jobnets defined under JP1/AJS3 using business scopes and AJS3 unit monitored objects. You can use two types of objects in JP1/AJS3 Console:

- AJS3 unit monitored objects
- Business scopes

The following figure shows the Scope window of JP1/AJS3 Console, containing the different objects you can use.

Figure 7-6: Scope window (Main Scope window) and objects



Below is a description of the different object types.

(1) AJS3 unit monitored objects

An AJS3 unit monitored object is an object that defines information about a jobnet to be monitored (such as the full name of the jobnet, the name of the host where the jobnet is defined). An AJS3 unit monitored object is displayed as an icon in the Scope window. You can monitor the status of the jobnet by watching for changes in the color of the icon.

(2) Business scopes

A business scope is an object that lets you group monitored objects together.

If required, you can establish hierarchies for business scopes. In a hierarchy, the business scope at the highest level is called the *root business scope*, and lower-level business scopes are called *nested business scopes*.

(a) Root business scopes

The *root business scope* is the highest-level business scope. It is displayed in the Scope window when you first log in to JP1/AJS3 Console Manager with JP1/AJS3 Console View.

In the root business scope, the JP1 user who is logged in to JP1/AJS3 Console Manager can define what objects to monitor on a user-specific basis.

(b) Nested business scopes

A *nested business scope* is a business scope that is defined below the root business scope in the hierarchy.

You can create up to three levels of nested business scopes. Nested business scopes are displayed as icons when an upper-level business scope is selected.

7.2.2 Modes

The Scope window (Main Scope window) has two modes. You alternate between these two modes according to whether you are defining the objects to monitor, or actually monitoring the objects.

Definition mode

Use this screen mode to create AJS3 unit monitored objects, and set the monitoring method for the objects.

Monitoring mode

Use this screen mode to monitor the AJS3 unit monitored objects you created. When you first log into JP1/AJS3 Console View, the Main Scope window is set to the monitoring mode.

Although the same JP1 user can log into the same JP1/AJS3 Console Manager host

from different copies of JP1/AJS3 Console View, the user modes available to such a user are limited as follows.

- The same JP1 user cannot log in while the existing user is in the process of creating definitions.
- The same JP1 user can log in during monitoring, but cannot then switch to **Definition mode** (only **Monitoring mode** is available).

The following table shows the modes available to a JP1 user who has more than one session open with a single JP1/AJS3 Console Manager host.

Table 7-2: Available modes

Mode	Can the same root business scope be used?		
	While Defining	While Monitoring	While Inactive
Definition mode	No	No	Yes
Monitoring mode	No	Yes	Yes

Legend:

Yes: Can be used

No: Cannot be used

7.2.3 Creating a monitored object

Before you can monitor applications using JP1/AJS3 Console, you must define the jobnets you want to monitor as AJS3 unit monitored objects in JP1/AJS3 Console.

(1) Creating AJS3 unit monitored objects

For each AJS3 unit monitored object, you must specify a name to represent the monitored object in JP1/AJS3 Console, the name of the host where the jobnet is defined, and the full name of the jobnet. Note that you can only monitor root jobnets. In JP1/AJS3 Console, you can define a maximum of 500 root jobnets as objects (AJS3 unit monitored objects and business scopes), across all route business scopes.

When you drag and drop an icon from the Main Scope window, the Define Details - [AJS3 Unit Monitored Objects] dialog box appears, as shown in the following figure.

Figure 7-7: Define Details - [AJS3 Unit Monitored Objects] dialog box

Name	MonitoredObjectA
Comment	
Icon file name	
Object to monitor	
Host name	host01
AJS3 unit name	AJSROOT1:/StockManagement/OrderProcessing

You can simplify the process of creating AJS3 unit monitored objects by using the function for automatically creating monitored objects that is provided in JP1/AJS3 - View.

You can specify any icon you want for a monitored object, by specifying the file name of the icon you want to display. Only a GIF file (.gif extension) can be set as an icon.

Supplementary note

The functionality for customizing icon files is not supported if you log in to JP1/AJS2 Console Manager from JP1/AJS2 Console 06-71.

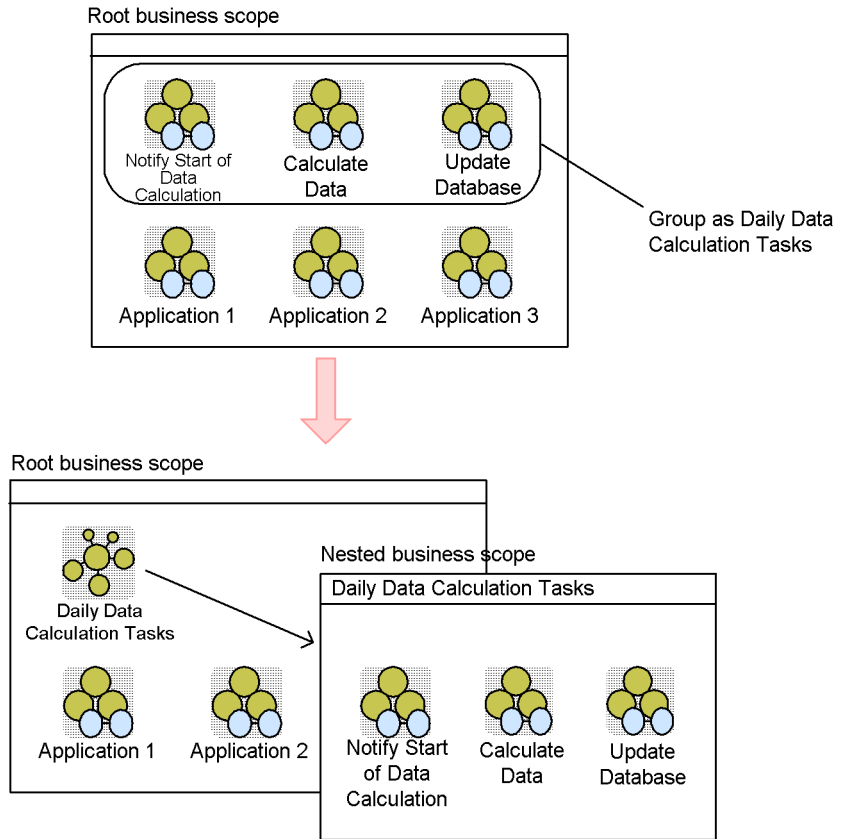
(2) Creating object hierarchies using nested business scopes

You can create a custom hierarchy in JP1/AJS3 Console that is unrelated to the hierarchy under JP1/AJS3. Sometimes you may want to manage AJS3 monitored objects in groups, if there are too many to monitor individually, for example. To manage these objects in groups, you can create nested business scopes and place the monitored objects in a hierarchy. You can create up to three levels of nested business scopes. When you select an upper-level business scope, the corresponding nested business scopes are displayed as icons in JP1/AJS3 Console View.

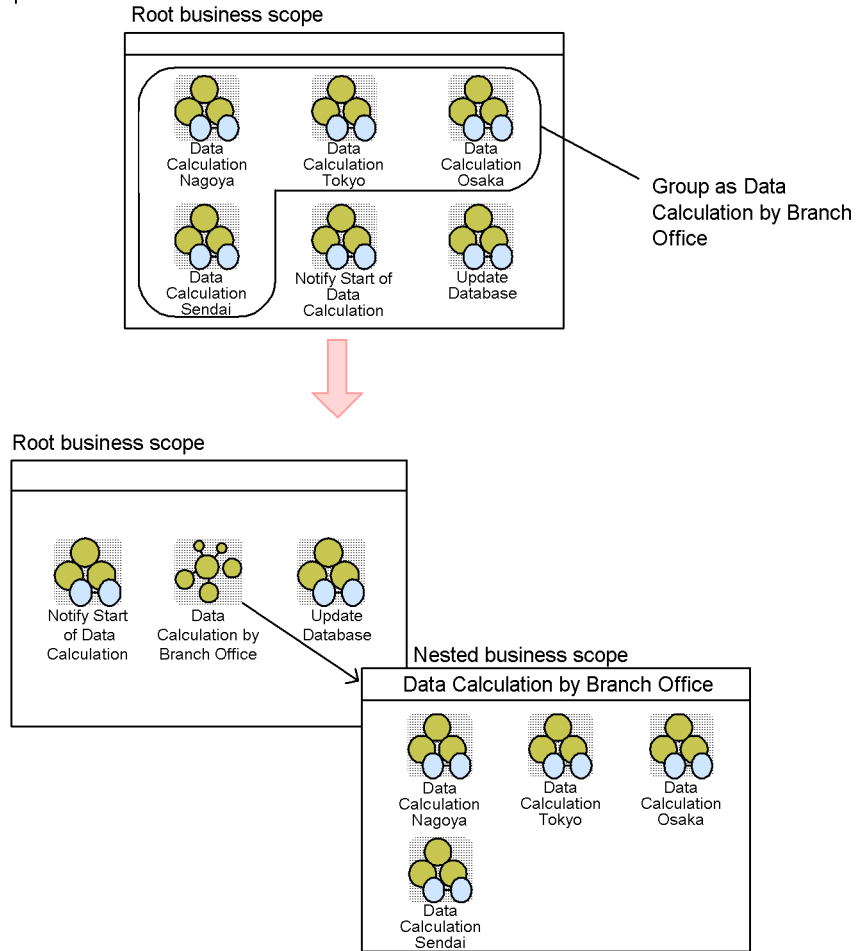
The following figures show hierarchies using nested business scopes.

Figure 7-8: Hierarchies using nested business scopes

Example 1



Example 2



In example 1, the user has created three AJS3 unit monitored objects. These objects are called **Notify start of data calculation**, **Calculate data**, and **Update database**. The user has then created a business scope called **Daily data calculation tasks**, and grouped the AJS3 unit monitored objects under this business scope. In example 2, the user has created AJS3 unit monitored objects such as **Data calculation Nagoya** and **Data calculation Tokyo** to monitor the data for each branch office. The user has grouped these objects under a specially created business scope called **Data calculation by branch office**.

Supplementary note

The functionality for creating nested business scopes is not supported if you log in to JP1/AJS2 Console Manager from JP1/AJS2 Console 06-71.

(3) Displaying a background image in the map area

You can specify a background image for display in the map area of a business scope.

For example, you could display an image of a map or a time axis as the background of the map area, and then arrange the icons on this image in a meaningful way.

Only a GIF file (.gif extension) or JPG file (.jpg extension) can be set as a background image for the map area.

Supplementary note

The functionality for setting background images is not supported if you log in to JP1/AJS2 Console Manager from JP1/AJS2 Console 06-71.

7.2.4 Setting the monitoring properties

The properties that you use to define how to monitor AJS3 unit monitored objects defined under a business scope are called *monitoring properties*. You can define the following three properties.

- Monitoring method
- Method of displaying a hold plan
- Monitoring interval

(1) Setting the monitoring method.

There are four monitoring methods available. You choose a method based on two factors: the range of generations to monitor for a jobnet, and the order of priority in which different status levels are displayed for the monitored generations. The four monitoring methods are as follows.

- Prioritize current time
- Prioritize all unit times
- Prioritize current time schedules
- Prioritize all unit time schedules

Each monitoring method is described below. The description is divided into the following two sections: covering the range of generations to monitor for a jobnet, and the order of priority in which different status levels are displayed.

If version 06-71 of JP1/AJS2 - Manager is running as the manager host where the jobnet you are monitoring is defined, you cannot monitor the jobnet while the monitoring mode is set to **Prioritize current time schedules** or **Prioritize all unit time schedules**. The status of the AJS3 unit monitored objects will appear as **Unknown**.

(a) Setting the range of generations to monitor in JP1/AJS3 Console

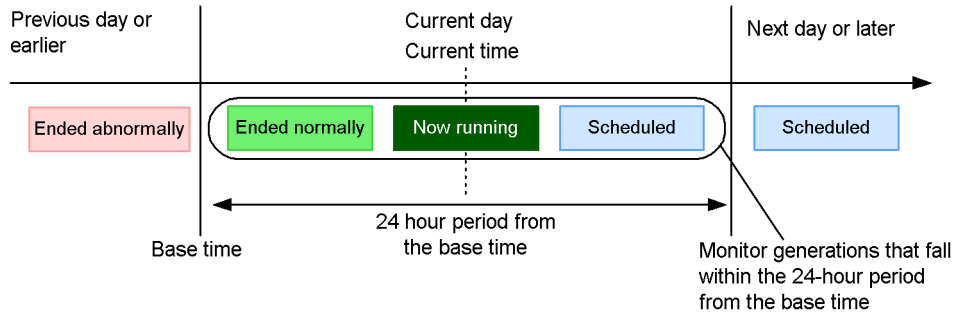
Jobnets, which are the basis of AJS3 unit monitored objects, consist of multiple generations. Each generation has its own status. You therefore need to decide the range of generations you want to monitor for a jobnet.

■ Monitor current day

With the monitor current day option, JP1/AJS3 Console monitors all generations within a period of 24 hours from the base time.

The following figure shows an example of monitoring the current day.

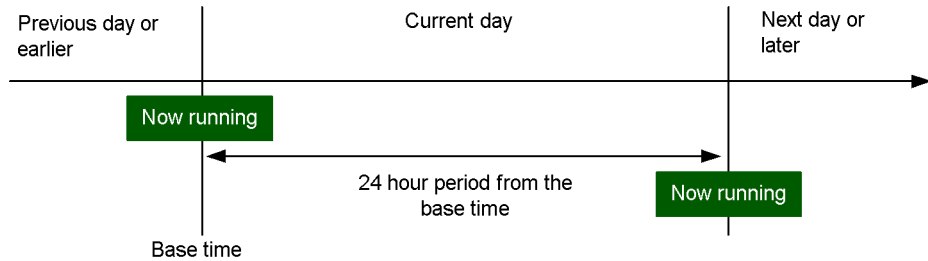
Figure 7-9: Example of monitoring the current day



When a generation extends beyond the current day, it is considered to be within the 24-hour period from the base time and is monitored.

The following figure shows an example of monitoring the current day when a generation extends beyond the 24-hour period.

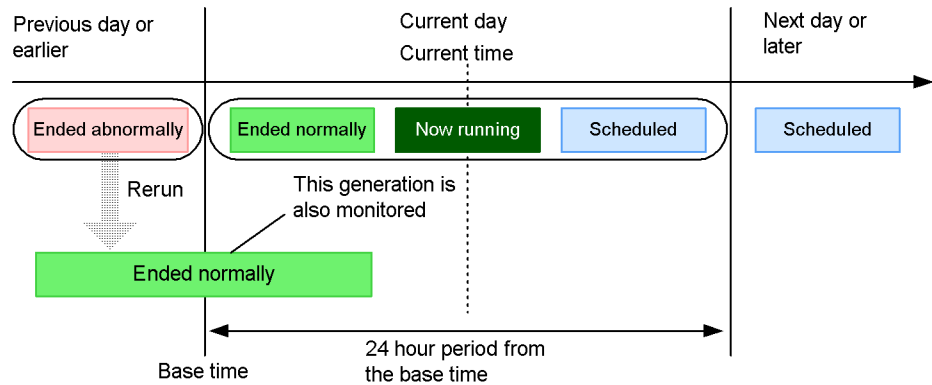
Figure 7-10: Example of monitoring the current day (when a generation extends beyond the 24-hour period)



If a generation that ran on the previous day or earlier is rerun, and the new start time or end time falls within the 24-hour period from the base time, it will also be monitored.

The following figure shows an example of monitoring the current day when a preceding generation is rerun.

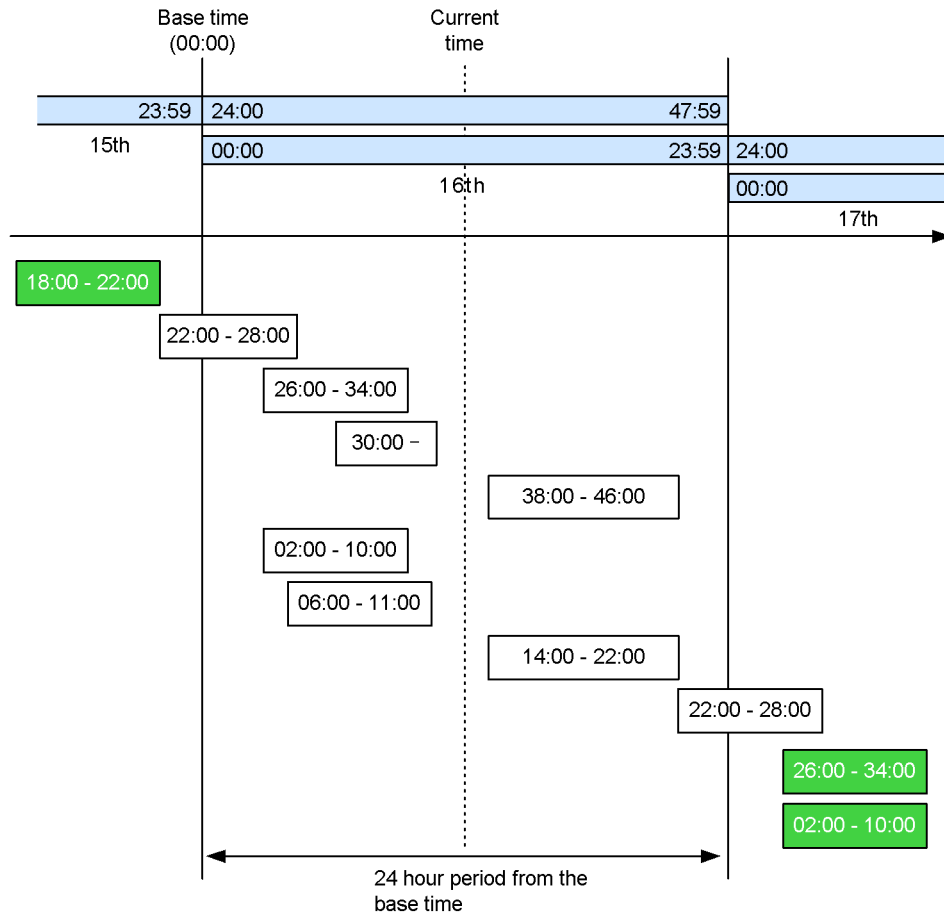
Figure 7-11: Example of monitoring the current day (when a preceding generation is rerun)



Note that even if the jobnet you are monitoring is on a 48-hour processing cycle, JP1/AJS3 Console will only monitor the generations that fall within the 24-hour period from the base time.

The following figure shows an example of monitoring the current day when the jobnet has a 48-hour processing cycle.

Figure 7-12: Example of monitoring the current day (with a 48-hour processing cycle)



Legend:

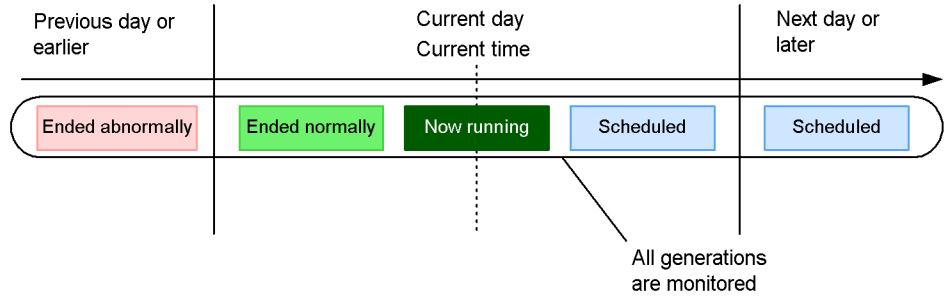
- : Generations to be monitored
- : Generations not monitored

■ Monitor all generations

With the monitor all generations option, JP1/AJS3 Console monitors every generation in the monitored jobnets.

The following figure shows an example of monitoring all generations.

Figure 7-13: Example of monitoring all generations



(b) Priority of statuses

There are two methods of prioritizing the status levels displayed for monitored objects. The first prioritizes results over schedules, while the second prioritizes schedules over results.

The following table describes the display priority for each monitoring method.

Table 7-3: Display priority for monitoring methods

Monitoring method	Display priority
Prioritize current time	Prioritize results over schedules
Prioritize all unit times	
Prioritize current time schedules	Prioritize schedules over results
Prioritize all unit time schedules	

The following tables show the order of priority in which different statuses are displayed.

- If you prioritize results over schedules

If you prioritize results, JP1/AJS3 Console displays the generations in the following order of priority, according to their statuses.

Table 7-4: Display priority of statuses when results are prioritized

Priority	Status of monitored generation	When more than one generation has the same status
1 (highest)	Now running	Prioritize according to closeness of start time (re-execution start time) to current time
2	Being held	Prioritize according to closeness of scheduled start time to current time
3	Waiting for start time[#] (re-execution time)	Prioritize according to which generation will reach the start time first
4	Shutdown	--
5	Results	Prioritize according to closeness of end time to current time (includes rerun generations)
6	Now monitoring	Prioritize according to closeness of start time to current time
7 (lowest)	Scheduled	Prioritize according to closeness of scheduled start time to current time

#

Only generations awaiting re-execution are given **Waiting for start time** status, which has a higher display priority than normal generations. Normal generations that are *waiting for start time* are included under **Scheduled**.

- If you prioritize schedules over results

If you prioritize schedules, JP1/AJS3 Console displays the generations in the following order of priority, according to their statuses.

Table 7-5: Display priority of statuses when schedules are prioritized

Priority	Status of monitored generation	When more than one generation has the same status
1 (highest)	Now running	Prioritize according to closeness of start time (re-execution start time) to current time
2	Being held	Prioritize according to closeness of scheduled start time to current time
3	Waiting for start time[#] (re-execution time)	Prioritize according to which generation will reach the start time first
4	Shutdown	--

Priority	Status of monitored generation	When more than one generation has the same status
5	Now monitoring	Prioritize according to closeness of start time to current time
6	Scheduled	Prioritize according to closeness of scheduled start time to current time
7 (lowest)	Results	Prioritize according to closeness of end time to current time (includes rerun generations)

#

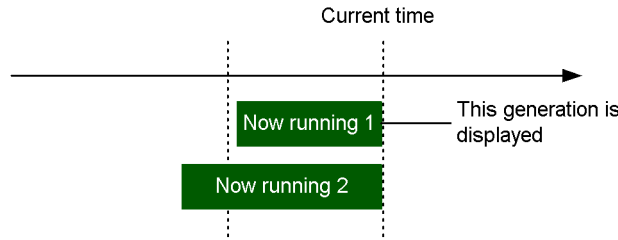
Only generations awaiting re-execution are given **Waiting for start time** status, which has a higher display priority than normal generations. Normal generations that are *waiting for start time* are included under **Scheduled**.

You can see from these two tables that generations with the **Now running** status are displayed with the highest priority. If more than one generation has the same status, they are prioritized in order from the generation with the start time, scheduled start time, or end time that is closest to the current time.

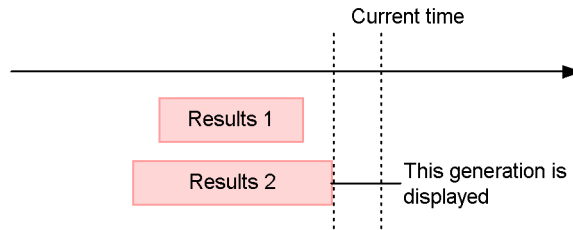
The following figure shows the generation that is displayed when more than one generation has the same status.

Figure 7-14: Display example for when more than one generation has the same status

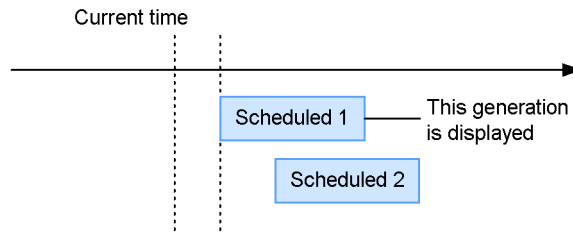
- Multiple generations in Now running status (also applies to Now monitoring status)



- Multiple generations in Results status



- Multiple generations in Scheduled status



(c) Examples of status display when results are prioritized and when schedules are prioritized

Examples of the status displayed for a generation in the case where results are prioritized, and the case where schedules are prioritized, are shown below.

■ **Past results and future plans for jobnets and the status displayed in JP1/AJS3 Console**

The following table shows the status levels (including past results and future plans) of the generations within the range being monitored, and the status displayed for the generations in JP1/AJS3 Console.

Table 7-6: Status levels (including past results and future plans) for each generation and the status displayed for the AJS3 unit monitored object

Status of each generation (including past results and future plans)					Status displayed for AJS3 unit monitored object			
Previous day and earlier	Current day			Next day and later	Prioritize results		Prioritize schedules	
	Past	Present	Future		Current day	All generations	Current day	All generations
Not registered	Not registered	Not registered	Not registered	Not registered	Not registered	Not registered	Not registered	Not registered
--	--	--	--	--	Not sched. to exe.	Not sched. to exe.	Not sched. to exe.	Not sched. to exe.
Results	--	--	--	--	Not sched. to exe.	Results	Not sched. to exe.	Results
--	Results	--	--	--	Results	Results	Results	Results
--	--	Now running	--	--	Now running	Now running	Now running	Now running
--	--	--	Scheduled	--	Scheduled	Scheduled	Scheduled	Scheduled
--	--	--	--	Scheduled	Not sched. to exe.	Scheduled	Not sched. to exe.	Scheduled
Results 1	Results 2	Now running	Scheduled1	Scheduled2	Now running	Now running	Now running	Now running
Results 1	Results 2	--	--	--	Results 2	Results 2	Results 2	Results 2
Results	--	Now running	--	--	Now running	Now running	Now running	Now running
Results	--	--	Scheduled	--	Scheduled	Results	Scheduled	Scheduled
Results	--	--	--	Scheduled	Not sched. to exe.	Results	Not sched. to exe.	Scheduled
--	Results	Now running	--	--	Now running	Now running	Now running	Now running

Status of each generation (including past results and future plans)					Status displayed for AJS3 unit monitored object			
Previous day and earlier	Current day			Next day and later	Prioritize results		Prioritize schedules	
	Past	Present	Future		Current day	All generations	Current day	All generations
--	Results	--	Schedule d	--	Results	Results	Schedule d	Schedule d
--	Results	--	--	Schedule d	Results	Results	Results	Schedule d
--	--	Now running	Schedule d	--	Now running	Now running	Now running	Now running
--	--	Now running	--	Schedule d	Now running	Now running	Now running	Now running
--	--	--	Schedule d1	Schedule d2	Schedule d1	Schedule d1	Schedule d1	Schedule d1
Results 1	Results 2	Now running	--	--	Now running	Now running	Now running	Now running
Results 1	Results 2	--	Schedule d	--	Results 2	Results 2	Schedule d	Schedule d
Results 1	Results 2	--	--	Schedule d	Results 2	Results 2	Results 2	Schedule d
Results	--	Now running	Schedule d	--	Now running	Now running	Now running	Now running
Results	--	Now running	--	Schedule d	Now running	Now running	Now running	Now running
Results	--	--	Schedule d1	Schedule d2	Schedule d1	Results	Schedule d1	Schedule d1
--	Results	Now running	Schedule d	--	Now running	Now running	Now running	Now running
--	Results	Now running	--	Schedule d	Now running	Now running	Now running	Now running
--	Results	--	Schedule d 1	Schedule d 2	Results	Results	Schedule d 1	Schedule d 1
--	--	Now running	Schedule d 1	Schedule d 2	Now running	Now running	Now running	Now running

Status of each generation (including past results and future plans)					Status displayed for AJS3 unit monitored object			
Previous day and earlier	Current day			Next day and later	Prioritize results		Prioritize schedules	
	Past	Present	Future		Current day	All generations	Current day	All generations
Results 1	Results 2	Now running	Scheduled	--	Now running	Now running	Now running	Now running
Results 1	Results 2	Now running	--	Scheduled	Now running	Now running	Now running	Now running
Results 1	Results 2	--	Scheduled 1	Scheduled 2	Results 2	Results 2	Scheduled 1	Scheduled 1
Results	--	Now running	Scheduled 1	Scheduled 2	Now running	Now running	Now running	Now running
--	Results	Now running	Scheduled 1	Scheduled 2	Now running	Now running	Now running	Now running

Legend:

-- : No generation present

If there are no generations (including past results and future plans) inside the range that you are monitoring, JP1/AJS3 Console displays the status **Not scheduled to execute**. JP1/AJS3 Console does not display generations with the status **Waiting for start condition**.

■ Status levels displayed for jobnets with defined start conditions

If a jobnet being monitored has a defined start condition, the status is displayed as follows.

Table 7-7: Status display for jobnets with start conditions

Start condition fulfilled?	Status		Status displayed for AJS3 unit monitored object	
	Start condition	Root jobnet	Prioritize results	Prioritize schedules
Not fulfilled	Now monitoring	Wait for start cond.	Now monitoring	Now monitoring
	Unmonitored + Ended	--	Unmonitored + Ended	Unmonitored + Ended or Waiting for start time

Start condition fulfilled?	Status		Status displayed for AJS3 unit monitored object	
	Start condition	Root jobnet	Prioritize results	Prioritize schedules
	Monitor terminated	--	Monitor terminated	Monitor terminated or Waiting for start time
	Interrupted monitoring	--	Interrupted monitoring	Interrupted monitoring or Waiting for start time
Fulfilled	Now monitoring	Now running	Now running	Now running
		Wait for start cond.		Now monitoring
	Monitor terminated	Results	Now running	Results
		Wait for start cond.		Monitor terminated or Waiting for start time
	Interrupted monitoring	Results before Monitor terminated	Results	Results or Waiting for start time
		Results after Monitor terminated		Interrupted monitoring or Waiting for start time
	Monitor-end normal	Results	Now running	Now running
		Now running		Results or Waiting for start time

Legend:

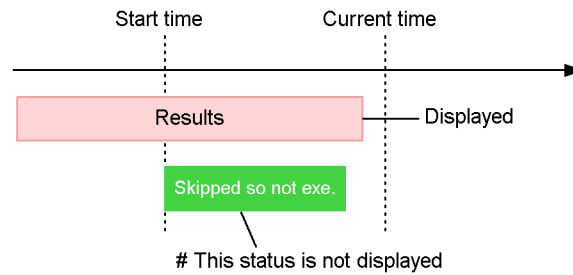
-- : No generation present

■ **Status displayed for jobnets that have schedule skip or multi-schedule as the schedule option**

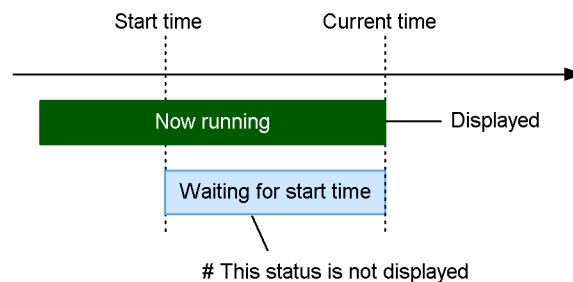
The following figure shows the status that is displayed for jobnets that have schedule skip or multi-schedule (concurrent exec. disabled) set as the schedule option.

Figure 7-15: Examples of status display for jobnets with schedule skip or multi-schedule (concurrent exec. disabled)

● Status display with the schedule skip option



● Status display with the multi-schedule (concurrent exec. disabled) option



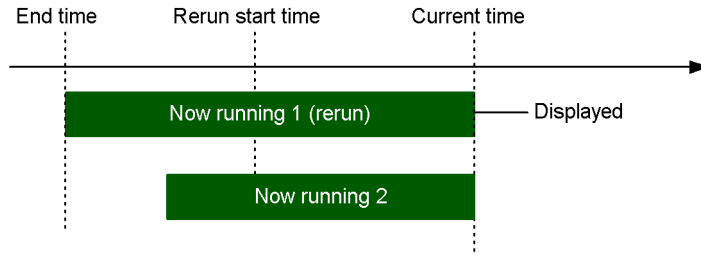
When the schedule option of a jobnet is set to schedule skip, the status of the skipped generations becomes **Skipped so not exe.** However, because the results of the generation preceding the skipped generation will be closer to the current time, the **Skipped so not exe.** status is not displayed.

Jobnets with multi-schedule (concurrent exec. disabled) set as the schedule option adopt the **Waiting for start time** status until the previous generation completes. However, since the previous generation has the **Now running** status, and hence higher display priority, the **Waiting for start time** status is not displayed.

■ Status displayed for rerun jobnets

If a jobnet that has ended is rerun, JP1/AJS3 Console determines the generation to display using the re-execution start times. The following figure shows the system behavior when a root jobnet is rerun after ending.

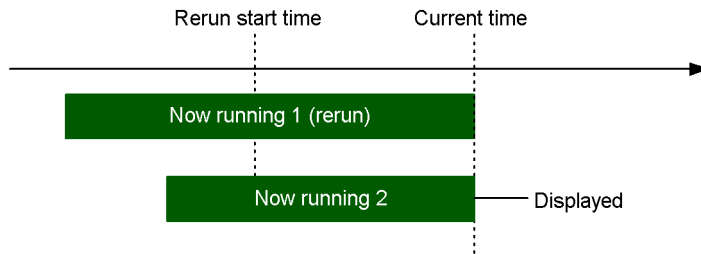
Figure 7-16: System behavior when a root jobnet is rerun after ending



In this case, the re-execution start time of **Now running 1** is closer to the current time than the start time of **Now running 2**. Therefore, the status of the **Now running 1** generation is displayed.

In contrast, if a jobnet is rerun while the status of the root jobnet is **Now running**, JP1/AJS3 Console determines the generation to display using execution start times. The following figure shows the system behavior when a root jobnet is rerun while running.

Figure 7-17: System behavior when a root jobnet is rerun while running



In this case, the start time of **Now running 2** is closer to the current time than the start time of **Now running 1**. Therefore, the status of the **Now running 2** generation is displayed.

(2) Displaying a hold plan

If you are monitoring a jobnet that has a hold attribute, you can define whether or JP1/AJS3 Console displays the status of the jobnet in a color that reflects the hold plan of the jobnet. JP1/AJS3 Console can indicate the hold plan of the jobnet when it is scheduled for execution.

You can choose from two methods of displaying a hold plan. The first method involves displaying a hold plan if the hold attribute of the root jobnet is set to **Hold**, or if you have set the hold attribute using the **Change Hold Attribute** function. The second method involves displaying a hold plan only if you have set the hold attribute of the root jobnet using the **Change Hold Attribute** function (when you have set **Hold** to **Yes**).

The following table shows whether JP1/AJS3 Console displays a hold plan, depending on the hold attribute of a jobnet.

Table 7-8: Displaying hold plans according to the hold attribute of a jobnet

Hold attribute	Hold attribute defined for root jobnet	Change hold attribute	
		Held	Not held
Do not hold	N	Y	N
Hold	C	Y	N
Hold if prev. = 'abend'	N	Y	N
Hold if prev. = 'warning' or 'abend'	N	Y	N

Legend:

Y: Display as hold plan

N: Do not display as hold plan

C: Do not display when setting is **Display Only a Change of Hold Attribute**

Supplementary note

The functionality for displaying a hold plan is not supported if you log in to JP1/AJS2 Console Manager from JP1/AJS2 Console 06-71, or if the manager host on which the monitored jobnet is defined is running JP1/AJS2 - Manager version 06-71 or earlier.

You can only use different colored icons to indicate the hold plan of an object when the object is an AJS3 unit monitored object. You cannot display icons that reflect the hold plan for a nested business scope.

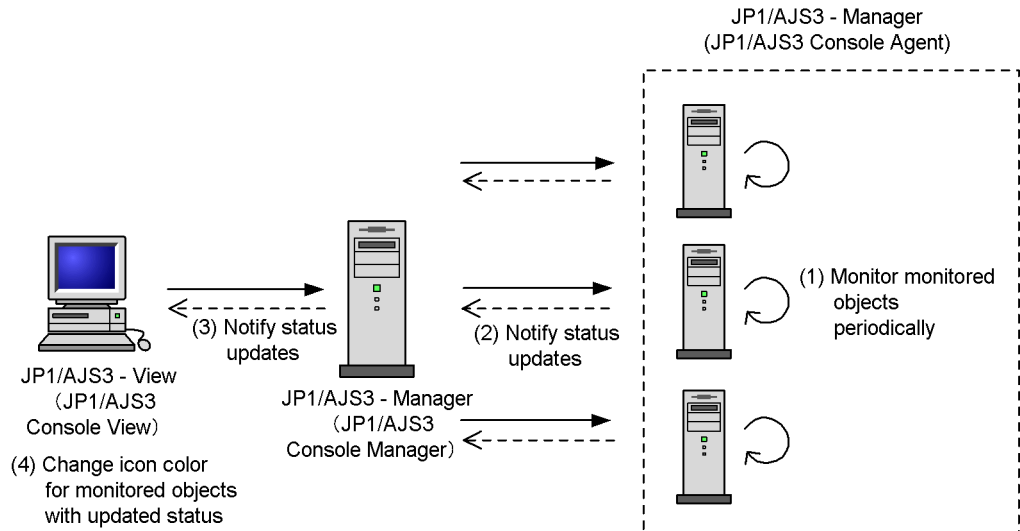
(3) Setting the monitoring interval

You can use the **Monitoring interval** setting to specify the interval, in seconds, at which JP1/AJS3 Console Agent sends status information to JP1/AJS3 Console Manager. The monitoring interval is set to 300 seconds by default.

In JP1/AJS3 Console, a status notification is sent whenever there is a status change in a monitored jobnet.

The flow of jobnet status notifications shown in the following figure.

Figure 7-18: Flow of status notification



Legend:

- ▶ : Monitoring
- - -> : Notification

Note: Names in parentheses are component names

First, JP1/AJS3 Console Agent periodically monitors the status of the monitored jobnet. If JP1/AJS3 Console Agent detects a change in the status of the jobnet, it notifies JP1/AJS3 Console Manager of the new status. JP1/AJS3 Console Manager then notifies JP1/AJS3 Console View of the new status information received from JP1/AJS3 Console Agent. JP1/AJS3 - View then updates the colors of the icons of the AJS3 unit monitored object and the business scope that includes the object, to reflect the new status.

7.3 Monitoring applications

Once you have created the objects to monitor and defined the monitoring properties, you can start monitoring applications using JP1/AJS3 Console. To start monitoring applications, switch the Main Scope window mode to the monitoring mode.

This section explains how the status of a monitored object is displayed, the on-screen operations used in the monitoring mode, and the operations you can perform on jobnets.

7.3.1 Updating statuses

Immediately after you log in, or immediately after you switch from **Definition mode** to **Monitoring mode**, the icon of each AJS3 unit monitored object is displayed in a color indicating that *no status is available for the object*. JP1/AJS3 Console View then begins to obtain status levels for each AJS3 unit monitored object, and updates the display to reflect the status levels as they are received.

When the status level of a monitored object is updated, there will be a time lag before the color of the AJS3 unit monitored object in the business scope is updated to reflect the change. The maximum length of the time lag is equivalent to the monitoring interval that you set for the monitoring method under monitoring properties (the default is 300 seconds).

You can update the status levels in JP1/AJS3 Console View at any time by choosing **View**, and then **Refresh**.

Supplementary note

The **Refresh** command is not supported if you log in to JP1/AJS2 Console Manager from JP1/AJS2 Console 06-71. If the host on which the monitored application is defined is running JP1/AJS2 - Manager version 06-71 or earlier, the status levels remain unchanged when you choose **Refresh**.

7.3.2 Status acquired by JP1/AJS3 Console

The status levels that JP1/AJS3 Console acquires for AJS3 unit monitored objects and business scopes are explained below.

(1) *Statuses for AJS3 unit monitored objects*

JP1/AJS3 Console can display the statuses shown in the list below for AJS3 unit monitored objects.

(a) **List of acquired statuses**

The following table shows the statuses for monitored jobnets that can be reported by JP1/AJS3 Console Manager and JP1/AJS3 Console View, and the corresponding default icon colors.

Table 7-9: Statuses of AJS3 unit monitored objects and corresponding colors

No.	Status	Color (default)
1	Not registered	None
2	Waiting for start time	Sky blue
3	Hold plan ^{#1}	Yellow
4	Being held	Yellow
5	Not sched. to exe.	Gray
6	Now running	Green
7	Running + Warning	Red
8	Running + Abend	Red
9	Ended normally	Light green
10	Ended with warning	Light red
11	Ended abnormally	Light red
12	Interrupted	Light red
13	Killed	Light red
14	Invalid exe. seq.	Light red
15	Skipped so not exe.	Gray
16	Shutdown	Brown
17	Now monitoring	Green
18	Wait for start cond.	Sky blue
19	Unmonitored + Ended	Gray
20	Monitor-end normal	Light green
21	Monitor terminated	Light red
22	Interrupted monitoring	Light green
23	No status ^{#2}	None
24	Unrecognizable ^{#3}	Gray
25	Unknown ^{#4}	Red

#1

A hold plan is only displayed when the function for displaying hold plans is enabled in the monitoring properties. Although color is used to indicate the hold plan, the hold plan is not displayed as a status name.

#2

This status indicates that no status has yet been received from JP1/AJS3 Console Agent.

#3

This status indicates that JP1/AJS3 Console Manager and JP1/AJS3 Console View are unable to recognize the status of a monitored jobnet.

#4

This status is displayed when any of the following is true:

- A jobnet specified in an AJS3 unit monitored objects is not a root jobnet.
- JP1/AJS3 Console could not connect to the host because the host name was invalid.
- A jobnet specified in an AJS3 unit monitored objects does not exist.
- You do not have permission to reference a jobnet specified in an AJS3 unit monitored objects.
- You do not have permission to access an upper-level unit for a jobnet specified in an AJS3 unit monitored objects.
- JP1/AJS3 Console cannot obtain the status of a jobnet since the scheduler service where the monitored jobnet is defined is undergoing maintenance.
- JP1/AJS3 Console cannot connect to the authentication server.
- JP1/AJS3 Console cannot obtain the status levels because JP1/AJS3 Console Agent is not running.
- JP1/AJS3 Console cannot connect to JP1/AJS3 Console Agent.
- A problem occurred that prevents JP1/AJS3 Console Agent from acquiring statuses.

If a status is displayed as **Unknown**, you can obtain more details in the Detailed Information - [AJS3 Unit Monitored Objects] dialog box.

(b) Delay status

With JP1/AJS3 Console, you can also monitor the delay status of a jobnet. The jobnet delay status is shown in parentheses after a status name listed in Table 7-9 (for example, **Now running (Delayed start)**).

There are four delay status levels that apply to AJS3 unit monitored objects.

- **Delayed end**
- **Delayed start**
- **Nested jobnet delayed end**
- **Nested jobnet delayed start**

(2) Statuses for business scopes

Business scopes can take the same status as AJS3 unit monitored objects. The only difference is that hold plans do not affect the display color for a business scope. The status displayed for a business scope is that of the lower-level business scope or AJS3 unit monitored object whose status has the highest priority.

(a) Acquired statuses and their priority

The order of the different statuses that can be displayed for a business scope are shown in the following table.

Table 7-10: Order of priority of statuses for business scopes

Priority	Status
1	Unknown
2	Shutdown
3	Running + Abend
4	Ended abnormally
5	Killed
6	Interrupted
7	Invalid exe. seq.
8	Running + Warning
9	Ended with warning
10	Being held
11	Now running
12	Wait for start cond.
13	Now monitoring
14	Waiting for start time
15	Skipped so not exe.

Priority	Status
16	Unmonitored + Ended
17	Monitor terminated
18	Interrupted monitoring
19	Ended normally
20	Monitor-end normal
21	Not sched. to exe.
22	Not registered
23	Unrecognizable
24	No status

(b) Order of priority of delay statuses

The delay status displayed for a business scope is that of the lower-level business scope or AJS3 unit monitored object whose delay status has the highest priority. The order of priority of the delay statuses is shown in the following table.

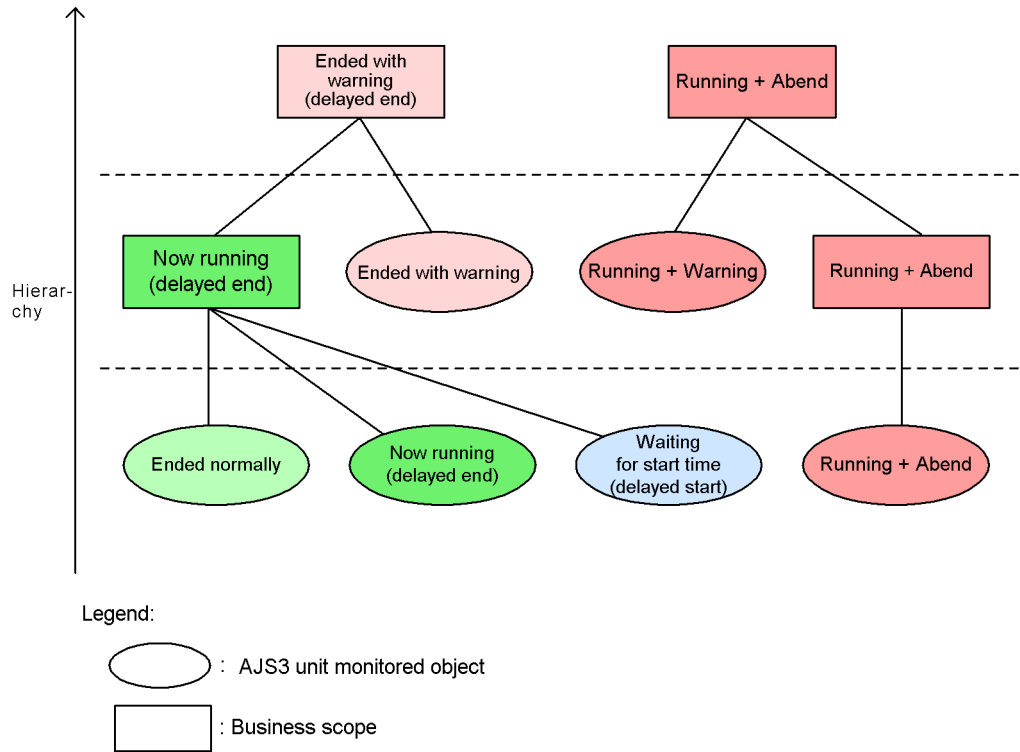
Table 7-11: Order of priority of delay statuses for business scopes

Priority	Delay status
1	Delayed end
2	Nested jobnet delayed end
3	Delayed start
4	Nested jobnet delayed start
5	Not delayed

(c) Status display example for a business scope

The following figure shows an example of the status display for a business scope.

Figure 7-19: Status display example for a business scope



In the case of a business scope, the statuses of all lower-level business scopes and AJS3 unit monitored objects are merged, and the upper-level business scope displays the status with the highest priority.

7.3.3 Manipulating monitored jobnets

In JP1/AJS3 Console, you can manipulate monitored jobnets by carrying out any of the following operations:

- Register a jobnet for execution (register a jobnet for planned execution)
- Cancel registration of a jobnet
- Cancel a hold attribute for a jobnet
- Rerun a jobnet (from a job that ended abnormally)

To perform these operations, the JP1 user who is logged in must have permission to manipulate the monitored jobnets.

For details about user management when using JP1/AJS3 Console, see *8.3 User management using JP1/AJS3 Console*.

Supplementary note

These operations are not supported if you log in to JP1/AJS2 Console Manager from JP1/AJS2 Console 06-71. If the host on which the monitored jobnet is defined is running JP1/AJS2 - Manager version 06-71 or earlier, these operations result in an error.

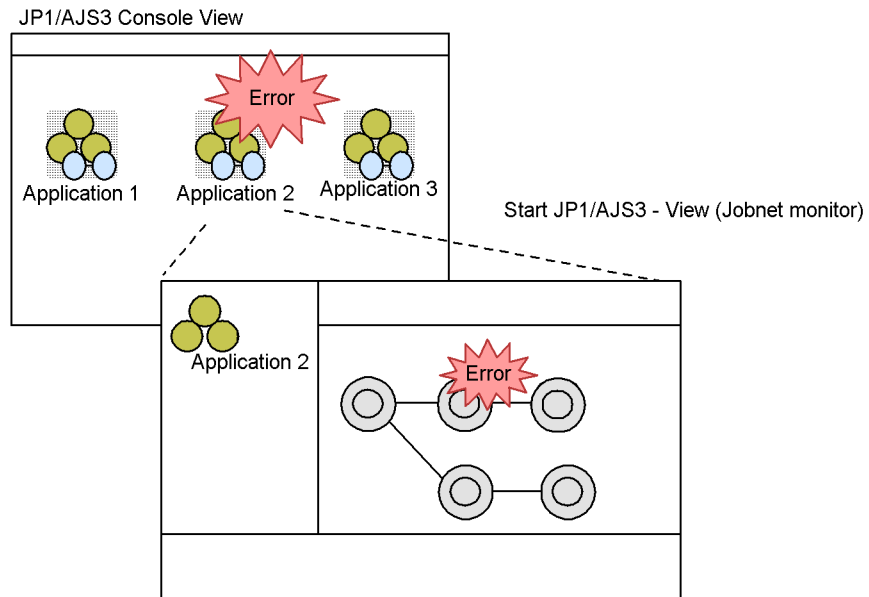
7.3.4 Starting JP1/AJS3 - View

You can start JP1/AJS3 - View from within JP1/AJS3 Console View.

If an error is detected in a jobnet that you are monitoring, you can start JP1/AJS3 - View directly from the monitored object, and check detailed information for the job where the error occurred. This enables you to easily identify the source of the problem.

The following figure shows the action of starting JP1/AJS3 - View.

Figure 7-20: Starting JP1/AJS3 - View



Chapter

8. User Management and Access Control

In JP1/AJS3, you can use the user authentication function of JP1/Base to manage the login authentication and operational permission of users. In addition, by setting the operational permission of JP1 users for each unit, you can prevent specific users from accessing a unit without authorization.

This chapter describes user management using the user authentication function of JP1/Base. This chapter also describes how to set access control for each unit within JP1/AJS3.

- 8.1 User management using the functions of JP1/Base
- 8.2 Settings for restricting access to units
- 8.3 User management using JP1/AJS3 Console

8.1 User management using the functions of JP1/Base

In JP1/AJS3, you can use the user authentication function and the user mapping function of JP1/Base to manage the login authentication and operational permission of users.

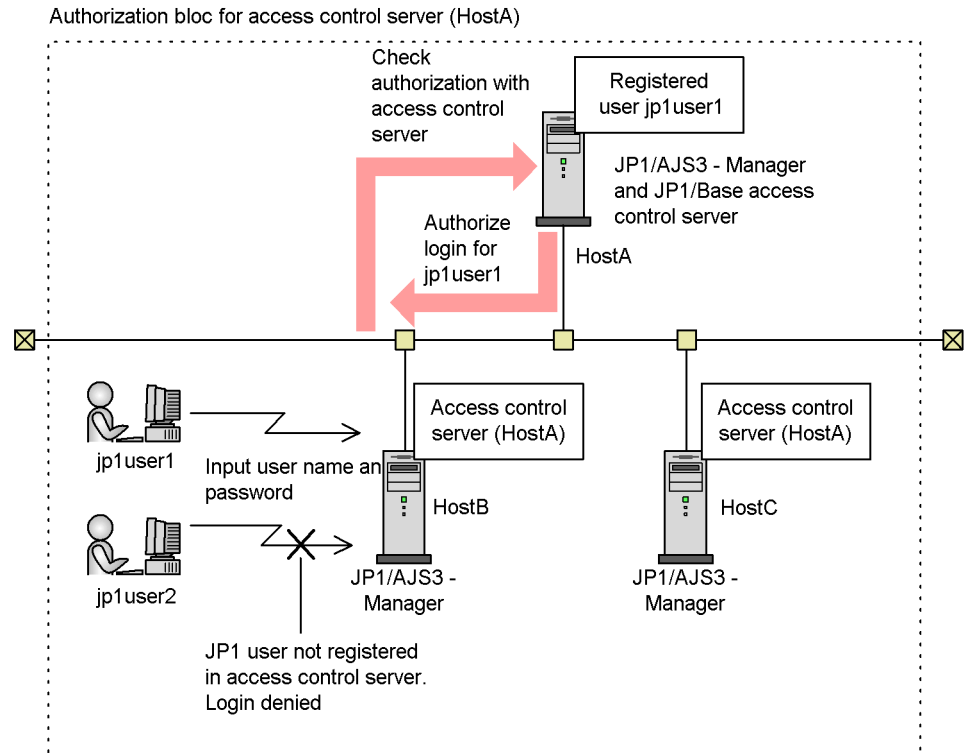
This section provides an overview of each of these functions and the setting of those functions. For details on settings and operational methods within JP1/Base, see the *Job Management Partner 1/Base User's Guide*.

8.1.1 User management using the JP1/Base user authentication function

In JP1/AJS3, you can use the JP1/Base user authentication function to manage the login authentication and operational permission of users.

The JP1/Base user authentication function manages the login authentication of users from JP1/AJS3 - View or other JP1 series programs (such as JP1/IM), and controls the operational permission levels of users who are logged in. The JP1/Base that manages login authentication and controls the operational permission levels of users who are logged in is called the *authentication server*. You register the users who use JP1/AJS3 in this authentication server as JP1 users, and then set operational permission for the units for each of these JP1 users. For copies of JP1/Base installed on a different server from the authentication server, you must define the host that is used as the authentication server. When a user attempts to log in to another host using JP1/AJS3 - View, the ability of the user to log in, and the access permission available to the user is determined by the JP1 user information registered in the authentication server. An example of user authentication is shown below.

Figure 8-1: Example of user authentication



In this example, HostA is defined as the authentication server. In HostB and HostC, HostA is specified as the authentication server. Hence HostA, HostB and HostC function as a single authentication bloc. A user called jp1user1 is registered as a JP1 user in the authentication server of HostA. In the case shown, the JP1 user called jp1user1 and another JP1 user called jp1user2 attempt to log in to HostB. HostA, which functions as the authentication server for HostB, determines whether each user has login permission based on the registered JP1 user information. In the example shown, jp1user2 is not registered in the authentication server, and so login permission is denied.

(1) Registering JP1 users

Users who use JP1/AJS3 and other JP1 series programs are called *JP1 users*. You register JP1 users in the authentication server. To register a JP1 user, you specify a JP1 user name and a password to be used by the JP1 user at login.

JP1 users registered in this manner are able to use not only JP1/AJS3, but also other JP1 series programs (such as JP1/IM).

(2) Setting access permission

Operational access to units within JP1/AJS3 is called access permission. You can set access permission for each JP1 user.

You set access permission by setting the operational permission, known as the JP1 permission level, for a series of groups known as JP1 resource groups.

There are three different types of JP1 permission level:

- Access permission for defining and executing jobnets.
- Access permission for executing and operating jobs.
- Access permission for agent management information

An explanation of each type of JP1 permission level is given below.

Access permission for defining and executing jobnets

- JP1_AJS_Admin

Administrator's permission. This permission level allows you to alter unit owners and the operational permission levels for resource groups. You can also define, execute and edit jobnets.

- JP1_AJS_Manager

This permission level allows you to define, execute and edit jobnets.

- JP1_AJS_Editor

This permission level allows you to define and edit jobnets.

- JP1_AJS_Operator

This permission level allows you to execute and reference jobnets.

- JP1_AJS_Guest

This permission level allows you to reference jobnets.

Access permission for executing and operating jobs

- JP1_JPQ_Admin

Administrator's permission. This permission level allows you to set job execution environments, operate queues and agents that execute jobs, and operate jobs that have been queued by other users.

- JP1_JPQ_Operator

This permission level allows you to operate queues and agents that execute jobs, and operate jobs that have been queued by other users.

- JP1_JPQ_User

This permission level allows you to register submitted jobs, and operate jobs that you have queued.

Access permission for agent management information

- JP1_JPQ_Admin
Administrator's permission. This permission level allows you to add, change, and delete the definitions of execution agents and execution agent groups.
- JP1_JPQ_Operator
This permission level allows you to change the job transfer restriction status for execution agents and execution agent groups.
- JP1_JPQ_User
This permission level allows you to view the status and definitions of execution agents and execution agent groups.

For details about each JP1 permission level, see 6.4(2) *Determining JP1 permission levels* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

A JP1 user mapped to an OS user who is a member of the Administrators group (Windows) or a superuser (UNIX) can perform all operations that require JP1_AJS permissions (not JP1_JPQ permissions) regardless of the granted JP1 permissions. For details on the mapping of OS users, see 8.1.2 *User management using the user mapping function of JP1/Base*.

A JP1 resource group is set for each unit within JP1/AJS3 as a way of controlling access to each unit by JP1 users.

For example, assume that a JP1 resource group called keiri has been set for a unit called jobnet A. Furthermore, assume that in the authentication server, the JP1 user called jp1user1 has a JP1 permission level set to JP1_AJS_Operator for the resource group keiri, and a JP1 permission level set to JP1_AJS_Editor for the resource group called eigyo. In this case, the JP1 user called jp1user1 can perform operations on the jobnet A at the permission level of JP1_AJS_Operator set for the resource group keiri. In other words, jp1user1 can register the jobnet A for execution, cancel a registration of the jobnet A for execution, change the schedule, or change the status of a job. However, jp1user1 cannot change the definition of the jobnet A, nor delete the jobnet. In contrast, if the JP1 resource group eigyo were set for the jobnet A, jp1user1 could change the definition of the jobnet A or delete the jobnet, but could not register the jobnet A for execution, cancel a registration of the jobnet A for execution, nor change the status of a schedule or a job. If the JP1 resource group called jinji were set for the jobnet A, the user jp1user1 would have no permission in relation to the jobnet A, and would therefore be unable to access the jobnet. However, if jp1user1 logged on as a member of the Administrators group (in the case of a Windows user) or had superuser privileges (in the case of a UNIX user), then jp1user1 could perform operations

regardless of the JP1 permission level of the JP1 resource group.

In this manner, controlling the access of JP1 users to each of the units within JP1/AJS3 is achieved by setting a resource group for each JP1/AJS3 unit. If you have not set a JP1 resource group for a unit then you cannot achieve access control using JP1 user permission levels.

8.1.2 User management using the user mapping function of JP1/Base

The JP1/Base user mapping function allows you to manage the operational permission of users in JP1/AJS3.

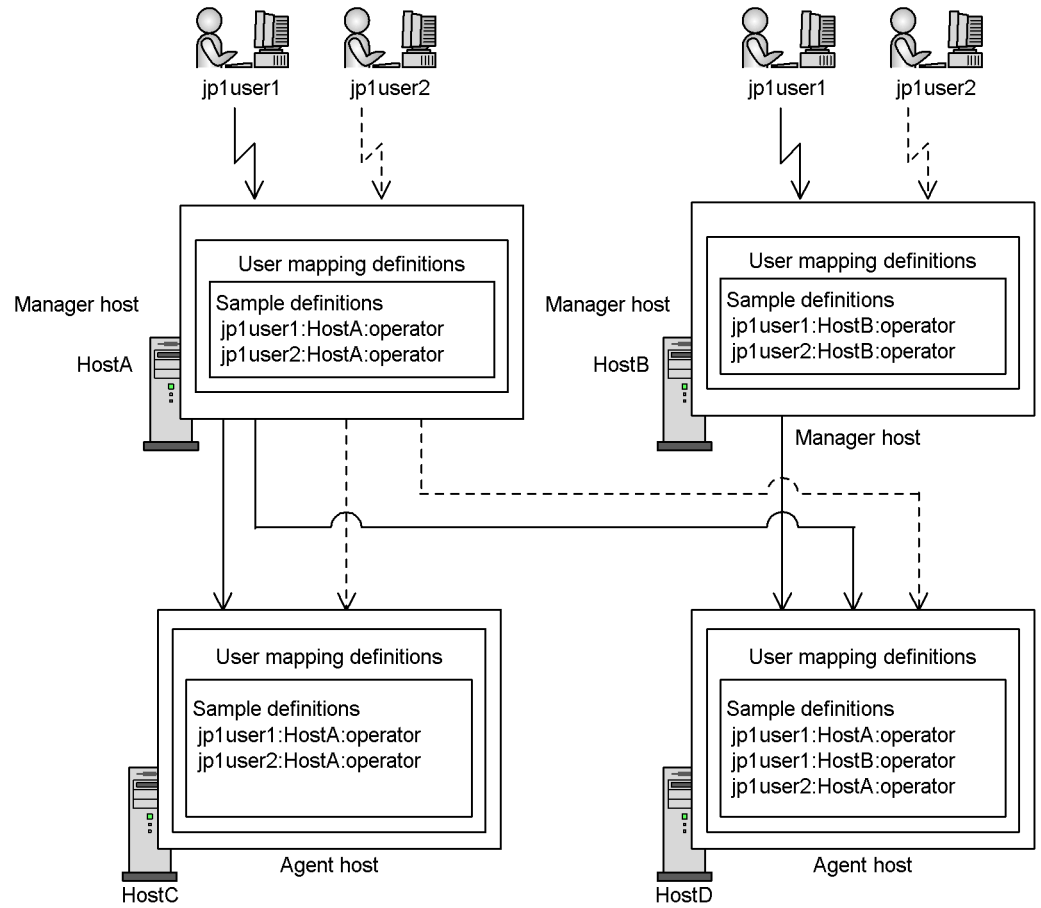
When a job or command is executed, user mapping compares the JP1 user at the source host where the execution request originated with the OS users registered in the OS of the execution target host.

User mapping is defined in the JP1/Base environment settings on each host. User mapping at a manager host maps OS users registered in the manager host to JP1 users who log in to the manager host. User mapping for an agent host specifies the manager host where the execution request originated as the server host, and maps the OS users of the agent host to the JP1 users.

Consequently, when a user at a manager host wants to execute a jobnet or a command at an agent host or another manager host, operations at each agent host are conducted in accordance with the user mapping definitions at that local host.

An example of user mapping is shown in the following figure.

Figure 8-2: Example of user mapping



Legend:

- > : `jp1user1` executes jobnet
- - -> : `jp1user2` executes jobnet

Sample user mapping definitions
 JP1 user name:server host name:mapped OS user

In the example shown, if the JP1 user called `jp1user1` executes a jobnet at HostA, and the execution target host is HostC, then the HostC OS user called operator, specified by the user mapping definitions at HostC, executes the jobs. Similarly, if the execution target host is HostD then the HostD OS user called operator, specified by the user mapping definitions at HostD, executes the jobs. In the same way, if `jp1user2` executes

a jobnet at HostA, then the OS users specified by the user mapping definitions at HostC and HostD execute the jobs. In contrast, if jp1user1 executes a jobnet at HostB, then the jobs can only be executed if HostD is the execution target host, because the user mapping at HostD also includes a user mapping specifying HostB as the server host. The jp1user1 in the HostC user mapping definitions does not specify jp1user1 of HostB, and so the jobs cannot be executed at HostC. If jp1user2 executes a jobnet at HostB, then the jobs cannot be executed at HostC or HostD, as neither host has a mapping definition for jp1user2 of HostB.

8.2 Settings for restricting access to units

This section describes how to set operational permission levels of JP1 users for each unit, to prevent unauthorized access to a jobnet or job.

For details on how to perform each of the settings, see the *Job Management Partner I/Automatic Job Management System 3 Operator's Guide*.

8.2.1 Unit owner permission

The JP1 user who defines a job or jobnet has owner permission for that unit. The JP1 user with owner permission can change the name of the JP1 resource group, the owner, and the type of user who can execute jobs (the execution user), regardless of the JP1 permission level. However, if a JP1 resource group without reference permission for that unit is set, even the owner of the unit cannot change the JP1 resource group name, the owner, or the type of job execution user. This is because the Detailed Definition dialog box for the unit cannot be opened in JP1/AJS3 - View. If the unit owner wants to change such information items, another JP1 user must set a JP1 resource group that grants the unit owner reference permission.

In addition, if no owner is set for a unit, then any user can change the name of the JP1 resource group, the owner, or the execution user.

If the type of the job execution user is set to **User who owns** and a JP1 user other than those listed below changes the job owner, the type of the job execution user changes to **User who registered**:

- In Windows, a JP1 user mapped to an OS user who is a member of the Administrators group
- In UNIX, a JP1 user mapped to an OS user who has superuser permission
- A JP1 user who has the JP1_AJS_Admin permission for the JP1 resource group set for the job

When the execution user is set to **User who registered**, the job specifies the JP1 user who registered the jobnet for execution as the execution user. This is to prevent an arbitrary user from executing the job at any user permission level.

8.2.2 Setting the JP1 resource group

For each JP1 user, you set the permission levels for units in JP1/Base, by defining a permission level for each JP1 resource group. The permission settings you specified are enabled by setting a JP1 resource group for each unit. However, if a JP1 user mapped to an OS user who is a member of the Administrators group (Windows) or a superuser (UNIX) is the primary user, all the JP1_AJS permissions are granted to the JP1 user, regardless of the JP1 resource group settings.

If you do not set a JP1 resource group for a unit, any JP1 user can perform any operation on the unit.

In manager job groups and manager jobnets, JP1 resource groups that are set at the reference target manager, and the permission levels for those JP1 resource groups, apply.

8.2.3 Setting the Executed by attribute

The **Executed by** attribute defines the user who can execute the jobs defined in the jobnet (the execution user) when you execute the jobnet.

There are two settings for the **Executed by** attribute defining the execution user.

- **User who registered**

The JP1 user who registered the jobnet for execution is the execution user. At the target host where the job is executed, the OS user that maps to the JP1 user who registered the jobnet for execution executes the job.

- **User who owns**

The JP1 user defined as the job owner is the execution user. At the target host where the job is executed, the OS user who maps to the JP1 user defined as the job owner executes the job.

In the case of an event job, because event jobs are executed with the user permission of the account from which JP1/AJS3 was started, the **Executed by** attribute is ignored if set. In the cases of an OR job or a judgment job, you cannot specify the user who executes the job. These jobs are executed under the user permissions of the user who is running JP1/AJS3.

You must map the JP1 user who corresponds with the user defined by the **Executed by** attribute (**User who registered** or **User who owns**) to a user at the target host where the job is executed.

8.2.4 Setting the User name (for PC jobs and Unix jobs)

When PC jobs or Unix jobs are defined in a jobnet, the user name specifies the OS user at the target host (the agent host) that executes these jobs. You can only specify a setting for the **User name** for PC jobs and Unix jobs.

The OS user you specify under the **User name** setting is the OS user that executes the job at the target host for the user selected in the **Executed by** attribute (**User who registered** or **User who owns**). If you do not specify a **User name**, the primary OS user that maps to the JP1 user (the execution user) executes the job at the target host.

You must map the user you specify under the **User name** setting to the JP1 user specified in the **Executed by** attribute (**User who registered** or **User who owns**).

Supplementary note

In the case of PC jobs, the only OS user information that is applied is the permission specified for the OS user. You can access the specified executable file through the account defined using JP1/AJS3 services. You can access other files from the executable file using the specified OS user. If you want to restrict the access privileges of the execution user to those of a local host user, specify the OS user name using a host name\user name type naming method.

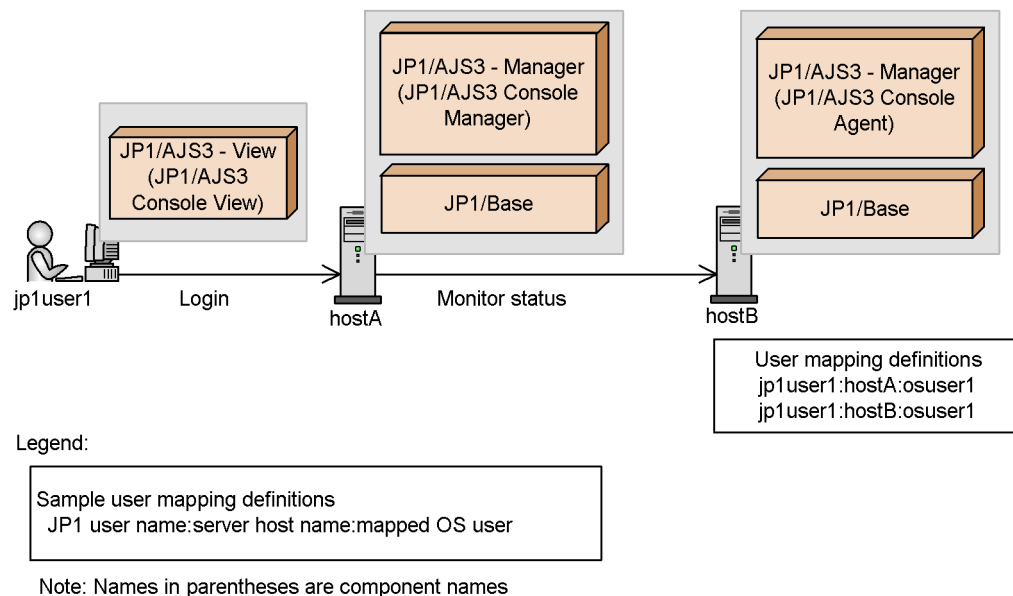
8.3 User management using JP1/AJS3 Console

To use JP1/AJS3 Console to monitor applications you must first perform the following:

- Register in the authentication server the JP1 user or users who log in with JP1/AJS3 Console.
- Set the access permission levels for the target units you want to monitor.
- Carry out user mapping at the monitoring target host (the JP1/AJS3 - Manager host where the target units for monitoring are defined).

An example of user management using JP1/AJS3 Console is shown in the following figure.

Figure 8-3: User management using JP1/AJS3 Console



To enable a JP1 user called `jp1user1` to use JP1/AJS3 Console to monitor a jobnet, perform the following:

- Register `jp1user1` in the authentication server.
- Set the access permission (the JP1 resource group and the JP1 permission level) required to monitor the status of the target unit, or carry out operations on the target unit.

- Carry out user mapping at the monitoring target host (HostB) to map OS users to jpluser1 specifying HostA as the server host.

Supplementary note

If you do not want to monitor jobnets defined on the JP1/AJS3 Console Manager host, there is no need to carry out user mapping at the JP1/AJS3 Console Manager host. If the authentication bloc of the JP1/AJS3 Console Manager host and the JP1/AJS3 Console Agent host differ, there is no need to set JP1 permission levels in the authentication server at the JP1/AJS3 Console Manager host.

For more details on user management using JP1/AJS3 Console, see *6.6 User management when monitoring work tasks centrally* in the *Job Management Partner 1/ Automatic Job Management System 3 System Design (Work Tasks) Guide*.

Chapter

9. Linking JP1/AJS3 with Other Products

As well as the basic operations described in Part 2 of this manual, JP1/AJS3 supports a range of functions implemented by linking with other products. This chapter describes JP1/AJS3 linkage.

9.1 Functions implemented by JP1/AJS3 product linkage

9.1 Functions implemented by JP1/AJS3 product linkage

The table below describes the products that can be linked with JP1/AJS3 and the functionality implemented in JP1/AJS3 through product linkage. For details on each function, see the *Job Management Partner 1/Automatic Job Management System 3 Linkage Guide* and the documentation for the linked product.

Table 9-1: Programs that can be linked with JP1/AJS3 and implemented functionality

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
1	ERP systems <ul style="list-style-type: none"> • SAP R/3 systems • SAP BW systems • Oracle E-Business Suite systems 	<p>Run ERP applications automatically:</p> <p>You can use JP1/AJS3 to automatically run processes created using SAP R/3 systems or Oracle E-Business Suite systems. By using the calendar function or job scheduling function of JP1/AJS3, you can accurately schedule and run applications from ERP systems.</p> <p>You define an ERP system application in a jobnet as an ERP system custom job. By associating an ERP system custom job with other jobs, you can automate many processes. For example, you can execute an ERP application when a specified event occurs, or automatically execute a recovery job if an error occurs in an ERP application.</p> <p>Linking SAP R/3 systems or SAP BW systems with JP1/AJS3 requires JP1/AJS3 for Enterprise Applications. JP1/AJS3 for Enterprise Applications is the new name for JP1/Application Manager for R/3(R) from JP1 Version 6.</p> <p>Linking Oracle E-Business Suite systems with JP1/AJS3 requires JP1/AJS2 for Oracle E-Business Suite. JP1/AJS2 for Oracle E-Business Suite is the new name for JP1/Application Manager for Oracle E-Business Suite from JP1 Version 6.</p>	<p>For details on linking JP1/AJS3 with an ERP system, see 5. <i>Linking ERP Systems</i> in the <i>Job Management Partner 1/Automatic Job Management System 3 Linkage Guide</i>.</p>
2	HP NonStop Server	<p>Execute processing with HP NonStop Server (Windows only):</p> <p>By linking JP1/AJS3 with HP NonStop Server, you can submit jobs to HP NonStop Server and execute jobs received from HP NonStop Server. You can check the execution results of the jobs in JP1/AJS3.</p> <p>To submit a job to HP NonStop Server from JP1/AJS3, queue the job as a QUEUE job.</p> <p>You must install the <i>Job Scheduler for MPP JP1 Linkage Option</i> in HP NonStop Server.</p>	<p>For details on linking JP1/AJS3 with HP NonStop Server, see 7. <i>Linking hp NonStop Server (Windows Only)</i> in the <i>Job Management Partner 1/Automatic Job Management System 3 Linkage Guide</i>.</p>

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
3	JP1/AJS3 - Definition Assistant	<p>Define jobnets and jobs as a batch operation: JP1/AJS3 - Definition Assistant allows you to export jobnets or jobs defined in the manager host to an Excel file, or to import them in batch from an Excel file into the manager host.</p> <p>JP1/AJS3 - Definition Assistant facilitates editing of the definitions of many hierarchical jobnets and jobs. Since the listing of the information about the defined jobnets and jobs is saved in an Excel file, updates to the definitions can be managed easily.</p>	<p>For details on linking JP1/AJS3 with JP1/AJS3 - Definition Assistant, see <i>Job Management Partner 1/ Automatic Job Management System 3 - Definition Assistant Description, Operator's Guide and Reference</i>.</p>
4	JP1/AJS3 - Web Operation Assistant	<p>Monitor and manipulate applications on different servers from a Web browser: You can use JP1/AJS3 - Web Operation Assistant to monitor all the jobs on the various servers in the system, using a Web browser. From the displayed information, you can also select and operate on units. Deploying JP1/AJS3 - Web Operation Assistant makes system monitoring easier.</p> <p>Using the filtering function, you can limit the information to be monitored and list only the relevant results.</p>	<p>For details on linking JP1/AJS3 with JP1/AJS3 - Web Operation Assistant, see <i>Job Management Partner 1/ Automatic Job Management System 3 - Web Operation Assistant Description, Operator's Guide and Reference</i>.</p>

9. Linking JP1/AJS3 with Other Products

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
5	HP NNM	<p>Monitor the execution status of JP1/AJS3 processes: You can use HP NNM version 7.5 or earlier to monitor the operation status of the manager hosts on a network, the execution status of jobs, and the operation status of the agent hosts. By using symbols for users, you can also monitor any desired target status.</p> <p>In the HP NNM window, the execution status levels of jobs are distinguished by the color of the associated symbol.</p> <p>JP1/AJS3 cannot be linked with HP Network Node Manager i Software v8.10 or later.</p>	<p>For details on linking JP1/AJS3 with HP NNM, see the following:</p> <ul style="list-style-type: none"> • <i>A. Monitoring Jobnets Using HP NNM in the Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i> • <i>B. Events Reported to HP NNM in the Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i> • <i>C. Commands Used When Linking with HP NNM in the Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i>
6	JP1/FTP	<p>Automate file transfer jobs: You can use JP1/FTP to define file transfer via FTP as a job in a jobnet. By defining a file transfer as a job in a jobnet, you can transfer files according to a schedule, or execute other processes when a file transfer occurs.</p> <p>A standard custom job for JP1/FTP is available in JP1/AJS3. You can execute file transfers by defining this JP1/FTP custom job in a jobnet.</p>	<p>For details on linking JP1/AJS3 with JP1/FTP, see <i>4. Linking JP1/FTP in the Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i>.</p>

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
7	JP1/IM	<p>Monitor the execution status of JP1/AJS3 processes: You can use JP1/IM to monitor the execution status of processes in JP1/AJS3. JP1/IM is a program that uses JP1 events to provide integrated management of JP1-series programs.</p> <p>Using JP1/IM, you can monitor JP1/AJS3 in the following ways:</p> <ul style="list-style-type: none"> • Monitoring JP1 events The Main Console window of JP1/IM displays information on the JP1 events that are output by JP1-series programs. JP1 events output by JP1/AJS3 are also displayed in the Main Console window. You can limit the JP1 events that are displayed by choosing to display only important events, or only those events originating from specified hosts. • Monitoring execution status icons The tree monitoring windows and visual monitoring windows in JP1/IM use icons to represent the execution status of the various jobnets and jobs. When an error or other event occurs, the icon changes color. <p>During monitoring in JP1/IM, if you want to check detailed information on the execution status, or change the definition of a jobnet, you can call up the JP1/AJS3 - View window from the JP1/IM window.</p>	<p>For details on linking JP1/AJS3 with JP1/IM, see <i>10. Monitoring Jobnets Using JP1/IM</i> in the <i>Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i>.</p>
8	JP1/NQSEXEC	<p>Execute processing with JP1/NQSEXEC: By linking JP1/AJS3 with JP1/NQSEXEC, you can submit jobs to JP1/NQSEXEC, and execute jobs received from JP1/NQSEXEC. When you submit jobs to JP1/NQSEXEC from JP1/AJS3, JP1/NQSEXEC automatically queues the jobs and executes them. You can check the execution results of the jobs in JP1/AJS3. You can queue jobs from JP1/AJS3 to JP1/NQSEXEC as either QUEUE jobs or submitted jobs.</p>	<p>For details on linking JP1/AJS3 with JP1/NQSEXEC, see <i>9. Setup for Linking JP1/NQSEXEC</i> in the <i>Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i>.</p>

9. Linking JP1/AJS3 with Other Products

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
9	JP1/OJE	<p>Execute processing in conjunction with a mainframe: Using JP1/OJE you can submit jobs to a mainframe and execute jobs received from a mainframe.</p> <p>JP1/OJE can link with the following mainframes:</p> <ul style="list-style-type: none"> • VOS3 • Another vendor's mainframe (MVS, OSIV/MSP, etc.) • AS/400 <p>When you submit jobs to JP1/OJE from JP1/AJS3, JP1/OJE automatically queues the jobs and executes them. You can check the execution results of the jobs in JP1/AJS3.</p> <p>To execute a mainframe job from JP1/AJS2, you run it as a QUEUE job or submit job.</p>	<p>For details on linking JP1/AJS3 with JP1/OJE, see <i>6. Linking Mainframes Using JP1/OJE</i> in the <i>Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i>.</p>
10	JP1/Script	<p>Create and execute high-level scripts:</p> <p>By using JP1/Script, you can execute high-level scripts in a Windows or UNIX environment. JP1/Script is a scripting language that enables you to construct high-level scripts with ease. In JP1/AJS3, you can define script files created with JP1/Script 06-00 or later as jobs, and then execute them.</p> <p>By linking JP1/AJS3 with JP1/Script, you can display GUIs from within JP1/AJS3 jobs. In JP1/AJS3, usually you cannot execute programs that display a GUI and wait for input as jobs. However, by running a program that displays a GUI from a JP1/Script file, you can operate the GUI in exactly the same manner as if it were run from the desktop. Attempting to display a GUI from a JP1/AJS3 job without using JP1/Script is not recommended.</p> <p>Linking with JP1/Script requires no special setup within JP1/AJS3.</p>	<p>For details on linking JP1/AJS3 with JP1/Script, see <i>8. Linking JP1/Script</i> in the <i>Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i>.</p>
11	JP1/Software Distribution	<p>Automatically distribute JP1/AJS3 definition information to other hosts:</p> <p>By using JP1/Software Distribution you can distribute defined jobnets and calendars to a multitude of hosts, and register them automatically. This significantly reduces the administrator's workload.</p> <p>JP1/Software Distribution is a program that performs automatic batch distribution of software and files. JP1/Software Distribution enables you to manage all facets of the distribution process. You can specify the name of the folder into which the distributed files are to be placed, and specify a schedule for the distribution.</p> <p>When JP1/AJS3 is used in a large-scale system, JP1/Software Distribution facilitates the task of defining jobnets and calendars on multiple hosts.</p>	<p>For details on linking JP1/AJS3 with JP1/Software Distribution, see <i>11. Automatic Distribution Using JP1/Software Distribution</i> in the <i>Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i>.</p>

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
12	TELstaff	<p>Automatically report errors: By linking JP1/AJS3 with TELstaff, you can automatically notify the administrator via mobile phone, pager, or a signal light if an error occurs during execution of a jobnet.</p> <p>To execute this type of automatic notification, you define TELstaff commands within JP1/AJS3 jobs. Linking with TELstaff requires no special setup within JP1/AJS3.</p>	For details on linking JP1/AJS3 with TELstaff, see your TELstaff manual.
13	Mail system	<p>Define the sending and receiving of email in a jobnet: By linking JP1/AJS3 with an email system, you can define the sending and receiving of email in a jobnet. This enables you to, for example, send email to the system administrator if an error occurs, or start a jobnet on receipt of email.</p> <p>To use mail system linkage under Windows, you will require MAPI compatible client software such as Outlook. In the case of UNIX, an SMTP mail server with the sendmail function is required. With UNIX, you can send and receive RFC822 compliant email.</p> <p>To define and execute email functions within jobnets, use an email sending job to send email, or an email reception monitoring job to monitor incoming mail.</p>	For details on linking JP1/AJS3 with a mail system, see 2. <i>Linking Mail Systems</i> in the <i>Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide</i> .
14	Message queue system	<p>Define the sending and receiving of messages in a jobnet: By linking JP1/AJS with a message queue system, you can define the sending and receiving of messages related to the message queue system within a jobnet. This enables you to, for example, send a message to the message queue system if an error occurs, or start a jobnet on receipt of a message from the message queue system.</p> <p>Linkage with a message queue system is not supported in version 9 or later. To enable linkage, a JP1/AJS version that supports this functionality is required on the execution host side.</p>	For the message queue systems that can be linked with JP1/AJS, see the manual for the JP1/AJS version installed on the execution host side.

Chapter

10. Supported JP1/AJS2 Functionality

This chapter describes the JP1/AJS2 functionality that is supported in JP1/AJS3.

- 10.1 Using a planning group to change the plans for root jobnets
- 10.2 Remote jobnet
- 10.3 Manager jobnet
- 10.4 QUEUE jobs and submit jobs
- 10.5 Queueless jobs

10.1 Using a planning group to change the plans for root jobnets

A planning group is a function of JP1/AJS3 that allows you to handle a group of root jobnets with different schedules as a single jobnet. By using a planning group, you can change the content of jobnets while the system is running.

We recommend that you use planning groups if your intended operation includes any of the following:

- You expect that you will want to change the definition of jobnets in the future, but will not want to halt system operations to do so.
- You want to define a changed version of a jobnet in advance, and then switch to the changed version at a specific time.
- You want to use the concurrent execution or schedule skip functions on multiple related jobnets, treating them as if they were a single jobnet.
- You want to execute multiple related jobnets by switching between them, and prevent the next jobnet from executing when a previous jobnet ends abnormally or ends with a warning.

For specific examples of planning group configurations, see *(4) Examples of configuring planning group definitions*.

(1) Functions of planning groups

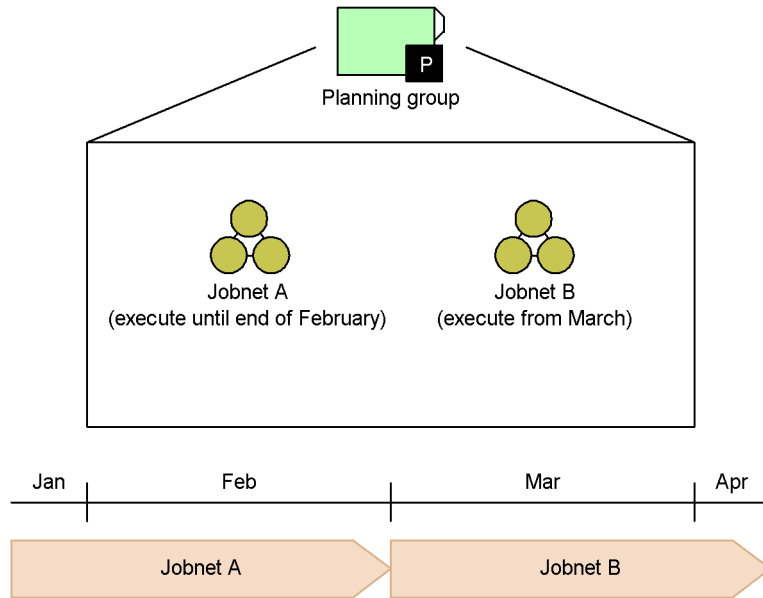
By using a planning group, you can switch smoothly between different root jobnets during system operation, without having to use a different name for each root jobnet in the GUI or in a command.

First, you define the specific applications (processes) that you want to execute in each of the multiple root jobnets that you will be switching between. For example, you could define applications for calculating order data and creating order forms in one jobnet. In another jobnet, you could define applications for calculating order data, creating order forms, and conducting stocktaking.

You then register the root jobnets for execution, giving each one a specific execution period. For example, you could register the jobnets so that the first jobnet executes until the end of February, and the other jobnet executes from March onwards. With this configuration, the running jobnet changes with the turn of the month, and different applications are executed in February and March, even though they appear as the same jobnet.

The following figure illustrates the operation of a planning group.

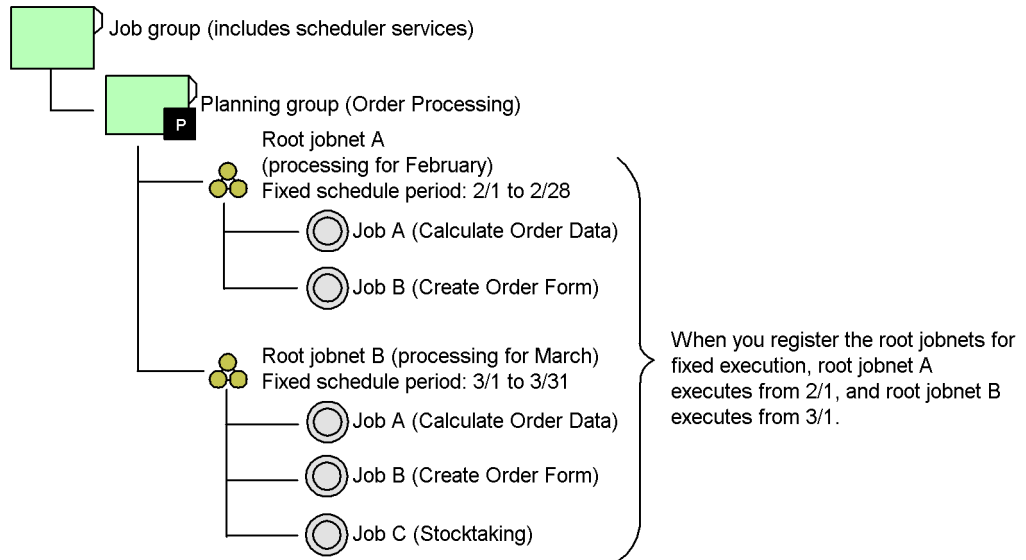
Figure 10-1: Operation of a planning group



If jobnet A is already running (has been registered for fixed execution until December 31), first create jobnet B and register it for fixed execution (from March 1 until December 31). Next, cancel registration of jobnet A, specifying the period from March 1 until December 31.

The following figure shows an example of unit configuration using a planning group.

Figure 10-2: Unit configuration using a planning group



You can only create a planning group directly under a job group (including a scheduler service). You can create multiple planning groups under a job group. You cannot create another planning group under a planning group.

Directly under a planning group, you can create a jobnet or a remote jobnet. The definition content and definition method for a jobnet or remote jobnet defined below a planning group is the same as for a normal jobnet or remote jobnet.

When you register for execution a root jobnet or root remote jobnet that is defined directly under a planning group, the only option available is fixed execution with a specific period. You cannot register these jobnets for planned execution, or for fixed execution with a specific number of future generations.

You can also define calendar information for a planning group, or have the planning group inherit the calendar information from an upper-level job group. You can reference a planning group from a manager job group or a calendar definition.

(a) Unit definition information shared between root jobnets

When you register a root jobnet defined under a planning group for fixed execution, you can link the root jobnet with another root jobnet under the same planning group. When you link root jobnets, the following settings in the unit definition information are shared between the root jobnets in the planning group.

- Concurrent execution setting and **Schedule option**
- Whether to hold the root jobnet when the previous unit ends abnormally or ends

with a warning

Below is a description of those settings in the definition information of a unit that is shared between root jobnets. The example uses a planning group that executes a different jobnet in August and September.

■ **Concurrent execution and the Schedule option**

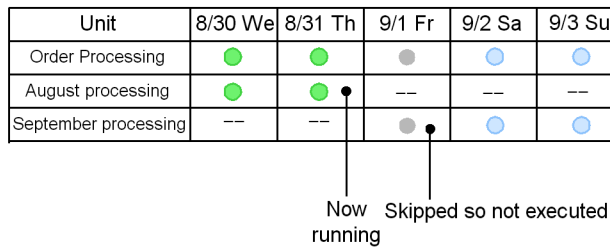
You can define the concurrent execution setting and **Schedule option (Schedule skip, Multi-schedule)** for a root jobnet defined under a planning group. For details about concurrent execution and schedule options, see 3.3.3 *Concurrent execution and schedule option*.

■ **Schedule skip**

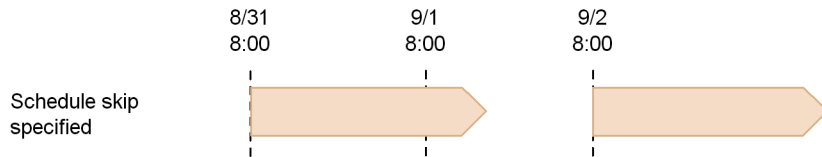
The following is an example of a schedule with the **Schedule option** set to **Schedule skip**, for a jobnet that is executed in September.

Figure 10-3: Schedule option set to Schedule skip

Current date: 9/1



Example of settings for a schedule executed daily at 8:00



- Legend:
- : Ended normally or Now running
 - : Skipped so not executed
 - : Scheduled
 - : Not scheduled

In this case, the generation scheduled for 8/31 has not ended by the start time of the next generation. Therefore, the generation scheduled for 9/1 is not executed, and enters the Skipped so not executed status.

■ **Multi-schedule (Concurrent exec. disabled)**

The following is an example of a schedule with the **Schedule option** set to

Multi-schedule, and with **Concurrent exec.** set to **Disable**, for a jobnet that is executed in September.

Figure 10-4: Schedule option set to Multi-schedule with concurrent execution disabled

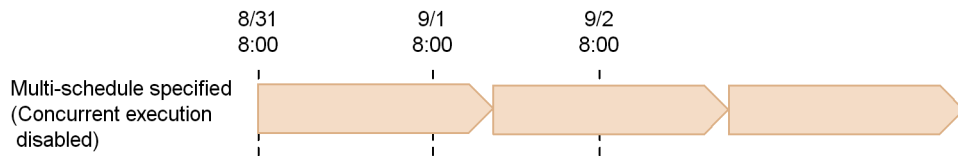
Current date: 9/1

Processing	8/30 We	8/31 Th	9/1 Fr	9/2 Sa	9/3 Su
Order Processing	●	●	●	●	●
August processing	●	●	---	---	---
September processing	---	---	●	●	●

Now running

Because Concurrent execution is disabled, the next generation is executed after waiting for the previous generation to end.

Example of settings for a schedule executed daily at 8:00



Legend:

- : Ended normally or Now running
- : Scheduled
- : Not scheduled

In this case, the generation scheduled for 8/31 has not ended by the start time of the next generation. Therefore, the generation scheduled for 9/1 is executed after the generation scheduled for 8/31 ends.

■ Multi-schedule (Concurrent exec. enabled)

The following is an example of a schedule with the **Schedule option** set to **Multi-schedule**, and with **Concurrent exec.** set to **Enable**, for a jobnet that is executed in September.

Figure 10-5: Schedule option set to Multi-schedule with concurrent execution enabled

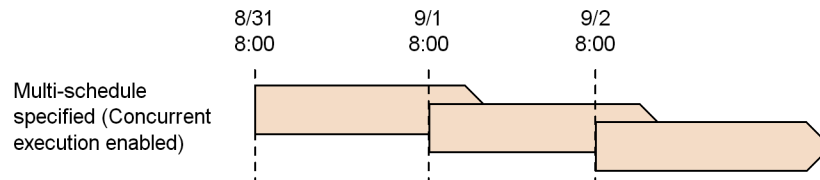
Current date: 9/1

Processing	8/30 We	8/31 Th	9/1 Fr	9/2 Sa	9/3 Su
Order Processing	●	●	●	●	●
August processing	●	●	--	--	--
September processing	--	--	●	●	●

Now running

Because Concurrent execution is enabled, the next generation is executed without waiting for the previous generation to end.

Example of settings for a schedule executed daily at 8:00



Legend:

- : Ended normally or Now running
- : Scheduled
- : Not scheduled

In this case, the generation scheduled for 8/31 has not ended by the start time of the next generation. Therefore, the generation scheduled for 9/1 is executed before the generation executed on 8/31 ends.

In the following circumstances, the succeeding generation is not executed even when concurrent execution is enabled.

- The generation to be executed is being held
- There is no generation to be executed

Previous generation is being held

In the following example, a generation is not executed despite concurrent execution being enabled, because the previous generation for execution is being held.

Figure 10-6: Previous generation is being held

Current date: 9/1

Processing	8/30 We	8/31 Th	9/1 Fr	9/2 Sa	9/3 Su
Order Processing	●	●	●	●	●
August processing	●	●	---	---	---
September processing	---	---	●	●	●

Being held

Because the previous generation is not yet running, this generation is not executed.

Legend:

- : Ended normally
- : Being held
- : Scheduled
- : Not scheduled

In this case, the generation scheduled for 8/31 is still in *Being held* status when the start time of the generation scheduled for 9/1 arrives. Because the previous generation has not started, the generation scheduled for 9/1 is not executed.

No previous generation for execution exists

The following schedule shows an example where a generation is not executed, despite concurrent execution being enabled, because no previous generation for execution exists.

Figure 10-7: No previous generation for execution

Processing for August: Multi-schedule (Concurrent execution disabled)
 Processing for September: Multi-schedule (Concurrent execution enabled)

Current date: 9/1

Processing	8/30 We	8/31 Th	9/1 Fr	9/2 Sa	9/3 Su
Order Processing	●	●	●	●	●
August processing	●	●	---	---	---
September processing	---	---	●	●	●

Now running#

The 9/1 generation is not executed because no previous generation has been executed.

The 8/31 generation is not executed because the previous generation has not ended.

Legend:

- : Now running
- : Scheduled
- : Not scheduled

The root jobnet has an unlimited timeout period

In this case, the generation scheduled for 8/30 has an unlimited timeout period and is still in *Now running* status when 9/1 arrives. The generation scheduled for 8/31 cannot run concurrently with the generation scheduled for 8/30, since concurrent execution is disabled for the latter. As a result, the generation scheduled for 9/1 is not executed, as no previous generation ever runs.

■ **Hold (if previous 'warning' or 'abend')**

With a root jobnet defined under a planning group, you can choose to hold execution of the next generation if the previous generation ends abnormally or with a warning.

The following is an example of the schedule when you set the Hold attribute of a jobnet that is scheduled to execute in September to **Hold if prev. = 'abend'** or **Hold if prev. = 'warning' or 'abend'**.

Figure 10-8: Using the hold attribute

Current date: 9/1

Processing	8/30 We	8/31 Th	9/1 Fr	9/2 Sa	9/3 Su
Order Processing	●	●	●	●	●
July processing	●	●	--	--	--
August processing	--	--	●	●	●

Because the results for the 8/31 generation ended abnormally or with a warning, the 9/1 generation is held.

Ended abnormally, or ended with a warning

Legend:

- : Ended normally
- : Ended abnormally or ended with a warning
- : Being held
- : Scheduled
- : Not scheduled

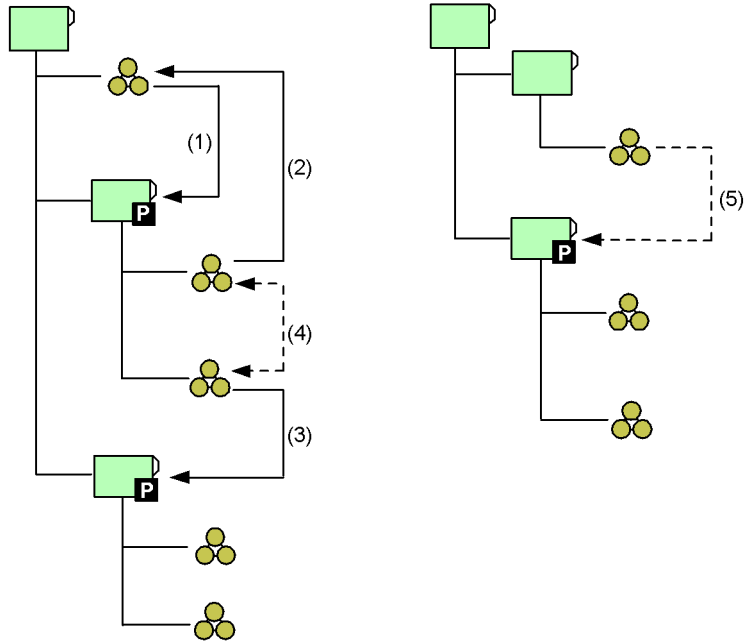
The generation scheduled for 8/31 ended abnormally before the scheduled start time of the generation scheduled for 9/1 was reached. The generation scheduled for 9/1 shifts to the Being held status.

(b) Exclusive schedules for planning groups

You can define a planning group in an exclusive schedule for a jobnet.

You can only use an exclusive schedule for a planning group in certain combinations. The following figure shows the exclusive schedules that are valid for a planning group.

Figure 10-9: Exclusive schedules for a planning group



Legend:

- ▶ (solid arrow) : can be defined as an exclusive schedule
- -▶ (dotted arrow) : cannot be defined as an exclusive schedule

Table 10-1: Valid exclusive schedules for a planning group

Number in figure	Valid?	Type of jobnet for which the exclusive schedule is defined	Unit specified in the exclusive schedule
(1)	Y	Jobnet	Planning group on the same hierarchical level
(2)	Y	Jobnet under a planning group	Jobnet on the same hierarchical level as the planning group
(3)	Y	Jobnet under a planning group	Different planning group on the same hierarchical level as the planning group
(4)	N	Jobnet under a planning group	Jobnet under the same planning group
(5)	N	Jobnet	Planning group on a different hierarchical level

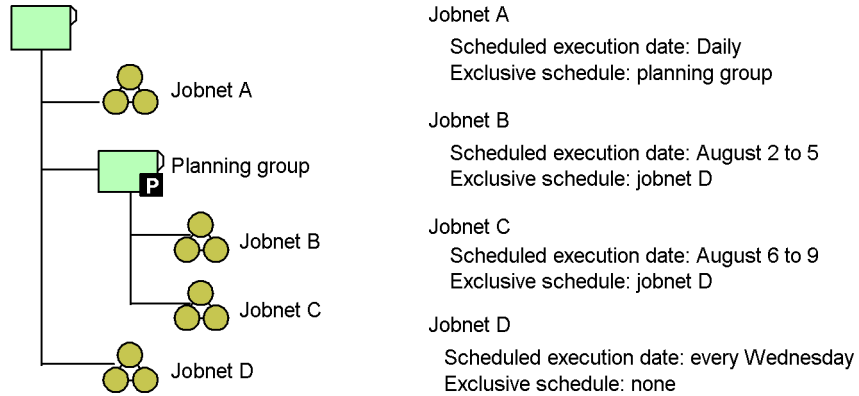
Legend:

Y: Can be specified

N: Cannot be specified

The following figure shows an example of an exclusive schedule including a planning group.

Figure 10-10: Exclusive schedule defined for a planning group



<Exclusive schedules by jobnet>

Jobnet	8/1 Tue	2 Wed	3 Thu	4 Fri	5 Sat	6 Sun	7 Mon	8 Tue	9 Wed	10 Thu	11 Fri
Jobnet A	●	--	--	--	--	--	--	--	--	●	●
Jobnet B	--	--	●	●	●	--	--	--	--	--	--
Jobnet C	--	--	--	--	--	●	●	●	--	--	--
Jobnet D	--	●	--	--	--	--	--	--	●	--	--

Legend:

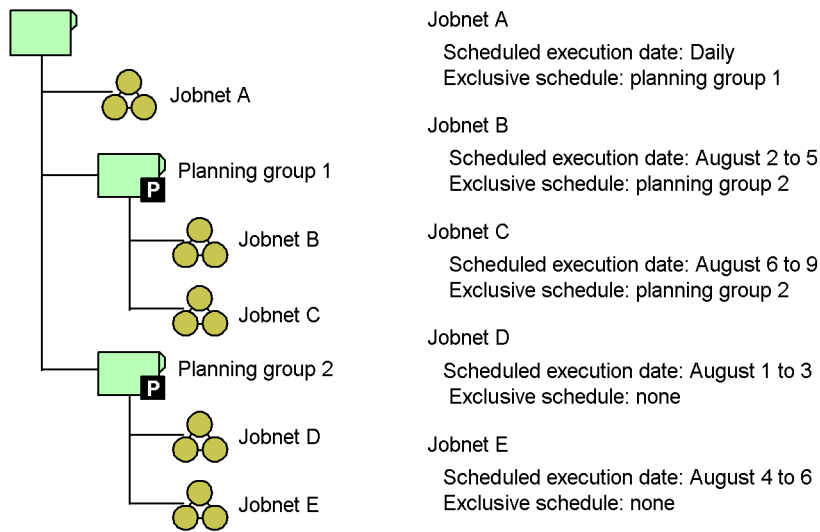
● : Scheduled execution date

-- : Not scheduled

When you specify a planning group as an exclusive schedule, the exclusive schedule is applied to the execution schedules of jobnets under the planning group. In the above example, jobnet A specifies a planning group as an exclusive schedule. This means that the execution schedules of jobnet B and jobnet C, which are under the planning group, are scheduled exclusively. Also, jobnet D is specified as the exclusive schedule for jobnet B and jobnet C, and the execution schedule of jobnet A is therefore scheduled exclusively from the execution schedule of jobnet D.

In the following example, exclusive schedules are specified between planning groups.

Figure 10-11: Defining an exclusive schedule between planning groups



< Exclusive schedules by jobnet >

Jobnet	8/1 Tue	2 Wed	3 Thu	4 Fri	5 Sat	6 Sun	7 Mon	8 Tue	9 Wed	10 Thu	11 Fri
Jobnet A	--	--	--	--	--	--	--	--	--	●	●
Jobnet B	--	--	--	--	--	--	--	--	--	--	--
Jobnet C	--	--	--	--	--	--	●	●	●	--	--
Jobnet D	●	●	●	--	--	--	--	--	--	--	--
Jobnet E	--	--	--	●	●	●	--	--	--	--	--

Legend:
 ● : Scheduled execution date
 -- : Not scheduled

Since planning group 2 is defined as the exclusive schedule for jobnets B and C, which are under planning group 1, the exclusive schedule also applies to jobnet D and jobnet E, since they are under planning group 2. The execution schedule for jobnet A, for which planning group 1 is the exclusive schedule, is therefore exclusive of jobnet D and jobnet E.

(2) Flow of planning group definition and execution

Use the following procedure to define and execute a root jobnet using a planning

group.

- Defining a root jobnet in a new planning group and executing the jobnet

To create and execute a new planning group and root jobnet:

1. Create a planning group.
2. Define a jobnet.
3. Register the jobnet for fixed execution with a specific period.

- Changing a root jobnet that is registered for execution (in operation) to a planning group and executing the jobnet

To change a root jobnet that is registered for execution (in operation) to a planning group and then execute the jobnet:

1. Cancel registration of the root jobnet.
2. Change the name of the root jobnet to the name of the planning group.
3. Create a planning group directly under the job group.
Give the planning group the name that belonged to the root jobnet before you changed it in step 2.
4. Copy the root jobnet whose name you changed in step 2 and paste it into the planning group. Then delete the original root jobnet.
5. Register the root jobnet you copied for fixed execution.

- Adding a new root jobnet definition to an existing planning group and executing the jobnet

To add a new root jobnet to an existing planning group and then execute the jobnet:

1. Define the new root jobnet to be added.
2. Register the jobnet for fixed execution with a specific period.

- Canceling the execution schedule of a registered jobnet and adding the execution schedule of a new root jobnet

To cancel a certain period within an execution schedule that is registered for fixed execution in an existing planning group, and add an execution schedule for a new root jobnet in the released period:

1. Cancel registration of the root jobnet under the planning group by specifying the cancel period. Alternatively, prohibit execution of the root jobnet.

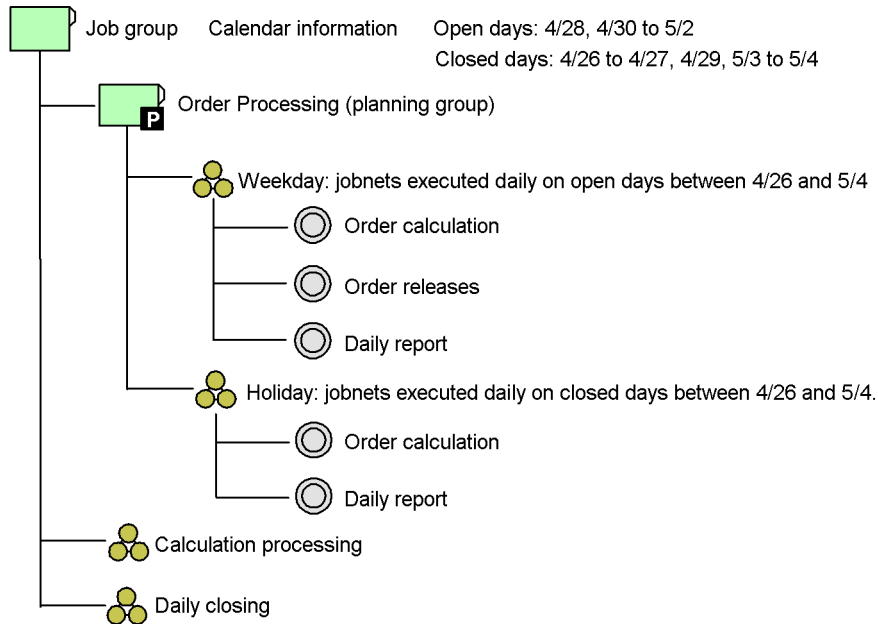
For details on canceling registration, see *4.5.1 Canceling registration of a jobnet*. For details on prohibiting execution, see *4.5.5 Prohibiting execution of a job or jobnet*.

2. Define a root jobnet that executes in the period where you canceled jobnet registration or prohibited jobnet execution.
3. Register the root jobnet you added in step 2 for fixed execution, specifying a period.

(3) Displaying a planning group in JP1/AJS3 - View

This subsection describes the operations for displaying schedule information, unit status (schedule) information, and results (execution results) in the JP1/AJS3 - View window. The display in JP1/AJS3 - View is described below, using a planning group with the following unit configuration as an example.

Figure 10-12: Sample unit configuration for a planning group



(a) Displaying schedule information in the Monthly Schedule window

When you display schedule information for a planning group in the Monthly Schedule window, the schedule area displays merged schedule information for all the units in the planning group.

The schedule information is merged and displayed when you first open the Monthly Schedule window, or when you refresh the window to reflect the latest information. The merged information that is displayed is for the schedule with the latest start time.

Table 10-2 shows the display in the schedule area when you select a planning group and display the Monthly Schedule window. The image is based on the unit

configuration from *Figure 10-12*.

Table 10-2: Display when you select a planning group (at end of execution on April 29)

Unit	4/26	4/27	4/28	4/29	4/30	5/1	5/2	5/3	5/4
	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Order Processing [#]	EN	EN	EN	EN	SE	SE	SE	SE	SE
Weekday	--	--	EN	--	SE	SE	SE	--	--
Order calculation	--	--	EN	--	SE	SE	SE	--	--
Order releases	--	--	EN	--	SE	SE	SE	--	--
Daily report	--	--	EN	--	SE	SE	SE	--	--
Holiday	EN	EN	--	EN	--	--	--	SE	SE
Order calculation	EN	EN	--	EN	--	--	--	SE	SE
Daily report	EN	EN	--	EN	--	--	--	SE	SE
Calculation processing	EN	EN	EN	EN	SE	SE	SE	SE	SE
Daily closing	EN	EN	EN	EN	SE	SE	SE	SE	SE

Legend:

EN : Ended normally

SE : Scheduled for execution

-- : Not scheduled for execution

#

Order processing is a planning group.

Table 10-3: Display when you select a job group above the planning group (at end of execution on April 29)

Unit	4/26	4/27	4/28	4/29	4/30	5/1	5/2	5/3	5/4
	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Order Processing [#]	EN	EN	EN	EN	SE	SE	SE	SE	SE

Unit	4/26	4/27	4/28	4/29	4/30	5/1	5/2	5/3	5/4
	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Calculation processing	EN	EN	EN	EN	SE	SE	SE	SE	SE
Daily closing	EN	EN	EN	EN	SE	SE	SE	SE	SE

Legend:

EN : Ended normally

SE : Scheduled for execution

#

Order processing is a planning group.

Note that there is an option in the Preferences dialog box under the **Schedule** tab that lets you choose whether to display planning groups in the Monthly Schedule window. For details about the Preferences dialog box, see *15.3.36 Preferences dialog box* in the *Job Management Partner 1/Automatic Job Management System 3 Operator's Guide*.

(b) Displaying the status or results of a planning group in JP1/AJS3 - View

When you view the status (execution schedule) and results (execution results) of a planning group in the list area of the JP1/AJS3 - View window, the list area automatically displays the status and results of the units closest to the current day.

The following table shows an example of status and execution results for a planning group.

Table 10-4: Example of status and execution results for a planning group (end of execution on May 5)

Unit	4/30	5/1	5/2	5/3	5/4	5/5	5/6	5/7
	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
Order Processing #	EN	EN	EN	EN	EN	EW	SE	SE
Weekday	EN	EN	EN	--	--	--	SE	SE
Order calculation	EN	EN	EN	--	--	--	SE	SE
Order releases	EN	EN	EN	--	--	--	SE	SE
Daily report	EN	EN	EN	--	--	--	SE	SE
Holiday	--	--	--	EN	EN	EW	--	--
Order calculation	--	--	--	EN	EN	EW	--	--

Unit	4/30	5/1	5/2	5/3	5/4	5/5	5/6	5/7
	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
Daily report	--	--	--	EN	EN	EW	--	--

Legend:

SE: Scheduled for execution (waiting for start time)

EN : Ended normally

EW : Ended with warning

-- : Not scheduled

#

Order processing is a planning group

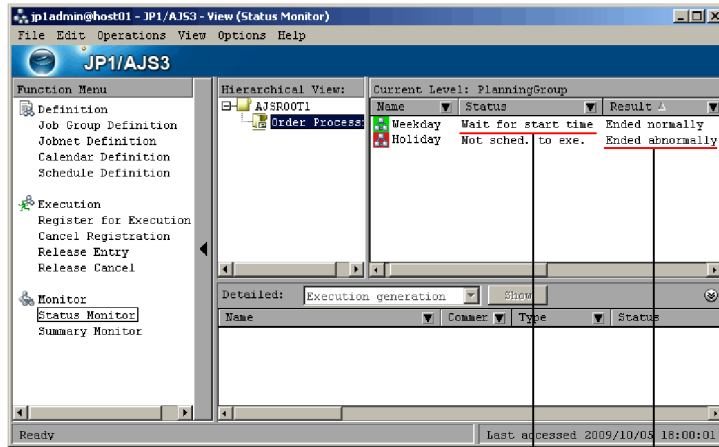
If you compare the results of the jobnet **Holiday** and the jobnet **Weekday**, you can see that the results of the jobnet **Holiday** are closer to the current date and time. Imagine that you now select an upper-level unit to the planning group to display the results of the planning group at the end of execution on May 5 in the JP1/AJS3 - View window. The result of the jobnet **Holiday** is displayed in the **Result** column.

When JP1/AJS3 displays the status of the unit in the JP1/AJS3 - View window, the status of the jobnet **Weekday** is nearer the current time than the jobnet **Holiday**. Therefore, the status of the jobnet **Weekday** is displayed in the **Status** column.

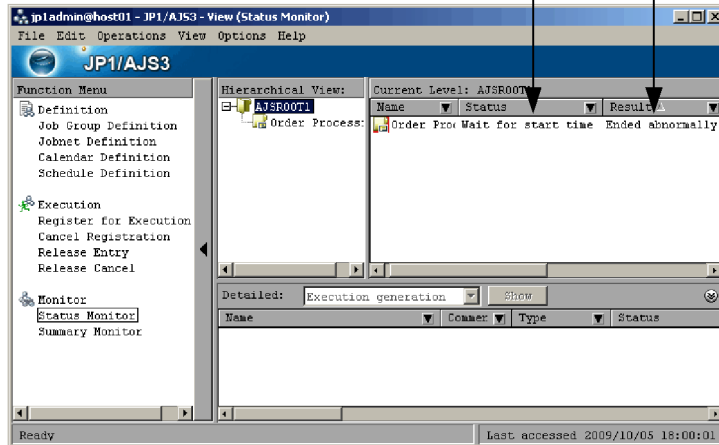
The following figure shows an example of how the status and execution results for a planning group are displayed in the list area of the JP1/AJS3 - View window.

Figure 10-13: Example of display of status and results of a planning group in the JP1/AJS3 - View window

Display example in the tree area of JP1/AJS3 - View when the planning group Order Processing is selected
 Hierarchical View: Current Level: Order Processing



Display example in the tree area of JP1/AJS3 - View when an upper level unit to the planning group is selected
 Hierarchical View: Current Level: AJSROOT1



(4) Examples of configuring planning group definitions

The following are examples of ways in which you can configure the definitions in a planning group to achieve certain results. For example:

- Processing that switches definitions every month
- Processing that switches between two root jobnets

- Processing that switches day and night definitions
- Processing that changes the schedule for an existing planning group and adds a new jobnet

The figures and tables below show an example of root jobnets defined under a planning group, showing the definition of the root jobnets as displayed in the schedule area of the Monthly Schedule window.

(a) Switching definitions each month

The following figure and table show an example in which the planning group contains definitions for a jobnet that executes in February and a jobnet that executes in March.

Figure 10-14: Example of a planning group that switches jobnets every month

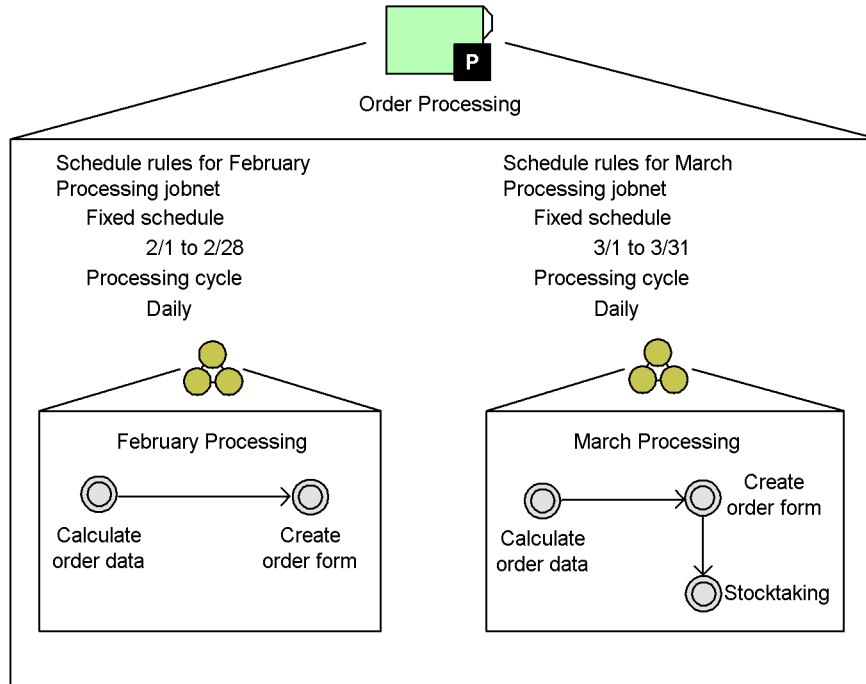


Table 10-5: Image displayed in the schedule area of the Monthly Schedule window for a planning group that switches jobnets every month

Unit	2/22	2/23	2/24	2/25	2/26	2/27	2/28	3/1	3/2	3/3	3/4	3/5
	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue
Order Processing [#]	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE
February Processing	SE	SE	SE	SE	SE	SE	SE	--	--	--	--	--
March Processing	--	--	--	--	--	--	--	SE	SE	SE	SE	SE

Legend:

SE : Scheduled for execution

-- : Not scheduled

#

Order Processing is the planning group.

(b) Switching between two root jobnets

The following figure and table show an example of a planning group containing definitions for a jobnet that executes on weekdays (open days), and a jobnet that executes on holidays (closed days).

Figure 10-15: Example of jobnet definition in a planning group that switches between jobnets

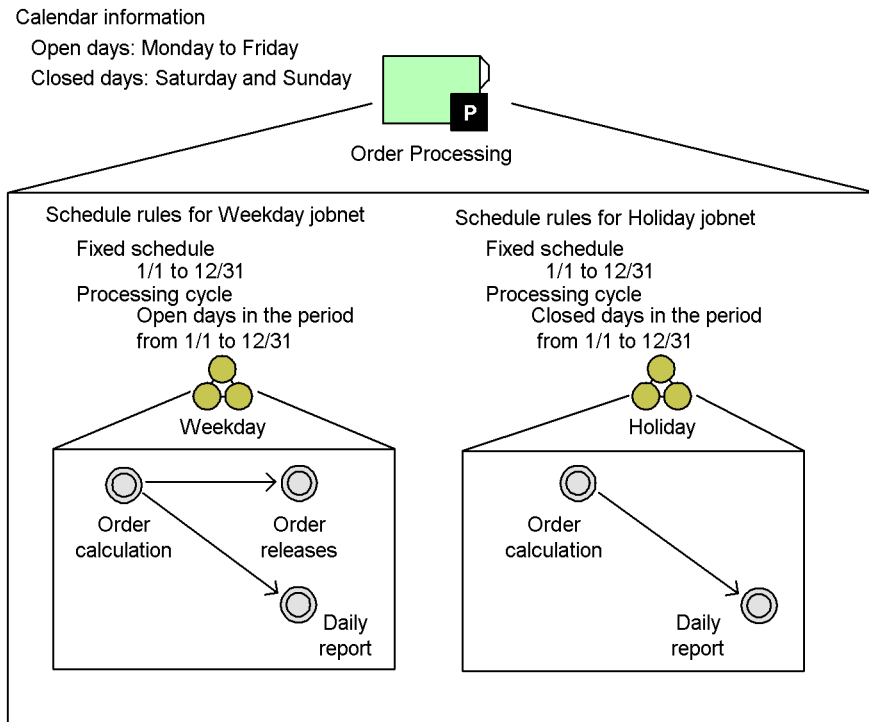


Table 10-6: Image displayed in the schedule area of the Monthly Schedule window for a planning group that alternates between jobnets

Unit	2/22	2/23	2/24	2/25	2/26	2/27	2/28	3/1	3/2	3/3	3/4	3/5
	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue
Order processing#	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE
Weekday	SE	--	--	SE	SE	SE	SE	SE	--	--	SE	SE
Holiday	--	SE	SE	--	--	--	--	--	SE	SE	--	--

Legend:

SE : Scheduled for execution

-- : Not scheduled

#

Order Processing is the planning group.

(c) Switching between jobnets in the same day

The following figure and table show an example of a planning group containing definitions for a jobnet that executes during the day, and a jobnet that executes at night.

Figure 10-16: Example of jobnet definition in a planning group that switches between jobnets within the same day

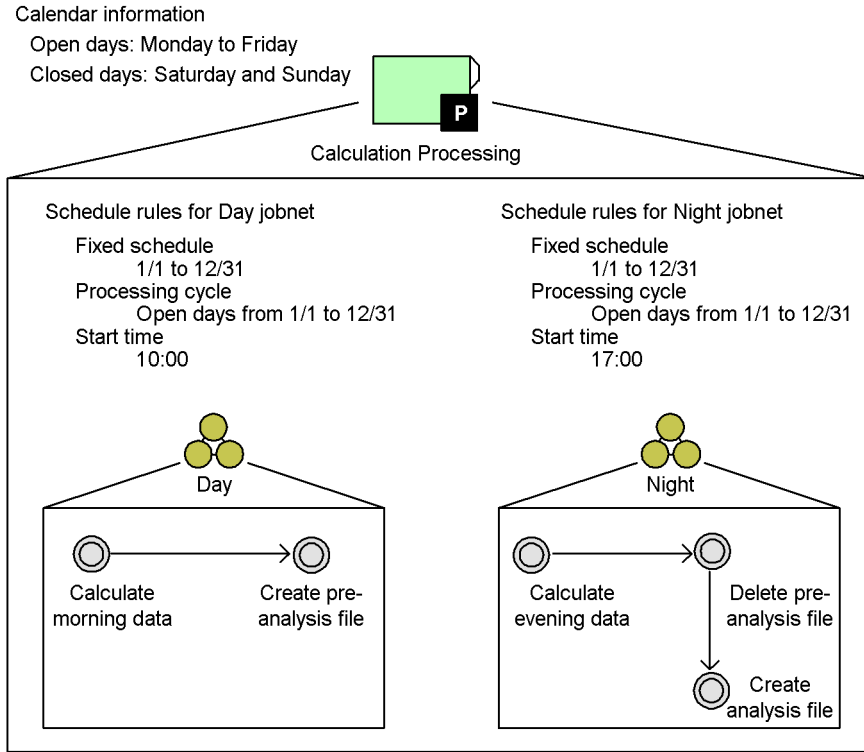


Table 10-7: Image displayed in the schedule area of the Monthly Schedule window for a planning group that switches between jobnets within the same day

Unit	2/22	2/23	2/24	2/25	2/26	2/27	2/28	3/1	3/2	3/3	3/4	3/5
	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue
Calculation processing [#]	SE	--	--	SE	SE	SE	SE	SE	--	--	SE	SE

Unit	2/22	2/23	2/24	2/25	2/26	2/27	2/28	3/1	3/2	3/3	3/4	3/5
	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue
Day	SE	--	--	SE	SE	SE	SE	SE	--	--	SE	SE
Night	SE	--	--	SE	SE	SE	SE	SE	--	--	SE	SE

Legend:

SE : Scheduled for execution

-- : Not scheduled

#

Calculation Processing is the planning group

Supplementary note

The **Day** jobnet and the **Night** jobnet are executed according to the schedule information registering the jobnets for fixed execution.

The behavior of the system when the execution start time for the **Night** jobnet is reached while the **Day** jobnet is still running depends on the **Schedule option** and whether concurrent execution is enabled. For details on concurrent execution and schedule options for planning groups, see (1)(a) *Unit definition information shared between root jobnets*.

- When the Schedule option is set to **Schedule skip**, succeeding generations will have the status **Skipped so not executed**.
- When the **Schedule option** is set to **Multi-schedule**, and concurrent execution is disabled, succeeding root jobnets wait for the previous root jobnet to end before executing.
- When the **Schedule option** is set to **Multi-schedule**, and concurrent execution is enabled, succeeding root jobnets execute without waiting for the previous root jobnet to end.

(d) Modifying an existing planning group schedule, and adding a jobnet

Assume that there is a planning group that alternates between different jobnets for open days (weekdays) and closed days (holidays). Assume that you cancel registration and prohibit execution of the jobnets in the planning group between April 29 and May 5 of the registered execution period, and then add a jobnet that will execute in this released period. The following figure and table show an example of the resulting jobnet definitions.

Figure 10-17: Definition example for adding a new jobnet

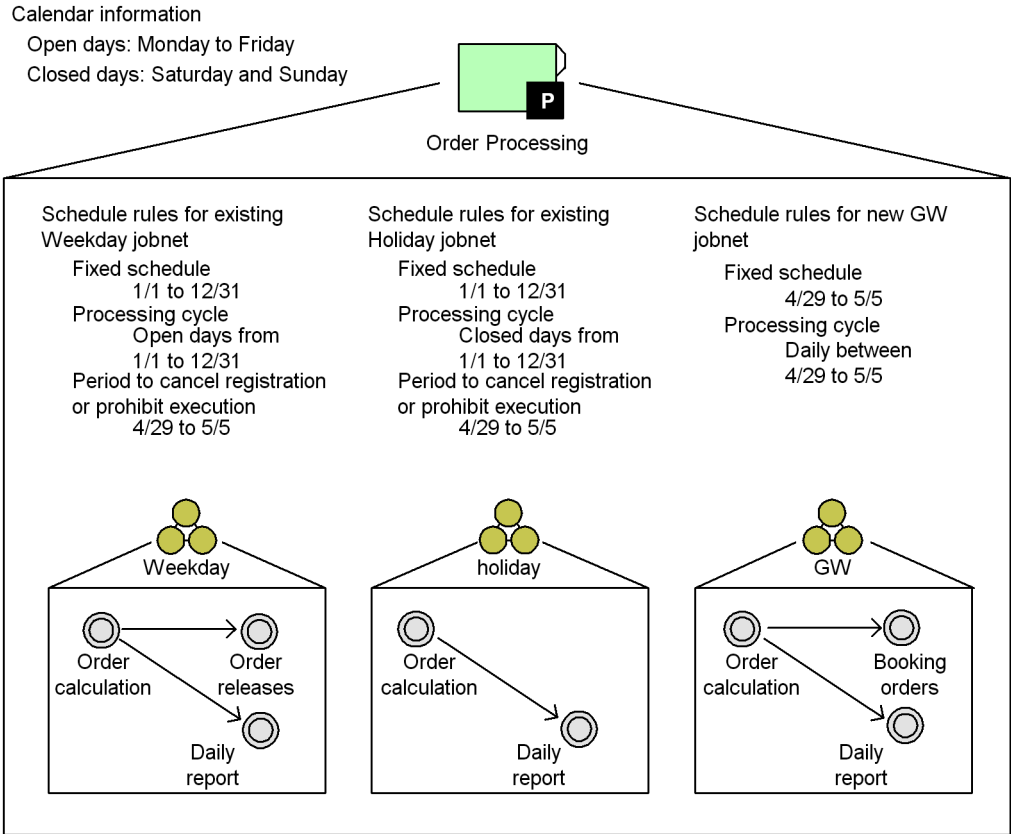


Table 10-8: Image displayed in the schedule area of the Monthly Schedule window when a jobnet is added

Unit	4/26	4/27	4/28	4/29	4/30	5/1	5/2	5/3	5/4	5/5	5/6	5/7
	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
Order Processing ^{#1}	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE
Weekday	--	--	SE	--	--	--	--	--	--	--	SE	SE
Holiday	SE	SE	--	--	--	--	--	--	--	--	--	--
GW ^{#2}	--	--	--	SE	SE	SE	SE	SE	SE	SE	--	--

Legend:

SE : Scheduled for execution

-- : Not scheduled

#1

Order Processing is the planning group.

#2

GW is the added jobnet.

(5) Supplementary notes

This subsection contains additional information about planning groups.

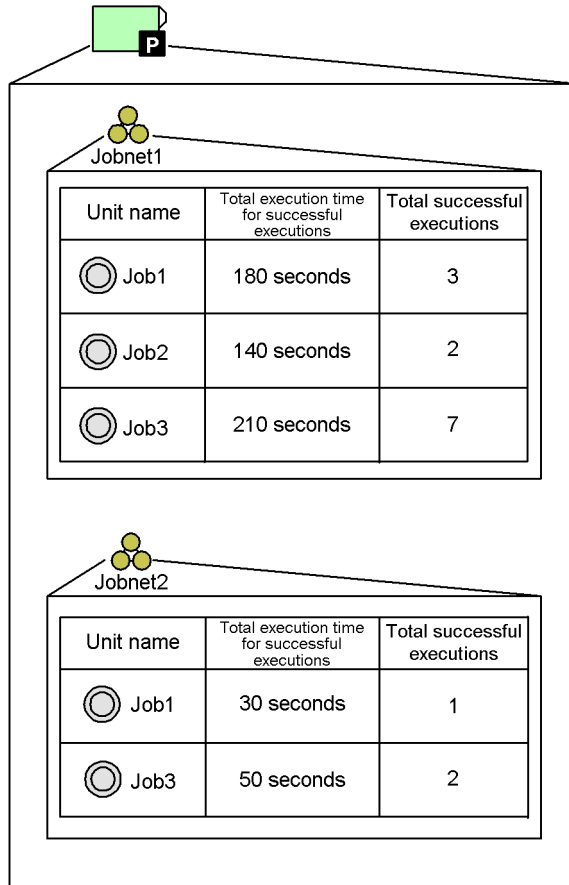
(a) Specifying unit names when using a command to manipulate a planning group

When you specify the name of a unit under a planning group as an option when you execute a command, you can omit the root jobnet name and specify the unit name directly. If you omit the root jobnet name, JP1/AJS3 will automatically select a unit with an existing generation when you execute the command. For details, see *1.1 Command syntax* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 1*.

(b) Statistics and execution simulation for planning groups

JP1/AJS3 calculates the execution time of a unit based on the average past execution times of the unit when it ended normally. The total number of successful executions and their total execution time is kept as statistical information on a root jobnet basis. The following figure shows the statistical information kept for a planning group.

Figure 10-18: Statistical information for a planning group



The system executes an execution simulation based on these statistics. For details on execution simulations, see 4.4.2(2) *Execution simulation*.

Consider Figure 10-18 as an example. The average execution times for Job1, Job2, and Job 3 under Jobnet1 are 60 seconds, 70 seconds and 30 seconds respectively. Assuming that Job1, Job2, and Job3 are to execute in that specific order, if the scheduled start time for Job1 is set to 12:00:00, the scheduled start times displayed for Job2 and Job3 are 12:01:00 and 12:02:10, respectively. The average execution times for Job1 and Job3 in Jobnet2 are 30 seconds and 25 seconds respectively. Even when you use planning groups, the system performs execution simulations based on individual root jobnets.

(c) Generation order in a planning group

The generation numbers in a planning group are generally assigned in order of the

scheduled start times, as with root jobnets in a job group. For details, see *4.2 Managing jobnet generations*.

However, when more than one root jobnet in a planning group has the same scheduled start time, JP1/AJS3 assigns generation numbers differently from normal root jobnets, in order of the ASCII codes of the root jobnet names. The following figure shows an example of generation numbers assigned in the order of the ASCII codes of the root jobnet names.

Figure 10-19: Generation order of units in a planning group

Current time 10:00

Unit name	Date and time						
	7/29 Tue	7/30 Wed	7/31 Thu	8/1 Fri	8/2Sat	8/3Sun	
Planning	12:00	12:00	12:00	12:00	12:00	12:00	12:00
A_Jobnet	12:00	12:00	12:00	--	--	--	--
B_Jobnet	--	--	--	12:00	12:00	12:00	12:00
Generation	Generation 2	Generation 1	Generation 0	Generation -1	Generation -2	Generation -3	Generation -4
Execution ID	@A100	@A101	@A102	@A103	@A104	@A105	@A106

If A_Jobnet and B_Jobnet in the planning group both start execution at 12:00 on 7/31, the generation numbers will be 0 for A_Jobnet and -1 for B_Jobnet.

(d) Order of execution registration in a planning group

When you have jobnets that are executed more than once a day defined under a planning group, register the jobnets for execution in order of their scheduled start times, starting from the earliest.

If you register a jobnet with a later scheduled start time first, the assigned generation numbers will be the opposite of the execution order. This may make monitoring the different jobnets more difficult.

The following figure shows how generation numbers are assigned when you register jobnets for execution in order from the jobnet with the latest start time.

Figure 10-20: Generation order when jobnets are registered in order from the latest start time

Schedule rules for jobnet A
 Processing cycle: daily
 Start time: 10:00
 Schedule rules for jobnet B
 Processing cycle: daily
 Start time: 16:00

Current time 20:00

Unit name	Date and time					
	7/30 Wed		7/31 Thu		8/1 Fri	
Planning	--	16:00	--	16:00	--	16:00
Jobnet A	--	--	--	--	--	--
Jobnet B	--	16:00	--	16:00	--	16:00
Generation	--	Generation 1	--	Generation 0	--	Generation -1
Execution ID	--	@A101	--	@A102	--	@A103

The jobnets are registered at 20:00 on 7/30, starting with jobnet B which has the later start time. Jobnet B is executed immediately because the current time is later than the start time for Jobnet B. The generation number for the 7/30 generation of jobnet B therefore becomes 1.

Current time 22:00

Unit name	Date and time					
	7/30 Wed		7/31 Thu		8/1 Fri	
Planning	10:00	16:00	10:00	16:00	10:00	16:00
Jobnet A	10:00	--	10:00	--	10:00	--
Jobnet B	--	16:00	--	16:00	--	16:00
Generation	Generation 2	Generation 1	Generation 0	Generation -1	Generation -2	Generation -3
Execution ID	@A104	@A101	@A105	@A102	@A106	@A103

Jobnet A has the earlier start date. When jobnet A is registered for execution from 7/30, it is executed immediately because the current time is later than the start time. The generation numbers are allocated in order of the scheduled start times, so the generation number for the 7/30 generation of jobnet A is 2.

With the example shown in *Figure 10-20*, if you specify generation 1 using the `ajsshows` command with the `-1` or `-g` option, the command will be performed on the execution generation of Jobnet B, not Jobnet A. Because the generation number of the registered jobnet is not 1, you must specify generation 2 as well as generation 1 when you monitor the jobnet status.

When you register jobnets for execution in order from the jobnet with the earliest scheduled start time, you ensure that the generation number of the registered jobnet

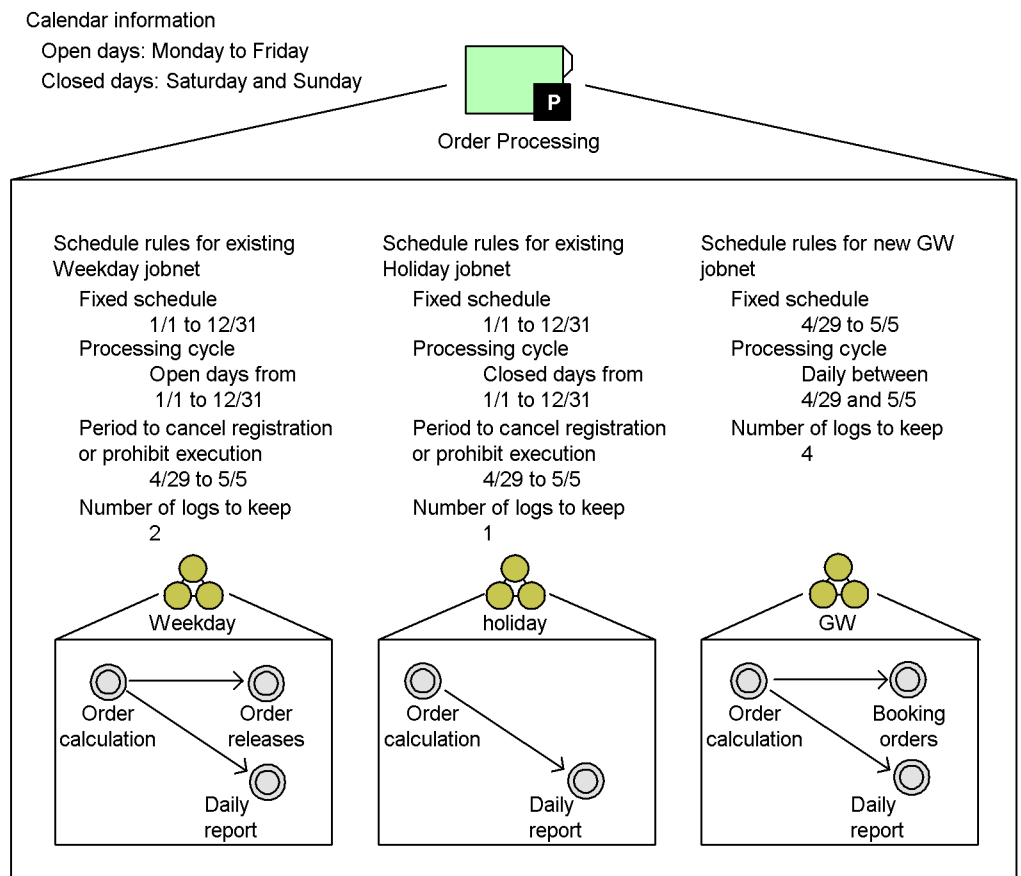
remains 1 because when the first jobnet ends, the jobnet with the next earliest start time executes. Therefore, you need only specify generation 1 when you monitor the jobnet status.

(e) Number of logs to keep in a planning group

You can set the number of logs to keep for each unit in a root jobnet that is switched in at a specific time by a planning group. For details on the number of logs to keep, see *4.2.3 Jobnet generation management*.

For example, if you set the number of logs to keep to 2, JP1/AJS3 only keeps and displays the results for two generations. Execution results for all earlier generations are deleted. The following figure shows examples of root jobnet definitions in which the number of logs to keep has been defined.

Figure 10-21: Example of root jobnet definitions with a defined number of logs to keep



Based on the definitions shown in the figure above, the following is displayed in the schedule area of the Monthly Schedule window at completion of execution on May 8.

Table 10-9: Image displayed in the schedule area of the Monthly Schedule window when the number of logs to keep for each root jobnet is defined

Unit	4/27	4/28	4/29	4/30	5/1	5/2	5/3	5/4	5/5	5/6	5/7	5/8
	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
Order Processing [#]	EN	--	--	--	--	EN	EN	EN	EN	--	EN	EN
Weekday	--	--	--	--	--	--	--	--	--	--	EN	EN
Holiday	EN	--	--	--	--	--	--	--	--	--	--	--
GW	--	--	--	--	--	EN	EN	EN	EN	--	--	--

Legend:

EN : Ended normally

-- : Not scheduled

#

Order Processing is the planning group.

In this example, the jobnet **Weekday** also executes on May 6. However, since the number of logs to keep for **Weekday** is set to 2, the execution results for **Weekday** on May 6 are deleted.

10.2 Remote jobnet

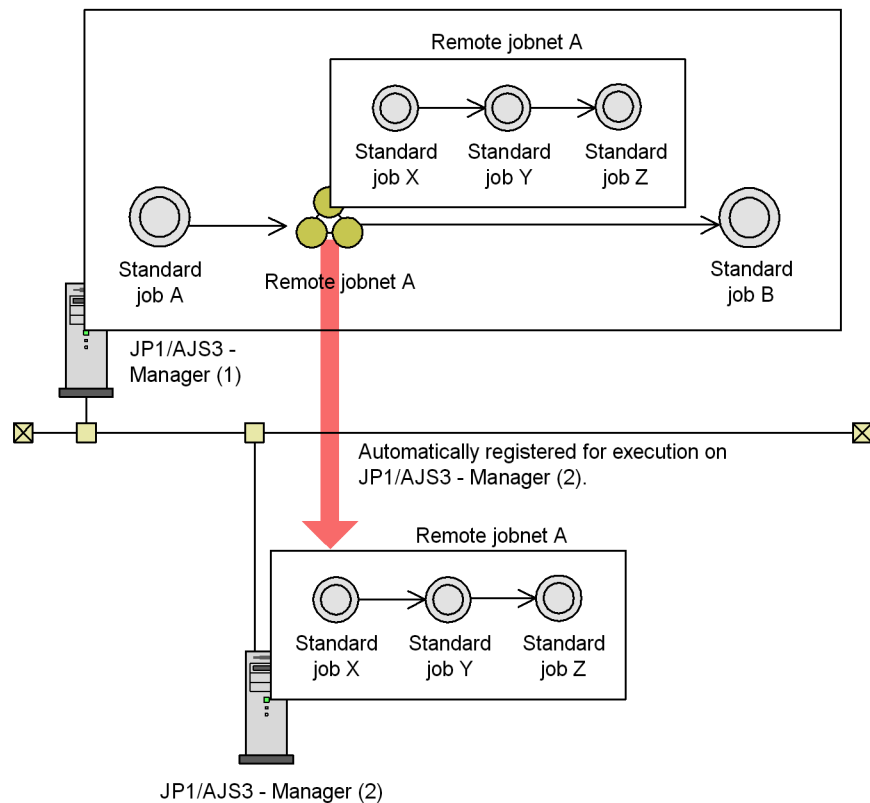
A *remote jobnet* is a jobnet defined on the local manager host, but transferred to a different manager host for execution. The use of remote jobnets allows the workload of JP1/AJS3 - Manager, when executing jobnets or jobs defined under the remote jobnet, to be distributed over multiple hosts.

Remote jobnets are registered for immediate execution on the manager host at the transfer destination and all schedule information set for the lower units is ignored.

No changes or additions can be made to the remote jobnet definitions on the transfer source host. Also, execution results will be deleted automatically if the remote jobnet exceeds the number of logs to keep setting on the transfer source host.

An example of defining a remote jobnet is as follows.

Figure 10-22: Example of defining a remote jobnet



When the jobnet in this example is executed, the processes are performed in the

following sequence:

1. Standard job A is executed.
2. When standard job A ends abnormally, remote jobnet A is transferred to the JP1/AJS3 - Manager service of JP1/AJS3 - Manager (2).

The JP1/AJS3 - Manager service of the transfer destination registers remote jobnet A for immediate execution as the root jobnet.

3. Standard jobs X, Y, and Z in the remote jobnet are executed sequentially.
4. When standard job Z ends and remote jobnet A ends normally, standard job B is executed.

When setting up a jobnet without using a remote jobnet, create each job in the jobnet such that the job is executed on the execution destination agent. In this case, in the Define Details dialog box of the jobnet, add the host name of the execution destination agent to **Target host**. Jobs in the jobnet for which **Target host** is omitted will inherit that of the above jobnet.

If the existence of a file is determined in a judgment job, execute as a previous job on the agent a script or batch file with a return code from which the existence of the file can be determined, and use this for the end code of the judgment job.

The following provides notes you must keep in mind when using remote jobnets.

■ Notes on remote jobnets

- Remote jobnets are immediately registered for execution at the transfer destination host, thereby invalidating all scheduling information set for the jobnets included in the remote jobnet.
- In the transfer destination host of a remote jobnet, set the scheduler service for executing the remote jobnet. Although the remote jobnet service can be used with the default scheduler service, we recommend that you use a scheduler service for executing remote jobnets that is not one of the services used for regular tasks. For details about adding a scheduler service, see *6.1.1 Settings for starting multiple scheduler services* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or see *14.1.1 Settings for starting multiple scheduler services* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.
- For the transfer destination host, specify the host name of the transfer source host and the JP1 user who registered the jobnet for execution in the user mapping definition. Similarly, for the transfer source host, specify the host name of the transfer destination host and the JP1 user who registered the jobnet for execution in the user mapping definition.
- Remote jobnets are registered for immediate execution on the transfer destination host. If the registration process spans more than one day, the jobnet might be

placed in the *Skipped so not exe.* status on the transfer destination host. If operation of the remote jobnet might take more than one day, specify `twoday` (two days) or `unlimit` (unlimited) in the `EXECDEFER` environment setting parameter for the scheduler service that executes the remote jobnet on the transfer destination host.

For details on environment setting parameters, see 2.2 *Setting up the scheduler service environment* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 2*.

- When you execute a remote jobnet, a job group with a unique name is created on the transfer destination host, and a jobnet with the same name as the remote jobnet is created and executed under this job group. We recommend that you give a remote jobnet a name that is unique within the system, so that you can easily identify which jobnet on the transfer destination host corresponds to each remote jobnet on the transfer source host.
- When you set the transfer source host as the transfer destination host of a remote jobnet, the transfer source host performs the processing that is usually performed by the transfer destination host, resulting in degraded performance compared to the execution of normal jobnets.
- The jobnet definitions are transferred at execution of the remote jobnet, increasing the startup overhead. When the remote jobnet contains only a few jobs, the network actually has a greater load than when the executing host is set in the job definition.
- Any error in a job in the remote jobnet is reported to the transfer source host only at completion of the whole remote jobnet. Also, the transfer source host cannot monitor the status of the individual jobs in the remote jobnet.
- Because a remote jobnet is handled as a single job, any failed job in the jobnet will not resume from a point part way through the remote jobnet when re-executed.
- The status of a remote jobnet is only managed from when the jobnet is started to when it is terminated. As a result, after the remote jobnet is terminated, any re-execution or status change of the units within the executed remote jobnet performed on the transfer destination host of the jobnet will not be applied on the transfer source host.

Once normal termination of the remote jobnet takes place, due to operations such as re-execution, on the transfer destination host, to continue execution, re-execute the subsequent unit of the remote jobnet, on the transfer source host.

- When a unit in the remote jobnet uses functionality that is not supported by JP1/AJS3 - Manager or JP1/AJS2 - Manager on the transfer destination host, the jobnet is placed in *Ended abnormally* status and the following message appears in the execution result details: `KAVS0650-E Invalid specification in the Unit definition parameter file (file-name) . (sentence:`

sentence-number). In this case, upgrade JP1/AJS2 - Manager on the remote host to a version that supports the unit's functionality, and then rerun the remote jobnet.

- If you cancel registration of a jobnet created by a remote jobnet on a transfer destination host or if you cold-start the scheduler service that defines the jobnet, the association with the remote jobnet on the transfer source host will be lost. In such cases, manually delete the job groups created by the remote jobnet because they are not deleted automatically. Similarly, if you cold-start the scheduler service that defines the remote jobnet on the transfer source host, the jobnet registered for execution on the transfer destination host is not unregistered. Cancel the registration manually.
- If registration of a remote jobnet is canceled on the transfer source host or if the number of logs to keep is exceeded while JP1/AJS3 - View is accessing a job group created by the remote jobnet, the job group that JP1/AJS3 - View is accessing is not deleted automatically. In such cases, delete the job group manually.
- When a remote jobnet is executed or registration for execution of a remote jobnet is canceled, special processes are started on the host where the jobnet was defined and on the host to which the jobnet was transferred. You may want to execute multiple remote jobnets concurrently at the definition source host, cancel registration of remote jobnets that have many logs to keep, or execute multiple remote jobnets on the same transfer destination host. However, if you do this, a significant amount of system resources will be consumed and the load on the system will increase. Such effects may raise the possibility of performance degradation or process generation failure. To avoid such problems, we recommend that you do not execute more than one remote jobnet concurrently.
- When a remote jobnet is executed or registration of its execution is canceled, the commands for operating the unit are internally executed on the host where the jobnet is defined and on the jobnet destination host. During cancellation of registration, the number of commands executed concurrently on the jobnet destination host is the same as the number of logs to keep. You need to take this into account when making estimates for JP1/AJS3.
- When a remote jobnet is registered for execution or its registration for execution is canceled, special processes are started on the transfer source host and the transfer destination host. At some point, you might want to concurrently execute remote jobnets on the transfer source host, cancel registration of remote jobnets that have many logs to keep, or execute multiple remote jobnets on the same transfer destination host. Doing this, however, consumes a significant amount of resources and increases the system load, possibly resulting in degraded performance or the generation of failed processes. To avoid these problems, we recommend that you do not execute more than one remote jobnet concurrently.
- When a remote jobnet is registered for execution or its registration for execution

is canceled, commands for operating the unit are internally executed on the transfer source host and the transfer destination host. When registration is canceled, the number of commands executed concurrently on the transfer destination host is the same as the number of logs to keep. You need to take this into account when estimating sizes for JP1/AJS3.

10.3 Manager jobnet

A *manager jobnet* is for referencing a jobnet defined on another manager host.

In a manager jobnet, you define the name of the jobnet to be referenced and the manager host on which it resides. Only a root jobnet can be defined. This allows you to reference a jobnet (defined as a manager jobnet) on a remote manager host, from JP1/AJS3 - View connected to the local manager host.

10.4 QUEUE jobs and submit jobs

This section describes QUEUE jobs and submit jobs.

(1) What are QUEUE jobs?

QUEUE jobs are executed via a specified queue. Each QUEUE job is registered in a queue, and is then executed by an agent connected to the queue. In a system that uses the standard database, QUEUE jobs are used when JP1/AJS3 is linked with another system (JP1/NQSEXEC or JP1/OJE, for example). QUEUE jobs can be used in JP1/AJS3 to implement the following processing:

- Create separate queues in which to register jobs according to their purpose.
- Connect multiple agent hosts to a queue and distribute jobs based on the priorities assigned to the agent hosts.

The same processing can be realized in a standard database configuration with PC jobs, Unix jobs, action jobs, and custom jobs, using an execution agent group rather than queues. We recommend that you use an execution agent group when defining new PC jobs, Unix jobs, action jobs, or custom jobs.

(2) What are submit jobs?

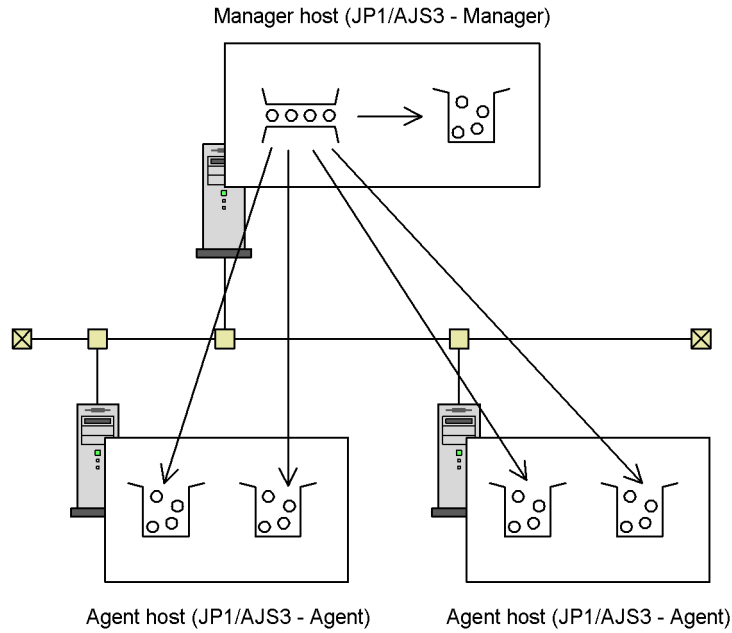
In JP1/AJS3, jobs can be defined and executed in a jobnet, or they can be executed singly by command, via direct registration in a queue in JP1/AJS3 - Manager. The jobs in the queue are sent to an agent connected to the queue for execution. If you specify the agent when directly registering a job for execution, it will be placed in the default queue. The act of directly registering a job in a queue is called *submitting* a job, and the job itself is referred to as a *submit job*. To submit a job, you use the `jpqjjobsub` or other command.

Jobs in a jobnet are executed by a scheduler service, which manages the job execution times and execution sequence, and runs each job according to its schedule. Submit jobs, on the other hand, are submitted and executed by a user executing a command. The execution start time can be specified as a command option.

(3) Queue and agent configurations

The following describes how jobs are executed using a queue and agent configuration. The following figure illustrates the queue and agent concept.

Figure 10-23: Using a queue and agents



Legend:

: Queue

: Agent

(a) What is a queue?

A queue is an area in which registered jobs can be placed temporarily so that the number of QUEUE jobs and submit jobs to be executed concurrently in JP1/AJS3 will not become overly large. One or more agents can be connected to the queue. When the user executes a submit job with an agent host name specified in the `-ah` option of the `jpqjobs` command, the job is registered in the internal queue whose name matches the specified agent host name. This matching queue is the *default queue*. Multiple agents can also be connected to a default queue. Jobs are queued in order of registration and are transferred sequentially to the connected agent hosts for execution.

You can define the maximum number of jobs that can be registered in a queue. You can also execute multiple jobs concurrently. However, if the number of jobs being executed exceeds the system's processing capacity, performance could decline and errors due to resource shortages could occur. For example, if you attempt to execute

more jobs than can be handled comfortably with the available system memory, memory swapping increases and execution performance declines significantly. Continuing to execute more jobs in this situation will cause problems such as jobs terminating abnormally due to insufficient memory.

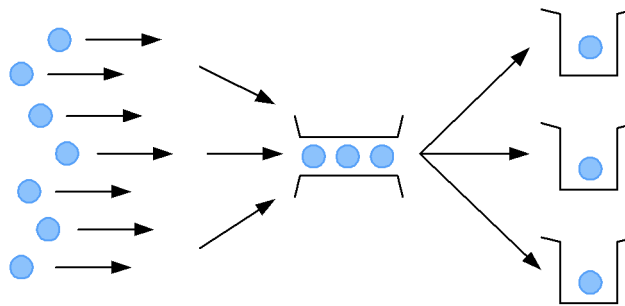
By using a queue, these sorts of problems can be avoided and jobs can be executed more efficiently. When the number of jobs to be executed concurrently reaches a set limit, subsequent jobs are made to wait and are passed in turn to the agent as preceding jobs finish.

Cautionary note

Submit jobs are not necessarily transferred and executed in order of registration in the queue. Submit jobs might switch places in the queue if, for example, the next submit job cannot be transferred because the execution-locked resource specified when it was registered is being used by another submit job. For details about the execution sequence of submit jobs, see *7.1.1(4)(b) Notes on the execution order of submit jobs* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

The following figure shows how a queue works.

Figure 10-24: Processing of a queue



(b) What is an agent?

An *agent* is a host that executes jobs. The term can denote a component of the system configuration, and it can denote the functionality and control inherent in an agent. QUEUE jobs and submit jobs are transferred sequentially from the queue to an agent connected to the queue, and are executed by the agent.

You can define how many jobs an agent can execute concurrently. Like the maximum number of jobs that can be registered in a queue, you must set the maximum number of concurrently executable jobs in an agent in line with the system processing capacity.

When multiple agents are connected to the queue, you can assign different priorities to them. This means that if an agent already has a large number of *Now running* jobs compared with the number of jobs it can execute concurrently, the next job will be sent

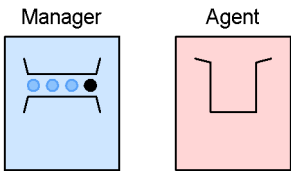
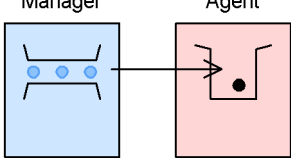
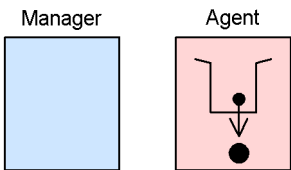
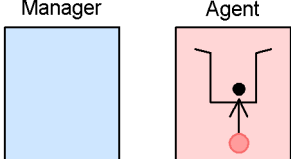
to the agent with the next highest priority. For details, see (6) *Distributing jobs among execution hosts (agent hosts)*.

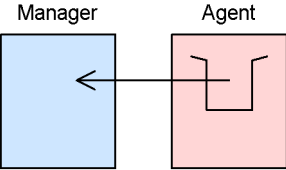
You can also distribute jobs within one machine. To do so, you define multiple aliases of the same machine (single IP address) as agents, and register different kinds of jobs in each agent. For details, see 7.1.1(3) *Running multiple execution hosts on the same computer* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

(c) Flow of execution of QUEUE jobs and submit jobs

The following table describes the processing flow, and the manager/agent relationship, when QUEUE jobs and submit jobs are executed in JP1/AJS3. The descriptions here assume that the job ends normally. In fact, a job could end abnormally or some other situation could apply.

Table 10-10: Processing flow when executing QUEUE jobs and submit jobs

	Job status	Description
Now queuing		The manager registers the job as requested, and places it in the queue. The job status becomes <i>Now queuing</i> .
Now running		The manager takes the job from the queue and transfers it to an agent, requesting the agent to execute the job.
		The agent OS starts running the executable file or script defined in the job. The job status becomes <i>Now running</i> , and the agent waits for the job to end.
		When the job ends, the agent records its return code and execution result.

Job status		Description
Ended normally		The agent sends the job's return code and execution result to the manager. On receipt, the job status changes to <i>Ended</i> .

(4) Execution environment for QUEUE jobs and submit jobs

To use QUEUE jobs or submit jobs in JP1/AJS3, you must define a queue using the `jpgimport` command. You can change the queue and agent configuration while JP1/AJS3 services are active, using the `jpgqueueadd` or other command. For details about setting up the execution environment, see 7.1.2 *Setup for using QUEUE jobs and submit jobs* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

For details about the `jpgimport` and `jpgqueueadd` commands, see 1.5.9(2) *Commands available in an execution environment for queue or submit jobs* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2*.

To execute a QUEUE job that links with another system such as JP1/NQSEXEC or JP1/OJE, you must first create a queue in the other system. For details about linkage with another system, see the *Job Management Partner 1/Automatic Job Management System 3 Linkage Guide*.

For details about changing the job execution environment to suit your system operation, see 7.1.2(1) *Setting the execution environment for QUEUE jobs and submit jobs* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

(5) Restricting the number of concurrent jobs

You can restrict the number of QUEUE jobs or submit jobs that can be executed concurrently in JP1/AJS3 according to resource availability at the execution host.

(a) Maximum number of concurrently executable jobs

When jobs are managed in a queue, you can restrict the number of jobs executed at the same time by setting a maximum number of concurrent jobs in the definition information in the agent host. The maximum is specified by period, in units of 30 minutes. You can set different limits for different time periods to distribute the processing load across the system.

The limits apply only to QUEUE jobs and submit jobs executed in JP1/AJS3. Set the limits on an agent basis. You cannot set a maximum number of concurrently executable jobs per queue. When the limit is reached, subsequent jobs remain in *Now*

queuing status (waiting to execute) until the active jobs finish.

(b) Methods of setting the maximum number of concurrently executable jobs

There are two ways of setting the maximum number of concurrently executable jobs: You can set a parameter in the agent host definitions in the file `jpgqsetup.conf` (job execution environment configuration definition file for QUEUE jobs and submit jobs) and then execute the `jpgqimport` command, or you can set the relevant command option when adding an agent host using the `jpgqagtadd` command.

If you use the `jpgqimport` command method, the default setting for the maximum number of concurrently executable jobs in the configuration definition file is as follows: limit start time 0000, limit end time 0000, and maximum jobs 5 (maximum of five concurrently executable jobs, 24 hours a day). You can change this setting later as needed, using the `jpgqagtalt` command. If you add an agent host using the `jpgqagtadd` command method, but do not specify the maximum number of concurrently executable jobs (`-cp` option), the default setting of 00:00-00:00=0 (zero concurrent jobs, 24 hours a day) is assumed.

A maximum of zero concurrently executable jobs means that no jobs can be executed. To enable job execution by the new agent host, specify 1 or higher, or change the setting afterwards using the `jpgqagtalt` command. For an example of specifying a maximum number of concurrently executable jobs using this command, see also *jpgqagtalt* in 3. *Commands Used for Special Operation* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2*.

(c) Load distribution based on the maximum number of concurrently executable jobs

When multiple agent hosts are connected to the queue, you can distribute the processing load among the agents, based on the *agent usage rates* (ratio of active jobs to the maximum number of concurrent jobs). For details, see (6) *Distributing jobs among execution hosts (agent hosts)*.

(d) Cautions on setting concurrent execution limits

For cautionary notes about concurrent execution limits, see *Notes on setting a maximum number of concurrently executable jobs* in 7.1.1(4)(a) *Maximum number of concurrently executable jobs (QUEUE jobs and submit jobs)* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

(6) Distributing jobs among execution hosts (agent hosts)

(a) Distributing jobs among multiple execution hosts (agent hosts) connected to the queue

When you register a QUEUE job or submit job for execution in JP1/AJS3, the job is placed in a queue (specified or default) managed by the job execution control. You can

connect multiple agent hosts to the queue, specifying a different priority for each. Jobs are distributed to the agent hosts based on their specified priorities.

Specify the agent host priorities to suit your system environment. If you assign different priorities, each job will be sent for execution to the agent host that has the highest priority. When that agent's usage rate[#] reaches 100%, the next job is sent to the agent host that has the next highest priority. For example, if the agent host with the highest priority is already fully occupied or is in failed or stopped state and cannot run the job, the agent host with the next highest priority can run it instead. Setting different priorities is useful for this type of system operation.

If you set the same priority for all agent hosts, each job will be sent to the agent host that has the lowest agent usage rate. If no particular agent host takes precedence, jobs are distributed evenly among the agent hosts. Setting identical priorities is useful for this type of load-balancing system.

#

Agent usage rate is the ratio of active jobs to the maximum number of concurrent jobs at an agent host. It is given by the following equation, rounded to four decimal points:

$$\text{Agent usage rate} = \text{number of active jobs} / \text{maximum number of concurrent jobs}$$

Cautionary note

When two or more agent hosts have the same priority and usage rate, the agent host with the highest agent ID[#] is selected by default to execute the job. For example, if the jobs are short and quickly completed, the agent host's usage rate hardly increases and differs only slightly from that of other agent hosts.

Consequently, because jobs are always assigned to the agent host having the highest agent ID, that agent executes more jobs than the others, resulting in asymmetric system usage. In such situations, you can distribute jobs more uniformly if, instead of the agent ID method, you set a job distribution method that prioritizes the agent host that has not been used for the longest time.

For details on defining priorities, see *7.1.7 Method of determining the agent host to which a job is distributed* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Configuration) Guide*.

#

Agent IDs are set in `jpgqsetup.conf` (job execution environment configuration definition file for QUEUE jobs and submit jobs) when you execute the `jpgqimport` command. Each time you add an agent host using the `jpgqagtadd` command, an ID is assigned in ascending order of the unused IDs.

When agent hosts are displayed by the `jpgqagtshow` command, they are

listed in ascending order of their agent IDs.

(b) Method of distributing jobs among multiple execution hosts (agent hosts) connected to the queue

In the following two cases, there may be a delay of up to 15 minutes in distributing QUEUE jobs or submit jobs in JP1/AJS3 to execution hosts (agent hosts) that are operating normally. During this period, the jobs remain in *Now queuing* status:

- When a failure occurs simultaneously on three or more execution hosts
- When three or more execution hosts have stopped

In such cases, restart the affected execution hosts.

(7) Exclusive control when executing submit jobs (execution-locked resource)

When submit jobs access the same resource, you can control their execution by specifying a preset *execution-locked resource* when you submit each job.

For example, suppose that multiple jobs (programs) access the same resource, but do not have a process (function) to lock that resource. In this case, you can define the shared resource as an *execution-locked resource*, using any name of your choice. Specify the same execution-locked resource when you submit each job. Submitted jobs for which the same execution-locked resource is specified cannot be executed at the same time, even if the maximum number of concurrent jobs is set to 2 or higher.

Use the `jpqimport` or `jpqresadd` command to specify an execution-locked resource.

Cautionary note

An execution-locked resource cannot be specified for QUEUE jobs. The execution-locked resource specification applies only to jobs submitted by the `jpqjobsub` command.

10.5 Queueless jobs

Jobs can be sent directly to an agent for execution, without passing through a queue. A job that is executed without using a queue is known as a *queueless job*. The number of queueless jobs being executed concurrently is controlled on the agent (queueless agent) side. This facilitates resource control at the agent host, but control cannot be implemented on the manager side. Control on the manager side is limited to adjusting throughput by jobnet sequence control and rescheduling.

Only PC jobs, and Unix jobs, and action jobs that are defined in a jobnet can be executed as queueless jobs. No other types of jobs can be executed as queueless jobs. This includes jobs that involve linkage with JP1/NQSEEXEC or JP1/OJE.

To define a PC job, Unix job, or action job as a queueless job, when defining the job details select the JP1/AJS3 Queueless Agent service (**Queueless Agent**) as the execution service.

Queueless jobs do not use execution agents. To define a queueless job, specify the execution host in **Target agent** in the detailed definition of a job.

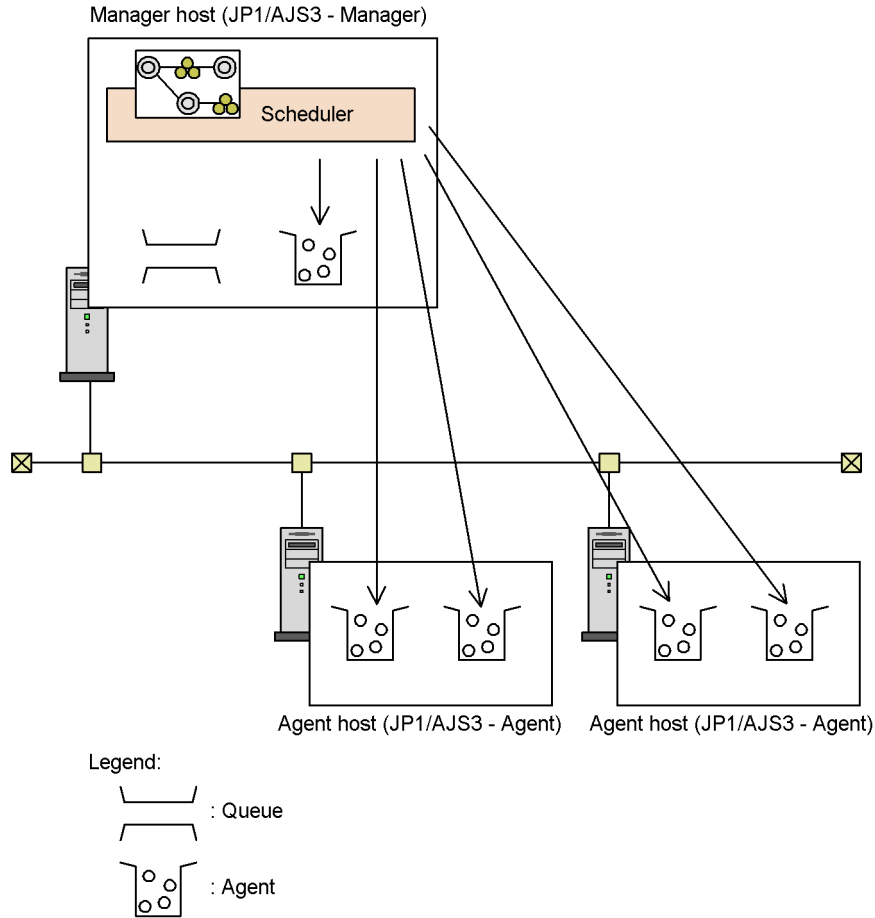
Before you can use queueless jobs, you must first set up the execution environment for queueless jobs on both the manager and agent. For the required setup, see *2.7 Setting up the queueless job execution environment* in the *Job Management Partner 1/ Automatic Job Management System 3 Configuration Guide 2*.

(1) Execution environment for queueless jobs

Concurrent execution and other aspects of queueless job control are managed by the agent's JP1/AJS3 Queueless Agent service. The service must be started on the agent host before queueless jobs can be executed.

The following figure shows how queueless jobs are executed.

Figure 10-25: Processing flow when executing queueless jobs



Queueless jobs in a jobnet are sent directly from the scheduler to the agent (queueless agent). This is a particular advantage as regards job execution performance when multiple scheduler services are configured in the system.

To execute a queueless job, the name of the JP1 user who issued the job execution request, and the host name of the manager, must be mapped to the OS user of the agent host. To execute a queueless job using a specified OS user, that OS user must be mapped to a JP1 user.

During execution of a queueless job, the communication line between the manager and agent is kept alive to reduce the number of connections and disconnections. Unlike queued jobs, whose status is polled at regular intervals, queueless jobs are not polled. An error in a queueless job is discovered through detection of a disconnection during

its execution.

(2) Status transition of queueless jobs

Because they bypass the queue, queueless jobs skip the *Now queuing* stage. Whereas queued jobs move from *Waiting to execute*, to *Now queuing*, and then to *Now running* (or *Failed to start*) status, queueless jobs move directly from *Waiting to execute* to *Now running* (or *Failed to start*) status.

(3) Restricting the number of concurrent queueless jobs

You can restrict the number of queueless jobs that can be executed concurrently by an agent host. To do so, set the maximum number of concurrent jobs in that agent's Queueless Agent service. To distribute the processing load in line with your system operation, you can create classes within the Queueless Agent service and set a different concurrent execution limit for each class.

When the limit is reached, subsequent jobs are managed in the Queueless Agent service memory and continue to wait for execution, up to the maximum number of waiting jobs set for the Queueless Agent service or for the specified class. Once this maximum number is exceeded, the next waiting job is placed in *Failed to start* status.

(a) Setting the maximum number of concurrent queueless jobs

To specify the maximum number of concurrent jobs and waiting jobs in the Queueless Agent service, set the environment setting parameters `AJSQL_JOBMAX` and `AJSQL_JOBWAITMAX`, respectively. Use the `jbssetcnf` command to set these parameters. For details about the parameters, see 2.7 *Setting up the queueless job execution environment* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 2*.

Using the `ajsqlalter` command, you can change the maximum number of concurrent jobs and waiting jobs while the Queueless Agent service is active. When the maximum number of concurrent jobs is set to zero, queueless jobs are not executed but are managed in the Queueless Agent service memory. They continue waiting to be executed until the maximum number of waiting jobs is exceeded. Change the maximum concurrent jobs and maximum waiting jobs to suit your particular system. For details about the `ajsqlalter` command, see `ajsqlalter` in 3. *Commands Used for Special Operation* in the manual *Job Management Partner 1/Automatic Job Management System 3 Command Reference 2*.

(b) Distributing queueless jobs based on class-specific concurrent execution limits

When you register a queueless job, you can assign it to a particular class that has a preset limit on the maximum number of jobs that the agent can execute concurrently. Class specification enables lock control of queueless jobs and the setting of time restrictions when particular jobs take precedence, helping to distribute the processing load in line with your system operation.

For details on setting the maximum concurrent jobs and maximum wait jobs per class, see *6.4.1 Executing jobs with a class specified in a queueless job environment* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (Windows)* or see *14.4.1 Executing jobs with a class specified in a queueless job environment* in the *Job Management Partner 1/Automatic Job Management System 3 Configuration Guide 1 (UNIX)*.

For example, suppose that you want to execute a particular job on its own because it uses an execution-locked resource. You can do this by defining a class and setting the maximum number of concurrent jobs in that class to 1. Then register the job in that class. As another example, let's say you want certain jobs to be executed during a particular time period. To do so, increase the concurrent execution limit for the class in which the jobs are to be registered, and reduce the limit for other classes. This allows targeted jobs to be executed preferentially.

(c) Relationship between the maximum number of concurrent execution jobs specified per class or specified for the Queueless Agent service as a whole

When the maximum number of concurrent jobs set for the Queueless Agent service is exceeded, the job is placed in wait state even if the maximum number of concurrent jobs set for the class in which the queueless job is registered has not been reached. Also, if the maximum number of waiting jobs set for the queueless agent is exceeded, the next waiting job enters *Failed to start* status even if the maximum number of waiting jobs set for the class has not been reached.

Appendixes

- A. Information Passed by Event Jobs
- B. Version Revisions
- C. Changes in 3020-3-S02-04(E)
- D. Glossary

A. Information Passed by Event Jobs

This appendix describes the information that can be inherited from an event job.

Information received by an event job can be referenced by succeeding jobs or jobnets. To enable referencing, open the Detailed Definition - [Receive JP1 Event] - [Passing Information] dialog box, then set the specific passing information as a macro variable. The information you can set in the macro variable depends on the type of event job. For examples of defining a macro variable, see 2.4.4(6) *Passing information received by an event job* in the *Job Management Partner 1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

The following table lists the passing information for each type of event job.

Table A-1: Passing information for each type of event job

Icon name	Variable	Description	Windows	UNIX	Size (bytes)
Common	CMTMOU ^{#1}	Whether the job timed out ε: Timed out £: Other than timeout	Y	Y	1
Receive JP1 event	EVID	Event ID	Y	Y	17
	EVUSR	Source user name	Y	Y	20
	EVGRP	Source group name	Y	Y	20
	EVHOST	Source event server	Y	Y	255
	EVIADDR	Source IP address	Y	Y	16
	EVMSG	Message text	Y	Y	1,023
	EVDETAIL	Detailed event information ^{#2}	Y	Y	1,023
	EVSEV	Extended severity levels	Y	Y	11
	EV: <i>extended-attribute-name</i> ^{#3}	Optional extended attribute	Y	Y	4,085
	EVENV1 to EVENV ^{#4}	Extracted data	Y ^{#5}	Y	4,085
	EVUSRID	Source user ID	Y	Y	10
	EVGRPID	Source group ID	Y	Y	10

Icon name	Variable	Description	Windows	UNIX	Size (bytes)
	EVPROCESSID	Source process ID	Y	Y	10
	EVDATE	Event date (yyyy/mm/dd)	Y	Y	10
	EVTIME	Event time (hh:mm:ss)	Y	Y	8
Monitoring files	FLFNAME	Full path of changed file	Y	Y	Windows: 258 UNIX: 509
	FLCOND	One of the following established monitoring conditions: c (Create) d (Delete) s (Change size) m (Final time write)	Y	Y	1
	FLCTIME	File update time (yyyy/mm/dd. hh:mm:ss)	Y	Y	19
	FLSIZE	Size of changed file	Y	Y	20
Receive mail	MLRCVADDRESS	Sender of received mail ^{#6}	Y	Y	Windows: 256 ^{#7} UNIX: 4,085
	MLRCVSUBJECT	Subject of received mail	Y	Y	Windows: 256 ^{#8} UNIX: 4,085
	MLRCVBODY	Name of file containing received mail text	Y	--	258
	MLRCVATTACHFILE ⁿⁿ (nn: 01 to 20)	Name of attached file	Y	--	258 ^{#9}
	MLRCVATTACHLIST	Name of file listing attached files	Y	--	258 ^{#10}
	MLRCVMAILBODY	Name of file containing the body of received mail (1 file set internally)	--	Y	452

A. Information Passed by Event Jobs

Icon name	Variable	Description	Windows	UNIX	Size (bytes)
	MLRCVTIME	Mail arrival time	Y	Y	40
Receive MQ message	MQRVCORRELATION	Correlation ID of received message	Y	Y	24
	MQRVDISCRIMINATION	Message ID of received message	Y	Y	24
	MQRVCQUEUE	Name of queue which received the message	Y	Y	48
	MQRVMODELQUEUE	Model queue name	Y	Y	48
	MQRVMESSAGEFILE	Name of file in which received message was stored as a message structure (1 file set internally)	Y	Y	Windows: 258 UNIX: 452
Receive MSMQ message	MSRCVQUEUEPATH	Queue path of received message	Y	--	259
	MSRCVMUTUAL	Correlation ID of received message	Y	--	20
	MSRCVMESSELABEL	Message label of received message	Y	--	249
	MSRCVAPPLICATION	Application information about received message (hex numeric)	Y	--	8
	MSRCVMESSAGEFILE	Name of file in which received message was stored as a message structure (1 file set internally)	Y	--	258
Log file trapping	LFFNAME	Full path of trapped log file	Y	Y	258
	LFDATA	Trapped data (truncated if over 511 bytes)	Y	Y	511
Monitoring event log	NELOG	Log type	Y	--	11
	NEEVKIND	Event type	Y	--	20

Icon name	Variable	Description	Windows	UNIX	Size (bytes)
	NESOURCE	Source	Y	--	255
	NECLASS	Class	Y	--	4,085
	NEEVID	Event ID	Y	--	16
	NEDETAIL	Explanation (truncated if over 1,023 bytes)	Y	--	1,023
Interval control	N/A	N/A	N/A	N/A	N/A

Legend:

Y: Can be specified

--: Cannot be specified

N/A: Not applicable

Note1

The following characters are replaced by blanks if contained in the passing information:

\r, \n, \b, and \f

The size is the string length, excluding \0.

Note2

Estimate the sizes of the macro variable names and passing information in advance so that their total size does not exceed 4,096 bytes. Be especially careful when start conditions are combined using AND. In this case, the macro variables and passing information of all event jobs defined in the start conditions are merged. The data merged in this way could easily exceed 4,096 bytes.

#1

If the event job ends normally on detecting an event, f is stored in CMTMOUT. If the timeout period is exceeded, t is stored in CMTMOUT. If the event job ends abnormally due to an error, nothing is stored in CMTMOUT.

#2

Set only if the detailed information in the JP1 event is text.

#3

The *extended-attribute-name* is a character string of up to 32 bytes, determined

by the JP1/Base event service. For details on extended attribute names, see the *Job Management Partner 1/Base User's Guide*.

#4

The extracted data, if any, is stored in the following sequence in `EVENV1` to `EVENV9`, after regular expressions are checked. Note that this sequence differs from the order in which the parameters are specified in the `ajsdefine` command.

- Source user name
- Source group name
- Source event server
- Message text
- Detailed event information
- Optional extended attribute

Within the extended attribute, regular expressions are checked in the order in which the JP1 events were received, and the extracted data, if any, is stored in order in `EVENV1` to `EVENV9`.

#5

This is set only when extended regular expressions are used.

#6

In Windows, this variable is set to the type of information specified for the `NextAddress` environment setting parameter. If you specify `Address`, the sender's email address is set. If you specify `Nickname`, the sender's display name (nickname) is set.

This information is not set on receipt of a GroupMail message when `Address` is specified for the `NextAddress` environment setting parameter.

For the setup to link with a mail system on a Windows host, see 2.2.4 *Setting up the environment for the mail system linkage* in the *Job Management Partner 1/ Automatic Job Management System 3 Linkage Guide*.

#7

Depending on the email system you are using, information longer than the stipulated value may be set if the sender has a long address. In this case, only the first 256 bytes are passed. Check the manual for your email system to find out whether values over 256 bytes may be set.

#8

Depending on the email system you are using, information longer than the

stipulated value may be set if the subject line is long. In this case, only the first 256 bytes are passed. Check the manual for your email system to find out whether values over 256 bytes may be set.

#9

Each attached file is saved, and the file name is set to the full path.

Multiple attached files are not saved in any particular order. When 21 or more files are received, only 20 file names can be passed.

#10

The attached files are saved and a list of the files is created. The list file name is set to the full path.

B. Version Revisions

This appendix lists the changes in each version of the JP1/AJS series programs.

B.1 Revisions in 09-00

The following lists the revisions in 09-00 for each program.

(1) JP1/AJS3 - Manager

- The standard database of JP1/AJS3 is now an embedded database.
- Functions related to an embedded database have been changed as follows:
 - The sizes of the large-scale, medium-scale, and small-scale database models have been changed.
 - The database area auto-increment function and the system log auto-increment function have been added.
 - The system log is no longer used.
 - The functions of the commands used to control an embedded database have been enhanced.
- The ISAM database is now used only for QUEUE jobs and submit jobs.
- An agent management function has been added for specifying a logical execution agent name as the destination host for a job or jobnet. Previously, users could only specify execution hosts by their real names.
- Jobs that are in the *Now queuing* status when the service is stopped are now returned to the *Wait for prev. to end* status when the service restarts (in hot-start mode), before being resubmitted.
- A jobnet release function has been added for replacing the definition of a jobnet that is registered for execution with another definition.
- The job execution control manager process (jqman) and event/action control manager process (jpomanager) can now be started on a scheduler service basis.
- A scheduler log file can now be output for an individual scheduler service or host.
- The following functions have been enhanced:
 - The method by which the number of logs to keep is managed
 - The process by which monitored generations of jobnets with start conditions are established
 - The process by which execution generations when a start condition is established are held

- A format specification has been added to the `ajsshow` command for outputting the standard output file name.
- The Manager Environment Settings dialog box is no longer provided. Instead, you can use the `jajs_config` command to set up the manager environment.
- A function has been added to support end delay monitoring based on how long a job takes to execute.
- The jobnet connector functionality has been enhanced to enable control of the execution order of root jobnets managed by different scheduler services.
- The definition pre-check has been enhanced so that if an invalid execution order is found in the units of the jobnet being checked, the names of the units are output to the check results file.
- The file permission check performed at execution of a Unix job has been enhanced to include checks of the access control list and secondary group settings as well as file permissions.
- A function has been added that enables event jobs to continue executing even if the JP1/AJS3 service stops on the execution host.
- A function has been added for exporting and importing the registration statuses of jobnets as registered execution-schedule information.
- Linkage with message queues on UNIX hosts (TP1/LiNK, TP1/Message Queue, MQSeries) is no longer supported.
- Windows Server 2008 has been added as platforms supported by JP1/AJS3 - Manager.
- A unit called a jobnet connector which controls the execution order of root jobnets has been added.
- An option has been added to output a detailed history of user operations, such as changes to jobnet definitions, to the scheduler log.
- The `ajslogprint` command for extracting log entries from the scheduler log has been added.

(2) JP1/AJS3 - Agent

- The Agent Environment Settings dialog box is no longer provided. Instead, you can use the `jajs_config` command to set up the agent environment.
- Linkage with a message queue system is no longer supported.
- The file permission check performed at execution of a Unix job has been enhanced to include checks of the access control list and secondary group settings as well as file permissions.
- Linkage with message queues on UNIX hosts (TP1/LiNK, TP1/Message Queue,

MQSeries) is no longer supported.

- Windows Server has been added as platforms supported by JP1/AJS3 - Agent.

(3) JP1/AJS3 - View

- An agent management function has been added for specifying a logical execution agent name as the destination host for a job or jobnet. Previously, users could only specify execution hosts by their real names.
- A jobnet release function has been added for replacing the definition of a jobnet that is registered for execution with another definition.
- Function menus have been added to the JP1/AJS3 - View window to facilitate task-oriented operation.
- The JP1/AJS3 - View window (Summary Monitor window) has been added. In this window, you can view the progress of jobnets and other information.
- JP1/AJS3 - View can now be started in the following modes:
 - Normal mode
In this mode, the JP1/AJS3 - View window is equipped with function menus.
 - Monitoring mode
A mode dedicated to monitoring jobs and jobnets. Only the JP1/AJS3 - View window (Summary Monitor window) is displayed.
 - Compatible mode
JP1/AJS3 - View operates in the same way as JP1/AJS2 - View version 8 or earlier.
- A Detailed Information area has been added to the JP1/AJS3 - View window (Main window), which displays detailed information about a unit.
- The concurrent execution setting of monitored generations and the holding behavior of execution generations (produced when a start condition is satisfied) can now be selected in the detailed definition of a start condition.
- A list filter function has been added for filtering the information in a list.
- A function has been added for saving list information in CSV format.
- You can now click a button in the Daily Schedule window and Monthly Schedule window to move between days and months.
- A list area has been added to the Jobnet Editor window and Jobnet Monitor window. This area displays the jobs defined in the jobnet.
- A Search window has been added, in which you can set detailed search conditions and perform operations on units listed in the search results.

- You can now use a mouse wheel to scroll inside JP1/AJS3 - View.
- A function has been added that allows you to select whether **Type** in list areas are grouped by type or displayed in detailed format.
- A function has been added for prohibiting changes to specific definition items in the Define Details dialog box.
- A function has been added for removing icons you no longer use from the icon list area in the Jobnet Editor window.
- Windows 7 has been added as a supported OS (JP1/AJS3 - View 09-00-05 or later).
- A function has been added to support end delay monitoring based on how long a job takes to execute.
- The jobnet connector functionality has been enhanced to enable control of the execution order of root jobnets managed by different scheduler services.
- An option has been added to the Filter Settings dialog box so that jobnets with hold plans can be treated as jobnets in *Being held* status for filtering purposes in the Daily Schedule window and Monthly Schedule window.
- The ability to define, operate, and monitor jobnet connectors which control the execution order of root jobnets has been added.
- A function that displays the preceding and succeeding jobs of a given job or jobnet in bold has been added.
- Support for Windows Vista has been added.

B.2 Revisions in 08-00

The following lists the revisions in 08-00 for each program.

(1) JP1/AJS2 - Manager

- The recommended values for the environment settings are now set during installation and setup.
- A Monitoring Files job can now monitor files larger than 2 gigabytes (large files).
- The `ajsstatus` command can now output the connection status of JP1/AJS2 - View.
- The following commands used to control an embedded database have been added:
 - `ajsembdbaddarea` command (expands a database area in an embedded database)
 - `ajsembdbaddlog` command (expands a log area in an embedded database)
 - `ajsembdbcancel` command (cancels execution of a command)

manipulating an embedded database)

- `ajsembdboplog` command (manipulates embedded database logs)
- `ajsembdbreclaim` command (maintains an embedded database)
- `ajsembdbrolog` command (unloads and reloads an embedded database)
- `ajsembdbstr` command (backs up and restores an embedded database)
- `ajsembdbstart` command (starts an embedded database)
- `ajsembdbstatus` command (monitors an embedded database)
- `ajsembdbstop` command (stops an embedded database)
- `ajsembdbunset` command (removes the setup of an embedded database)

With support of the `ajsembdbreclaim` command, the time required to reclaim free pages has been reduced.

- JP1/Performance Management - Agent Option for JP1/AJS2 can now be linked with JP1/AJS2 to analyze the operating status.
- The `jajs_start` command and the `jajs_start.cluster` command can now check whether a process has already been started when JP1/AJS2 is started. (UNIX only)

(2) JP1/AJS2 - Agent

- The recommended values for the environment settings are now set during installation and setup.
- A Monitoring Files job can now monitor files larger than 2 gigabytes (large files).

(3) JP1/AJS2 - View

- Icons have been changed.

B.3 Revisions in 07-50

(1) JP1/AJS2 - Manager

- Macro variables can now be used during registration for execution to specify information to be passed.
- Judgment jobs can now perform variable judgment.
- A function has been added that suppresses jobnet executions that follow an abnormally terminated jobnet and that will be started when their start conditions are satisfied.
- A definition pre-check function has been added for conducting a final check before starting production in the production environment after the unit definitions are migrated from the development environment.

- The `jpomanevreset` command has been added for deleting data accumulated in the event action manager if a large amount of unprocessed data accumulated in the event action manager has caused delay. To identify the start conditions and agents that have caused this problem, the `jpomanevshow` command has also been added for displaying information about agents that frequently send data to the manager and the start conditions.
- A function that alleviates consumption of the Desktop heap has been added. (Windows only)
- A function has been added for specifying the maximum wait time for the scheduler service to connect to a database.
- Messages that were output to only the integrated trace log can now be output to syslog also. (UNIX only)
- The following functions have been added to the data collection tool:
 - Specifying a logical host name
 - Filtering the data to be collected
 - Adding types of data that can be collected
- Descriptions of messages have been improved.
- An urgent command has been added that can be executed if an error occurs.
- A function has been added that places limits on, for example, the size of files that can be received, to prevent a part of job processing from affecting the entire system operation.
- A function has been added that performs a synchronized write when updating event job information or the wait information file.
- The monitoring interval for linkage with MQ Series can now be specified in seconds.
- If a TCP/IP connection error occurs, the retry interval and count can now be changed.
- The policy to determine the agent hosts to which a job will be dispatched can now be specified.
- All the detailed processes of the event action function can now be stopped to terminate the agent process for the event action function if any of the detailed processes have terminated upon receiving a signal.
- Microsoft(R) Visual C++ .NET Version 2003 is now supported as a compiler for the provided code functions.
- The `ajsshow` command can now display the hold attribute of a jobnet or job even when the jobnet or job has already terminated.

(2) JP1/AJS2 - Agent

- A definition pre-check function has been added for conducting a final check before starting production in the production environment after the unit definitions are migrated from the development environment.
- The following functions have been added to the data collection tool:
 - Specifying a logical host name
 - Filtering the data to be collected
 - Adding types of data that can be collected
- Descriptions of messages have been improved.
- The monitoring interval for linkage with MQ Series can now be specified in seconds.
- All the detailed processes of the event action function can now be stopped to terminate the agent process for the event action function if any of the detailed processes have terminated upon receiving a signal.
- A function has been added that performs a synchronized write when updating event job information or the wait information file.

(3) JP1/AJS2 - View

- Macro variables can now be used during registration for execution to specify information to be passed.
- Judgment jobs can now perform variable judgment.
- A function has been added that suppresses the jobnet executions that follow an abnormally terminated jobnet and that will be started when their start conditions are satisfied.
- The **Add**, **Change Time**, **Execute Immediately**, and **Release Change** options have been added to the JP1/AJS2 - View window.
- The **Paste (Extension)** menu command has been added for copying units and relationship lines at the same time.
- Relationship lines can now be drawn from multiple units to a single job network element.
- When opening the Jobnet Monitor window of JP1/AJS2 - View from JP1/AJS2 Console View, if there is already an activated JP1/AJS2 - View, the window can now be opened in JP1/AJS2 - View.
- The following functions have been added to the data collection tool:
 - Specifying a logical host name

- Filtering the data to be collected
- Adding types of data that can be collected
- Descriptions of messages have been improved.
- The maximum log file size for JP1/AJS2 - View has been increased.
- The maximum log file size for JP1/AJS2 Console View has been increased.
- In JP1/AJS2 - View, log information that previously was output many times in small units can now be output at one time.
- In JP1/AJS2 Console View, log information that previously was output many times in small units can now be output at one time.
- In the Windows version of JP1/AJS2 - View, **Help** has been added to the **Start** menu.

B.4 Revisions in 07-00

Version 07-00 features the addition of a new program, JP1/AJS2 - Advanced Manager, to enable the use of an embedded database (HiRDB) in a JP1/AJS2 scheduler database.

This section explains the changes in each version from 06-71 to 07-00, on a program-by-program basis.

(1) *About JP1/AJS2 - Manager*

- A function was provided to temporarily compress JP1/AJS2 and reconfigure the ISAM database (scheduler database and job execution environment database) without stopping active applications.
- ISAM databases can now be reconfigured in parallel.
- The number of scheduler services that can be added has been changed from 9 to 20.
- An option was added for outputting the execution timings of reference commands, such as `ajsshows` and the history of service processing requests from operation commands, as the operation log to the scheduler log.
- The number of logs to keep for a jobnet has been changed from 99 to 999.
- For a cold start of JP1/AJS2, the job execution environment database is deleted so that the startup time of JP1/AJS2 becomes shorter.
- A function is now supported for validating the user profile information in the environment setup for job execution control.
- By setting the number of days that job information is held to 0 days, jobs that terminate abnormally can now be handled by changing the save time.
- The JP1/AJS2 job information deletion can now be suppressed.

- Any event job can now be used in a DNS environment (host name in the FQDN format).
- Event job reception information can now be inherited as macro variables as the parameters of standard jobs and action jobs without having to pay attention to double quotation marks in the inherited information.
- The extended regular expression supported by JP1/Base can now be used in Receive event job monitoring jobs, Monitoring log files jobs, and Monitoring event log jobs according to the JP1/Base settings.
- A function to execute queueless jobs is now supported.

(2) About JP1/AJS2 - Agent

- Event job reception information can now be inherited as macro variables of the parameters of standard jobs and action jobs without being aware of double quotation marks in the inherited information.
- A function for executing queueless jobs was supported.
- When JP1/AJS2 - Agent starts, it no longer accesses the authentication server (07-00-/C or later).

(3) About JP1/AJS2 - View

- A user profile can now be used to set the JP1/AJS2 - View environment.
- A line feed character can now be inserted at any point in a unit name displayed in the map area of the Jobnet Editor and Jobnet Monitor windows.
- The default values in the dialog box can now be changed.
- Display items (columns) in the following locations can now be selected.
 - List area in the JP1/AJS2 - View window
 - Execution result list in the Daily Schedule window
 - Execution result list in the Monthly Schedule window

C. Changes in 3020-3-S02-04(E)

The following table list the changes in this manual (3020-3-S02-04(E)).

Table C-1: Changes in 3020-3-S02-04(E)

No.	Location	Changes
1	All	Windows 7 has been added as an OS supported by JP1/AJS3 - View.
2	3.1.1(1)	Descriptions of types and execution services have been added to the description of the attribute information for a job.
3	3.4	Descriptions related to start conditions have been improved.
4	4.5.11(5)	A note on executing the succeeding unit of a unit that has terminated abnormally has been added to the notes about rerunning jobnets or jobs.
5	5.1 5.2	The description of restricting job transfer for execution agents and jobs has been improved.
6	6.1.1	A description of units displayed with an -R mark when the <code>ajsshows</code> command is executed to display the status has been added.
7	8.1.1	Descriptions related to access permission for agent management information have been added.
8	8.2.1	The description of the unit owner permission has been improved.
9	10.2	The description of remote jobnets has been improved.

D. Glossary

abnormal end

A jobnet ends abnormally if one of the processes defined in the jobnet fails to execute properly. The jobnet is interrupted at that point and subsequent processes are not executed.

A job ends abnormally if it fails to execute properly. The process is interrupted at that point.

The embedded database system ends abnormally when an error causes its status to change from active to stopped or paused, without any intervention by the user. For details, see *D. How the Embedded Database Operates* in the manual *Job Management Partner 1/Automatic Job Management System 3 Troubleshooting*.

abnormal threshold

A value that is compared with a job's return code to evaluate whether the job ended normally or abnormally.

action job

A job that sends email, or sends events reporting the system status to JP1/ IM or the HP NNM.

agent host

A host that executes jobs on request from a manager host. JP1/AJS3 - Agent must be installed on the agent host, or since JP1/AJS3 - Manager also provides JP1/AJS3 - Agent functionality, JP1/AJS3 - Manager might be installed on the agent host.

The agent host executes the job on receipt of a job request from the manager host. At completion of the job, the agent host receives the execution result (return value) of the executable file and forwards it to the manager host.

AJS3 unit monitored object

An object for monitoring the status of root jobnets in JP1/AJS3. By defining the monitoring conditions in this object, you can then switch to monitoring mode and monitor the root jobnets.

AJSPATH

An environment variable for defining the paths used by JP1/AJS3. When this environment variable is defined, you do not need to specify the full path when specifying a jobnet name in a command.

backup box

A directory or a folder for storing backup files.

backup file

A file containing the units defined in JP1/AJS3.

base day

A date specified as the starting day of the month in the calendar information.

base time

The time that marks when a day ends and the next day begins in a JP1/AJS3 system. For example, if 8:00 a.m. is set as the base time, the previous day is regarded as lasting until 7:59 a.m.

calendar information

Information about open days and closed days for jobnet execution. You can define calendar information separately for each job group. The calendar information specifies the days on which jobnets in the job group can and cannot be executed. (When the processing cycle falls on a closed day, the jobnet can be executed on another day if a substitute schedule is defined.) For open days, you can specify the base day, base month, and base time.

closed day

A day on which jobnets are not executed. However, if **Execute without shift** is specified, the jobnet will be executed on that closed day.

cluster system

A system configured as multiple linked server systems, designed to continue operation even if one system fails. If a failure occurs in the server currently executing applications (primary node), the other standby server (secondary node) takes over and continues processing the applications. Therefore, a cluster system is also referred to as a *node switching system*.

The term *cluster system* can also mean load balancing based on parallel processing. In this manual, however, *cluster system* refers only to node-switching functionality for preventing interruption of application processing.

common user profile

A file containing the environment settings for JP1/AJS3 - View, accessible to all JP1 users. The system administrator saves the common user profile in JP1/AJS3 - Manager. JP1 users can download this file, enabling the same JP1/AJS3 - View environment to be set for all JP1 users.

A common user profile is useful when a large number of JP1 users will be using JP1/AJS3 - View in the same environment.

compatible ISAM configuration

A system configuration in which JP1/AJS3 information is managed exclusively by the

ISAM database.

This configuration is offered to help users migrate from JP1/AJS2 version 8 or earlier. It can restrict to the same degree as in previous versions, the use of resources such as hard disk and memory. However, from version 9 only a subset of the new features offered is provided.

correlation ID

Information for identifying sent and received messages. The correlation ID is received in the character code set specified by the sender.

custom job

A predefined job for executing a task with a specific purpose. JP1/AJS3 provides standard custom jobs such as file transfer and job requests to a mainframe. In addition, you can register your own frequently used jobs as custom jobs. When registering a custom job, you can represent it by creating an icon with a special shape and design, and you can create a dialog box for entering job information.

To use a custom job, the requisite program for the job must be installed.

Daily Schedule window

A window that displays each day's execution schedules, execution status, and execution results.

default queue

A queue created in an agent host for executing jobs. You must always create a default queue.

When you submit a job for execution, if you specify an agent host name as the destination, the job will be submitted to the default queue of the specified agent host.

dependent job

A job executed when the judgment result of a judgment job is true.

dependent jobnet

A jobnet executed when the judgment result of a judgment job is true.

embedded database

The standard database of JP1/AJS3. An embedded database offers high reliability, and is well suited to large-scale systems that handle large quantities of information.

embedded database administrator (database administrator)

A user authorized to assign and cancel various permissions for an embedded database (a user with DBA permissions).

Database administrators are managed within an embedded database.

embedded database operation commands

A generic term for commands whose name begins with `ajsembddb`.

embedded database service

A service that provides the environment for using the embedded database in Windows. This service must be started before you can use the embedded database. The name of the embedded database service is `JP1/AJS3 Database setup-identifier`.

embedded database system administrator

The owner of an embedded database practical directory and embedded database file system areas (data area and system area). The embedded database system administrator can execute commands for an embedded database.

The OS manages embedded database system administrators.

end with warning

A status indicating that a jobnet finished, but some of the processes defined in the jobnet were executed incorrectly. The jobnet continues to the end without interruption.

This ending method is used when an error is not so serious as to terminate the jobnet.

environment setting parameter

A parameter for defining the information required to operate JP1/AJS3, written in an environment settings file. With these parameters, you can specify the directory in which information about JP1/AJS3 units is stored, whether to output syslog messages, and other such preferences.

environment settings file

A file containing the settings required to operate JP1/AJS3, such as the scheduler service environment and job execution environment.

event

A specific event, such as email reception or file update, that occurred in the system. Events can be used to start a job or jobnet, and can be monitored using an event job.

event job

A job that monitors specific events occurring in the system. When an event job is initiated, it starts monitoring for file updates, incoming messages, or other specified events.

execution agent

The logical name of an agent host that executes jobs or jobnets. Based on the agent information defined in the manager host, the manager maps the execution agent specified in the job or jobnet to the physical host name of the agent host, and distributes the job or jobnet accordingly.

execution agent group

A group of execution agents configured to realize load distribution. The manager distributes jobs among the execution agents according to their assigned priorities.

execution ID

A number assigned to an execution schedule of the uppermost jobnet.

execution-locked resource

A means of preventing multiple jobs from executing at the same time, by specifying the same resource name (execution-locked resource name) for each job.

fixed execution registration

A method of registering a jobnet so that it starts and runs at a predetermined date and time calculated by the system from schedule definitions.

fixed schedule

A schedule set by absolute times when a jobnet is registered for fixed execution.

HP NNM

A suite of integrated network management tools from Hewlett-Packard Co. for managing network configuration, performance, and failures.

immediate execution registration

A method for starting and processing a jobnet immediately after registering it for execution.

ISAM database

The database that manages the execution environment for QUEUE jobs and submit jobs. Data is indexed using the Indexed Sequential Access Method (ISAM) and is managed in the database. The ISAM database is provided as standard with JP1/Base.

job

A group of commands, shell scripts, or Windows executable files.

job execution environment

A job execution environment consists of a JP1/AJS3 manager and agents.

The job execution environment for the manager is used to manage the definition information for execution agents (such as the maximum number of concurrently executable jobs and job transfer restriction status), job distribution method, and job execution results.

The job execution environment for the agent is used mainly to manage how a job is executed.

These job execution environments are managed by using a database and environment setting parameters.

When QUEUE jobs and submitted jobs are used, the ISAM database and environment setting parameters are used as the job execution environment for the QUEUE jobs and submitted jobs.

Note that queueless jobs are managed in the queueless job execution environment.

job group

A folder for classifying and managing jobnets.

job network element

The generic term for these elements is *unit*.

jobnet

A set of jobs associated in execution order. When a jobnet is executed, the jobs in the jobnet are automatically executed in their predetermined order.

jobnet connector

A unit for controlling the execution order of root jobnets. A jobnet connector establishes connections between root jobnets and controls their execution order by having connected generations wait for their counterparts to start or finish.

Jobnet Editor window

A window in which you can create new jobnets or edit existing jobnets.

Jobnet Monitor window

A window that displays the execution status or detailed execution results of jobnets or jobs. You can manipulate jobnets or jobs in this window.

JP1 event

Event information that is reported to JP1/Base when an event occurs in the system. JP1 events are reported to other systems via JP1/Base.

JP1 permission level

A name that indicates the operations that a JP1 user is allowed to perform on management targets (resources) defined in JP1/AJS3, including applications and events. Use JP1/Base to define JP1 permission levels.

JP1 resource group

A name given to a specific JP1/AJS3 unit for controlling access by JP1 users to that unit.

JP1 user

A user designation for using JP1/AJS3 or JP1/IM - Manager. Each JP1 user is registered in the authentication server, which controls the user's access to management targets (resources).

JP1/AJS3 - Definition Assistant

This program allows you to register a large amount of JP1/AJS3 definition information edited using an Excel template into a manager host, or to retrieve JP1/AJS3 definition information from a manager host to an Excel template. The Excel templates provided by JP1/AJS3 - Definition Assistant are called *definition management templates*. With a definition management template in the spreadsheet format, you can enter or edit definition information efficiently by using automatic filling, automatic filtering, and other Excel functionalities.

JP1/AJS3 Console Agent

A JP1/AJS3 component that regularly monitors the status of objects (root jobnets) on the local host, specified in JP1/AJS3 Console Manager. Any change in status is notified to JP1/AJS3 Console Manager.

JP1/AJS3 Console Manager

A JP1/AJS3 component that stores definitions about monitored objects defined in JP1/AJS3 Console View, and gets status information about monitored objects by issuing requests to JP1/AJS3 Console Agent.

JP1/AJS3 Console View

A JP1/AJS3 component that allows you to define objects to be monitored, using a graphical user interface. The definitions are stored in JP1/AJS3 Console Manager. Using JP1/AJS3 Console View, you can view and monitor the status of target objects notified by JP1/AJS3 Console Agent to JP1/AJS3 Console Manager. You need to log in to JP1/AJS3 Console Manager before using JP1/AJS3 Console View.

JP1/AJS3 for Enterprise Applications

A program that allows you to control jobs in an R/3 system from another system. You can submit, delete, and monitor R/3 jobs.

R/3 jobs can be executed automatically from JP1/AJS3 if you register them as custom jobs for JP1/AJS3 for Enterprise Applications when you define a JP1/AJS3 jobnet.

JP1/AJS3 for Enterprise Applications is the successor to JP1/Application Manager for R/3.

JP1/AJS2 for Oracle E-Business Suite

A program that allows you to access Oracle E-Business Suite from another system and to request concurrent execution of applications.

Requests for concurrent execution can be issued from JP1/AJS3 if you register the requests as custom jobs for JP1/AJS2 for Oracle E-Business Suite when you define a JP1/AJS3 jobnet.

Using JP1/AJS3's schedule definition facility, you can specify the processing cycles and the execution dates of concurrent requests.

JP1/AJS2 for Oracle E-Business Suite is the successor to JP1/Application Manager for Oracle E-Business Suite.

JP1/Base

A program that provides the event service function. JP1/Base allows you to control the order in which services start, and it lets you send and receive JP1 events. JP1/Base is a prerequisite program for JP1/IM and JP1/AJS3. When JP1/IM is deployed in a system with JP1/AJS3, JP1/Base provides functionality for restricting operations by JP1 users.

JP1/FTP

A program for performing file transfer tasks efficiently, including file transfer/reception linked to application execution, scheduled file transfer, and automated program execution following file reception. JP1/FTP supports monitoring of transfer status, enhancing file transfer reliability.

JP1/IM

A program for centrally monitoring a distributed system. Using the windows in JP1/IM - View, the system administrator can monitor JP1 events, which provide information about job execution status or problems in the distributed system.

JP1/NQSEXEC

A program for executing routine batch processing on a distributed system and for running batch jobs efficiently.

JP1/OJE for Midrange Computer

A program for submitting batch jobs to AS/400 from a Windows or UNIX host, or for submitting batch jobs from AS/400 to a Windows or UNIX host.

JP1/OJE for VOS3

A program that links with JP1/AJS3 for executing and monitoring batch jobs between a Windows or UNIX system and a mainframe (VOS3).

JP1/Script

A program for creating and executing scripts (batch files) that control jobs on Windows. Job operation can be automated by linking JP1/Script with JP1/AJS3.

JP1/Software Distribution

A general term for a system that distributes software and manages clients using batch operations over a network.

By linking with JP1/AJS3 using the JP1/Software Distribution command interface, the user can automate software distribution and other tasks.

judgment job

A job that executes a dependent job or jobnet if the judgment result of a specified condition is true.

judgment value

A value for evaluating whether a job ended normally or abnormally.

kill

To forcibly terminate a unit being executed.

When the root jobnet is killed, all the jobs being executed are killed and the jobnets are terminated.

list file

A file containing a list of extracts from sent and received mail.

logical host

A logical server that provides the JP1 execution environment for running a cluster system. If a failure occurs on the primary node, the logical host is switched to the secondary node.

Each logical host has a unique IP address. At failover, the secondary node inherits the IP address. Thus, if the physical server fails, clients can access the secondary node using the same IP address. To the clients, it appears that one server is operating continuously.

macro variable

A variable defined for a succeeding job for referencing information received in an event. By defining a macro variable name in an event job, you can pass the event information to a succeeding job or jobnet.

Specify macro variables in the form ?AJS2xxxxxxxx? : *name-of-information-to-pass*.

mail filtering application

A program or a shell script that converts email formats.

A mail filtering application is required to convert the character set when exchanging email in formats other than RFC822.

mail receipt parameter file

A file containing the mail receipt monitoring parameters defined by the user. The file extension is `.prm`. This file is created automatically when the user defines a Receive Email Event job.

mail send parameter file

A file containing the mail send parameters defined by the user. The file extension is `.prm`. This file is created automatically when the user defines a Send Email Action job.

manager host

A host that manages jobnet definitions and schedule information in a database, and requests agent hosts to execute jobs. You must install JP1/AJS3 - Manager on the manager host.

The manager host creates jobnet execution schedules from the defined schedule information. At jobnet run time, the manager host starts the executable files defined as jobs, forwards the job definitions to an agent host, and requests the agent host to execute the jobs. When execution completes, the execution result is received by the agent host and the database is updated. Based on the updated information, the manager host executes a succeeding job or schedules the next execution of the jobnet.

manager job group

A job group for monitoring JP1/AJS3 - Manager applications from another JP1/AJS3 - Manager.

manager jobnet

A jobnet for monitoring JP1/AJS3 - Manager applications from another JP1/AJS3 - Manager.

MAPI (Messaging Application Programming Interface)

The standard messaging API for Windows.

max. shiftable days

A set number of days within which to shift the next scheduled execution date when the recalculated date falls on a closed day.

maximum number of concurrently executable jobs

The maximum number of jobs that can be executed concurrently.

message ID

One item in an MQSeries message descriptor. Message IDs are stored in the character set specified by the sender. They can be used as storage locations to help identify messages.

MIME (Multipurpose Internet Mail Extensions)

An extended SMTP function used for sending and receiving non-ASCII data.

MIME specifies various procedures, such as how data is to be transmitted between email systems, and the format of control messages for email transfer.

Monthly Schedule window

A window that displays each month's execution schedules and execution results.

nested jobnet

A jobnet defined within another jobnet.

node switching system

See *cluster system*.

normal end

A normal end of a jobnet occurs when all the processes defined in the jobnet have executed correctly and the jobnet has completed.

A normal end of a job occurs when the job has executed correctly.

open day

A day when jobnets run.

physical host

An environment unique to each of the servers (nodes) in a cluster system. When a secondary node takes over from the primary node, the environment of the physical host remains unchanged and is not inherited by the other server.

planned execution registration

A method of registering a jobnet so that it starts and executes according to schedule definitions.

planning group

A unit for switching execution among multiple root jobnets in a planned manner. Directly under a planning group, you can create a number of root jobnets, each defined differently and with differing execution schedules. This enables the root jobnets to be executed automatically in turn, according to the set schedules.

preceding job

A job executed immediately before another job or jobnet.

preceding jobnet

A jobnet executed immediately before another job or jobnet.

processing cycle

The interval between one execution start date and the next execution start date of a jobnet. By defining a processing cycle, you can execute a jobnet at regular intervals.

queue

An area for temporarily keeping jobs registered for execution. Jobs are submitted to the queue in order of registration, and are sequentially transferred for execution to the agent connected to that queue.

The queue controls the number of jobs that the agent executes concurrently, thereby preventing any degradation in performance caused by a large number of jobs being executed at the same time.

queueless job

A job transferred directly from the manager to an agent host for execution, without using a queue. Queueless jobs simplify processing because they are not managed in a queue by the job execution control. As a result, they offer better performance than ordinary queued jobs, allowing more jobs to be executed within a given period of time. However, job execution control functions such as execution agent names and execution agent groups are not available with queueless jobs.

You can define PC jobs and Unix jobs in a jobnet as queueless jobs by specifying **Queueless Agent** as the execution service.

Unless otherwise indicated, the descriptions in this manual apply to jobs for which **Standard** is specified as the execution service.

queueless job execution environment

A queueless job execution environment consists of execution environments for the JPI/AJS3 manager (scheduler service and queueless file transfer service) and queueless agents (queueless agent services). The execution of queueless jobs is managed by using the environment setting parameters for the job execution environment.

Note that the job execution environment must be set up by using the `ajsqlsetup` command before environment setting parameters are set.

queuing job

A job submitted directly to a queue and waiting to be executed.

recovery job

A job to be executed when a job or jobnet ends abnormally.

recovery jobnet

A jobnet to be executed when a job or jobnet ends abnormally.

schedule by days from start

A schedule defined for recalculating the next scheduled execution date, using as the base day the next scheduled execution date determined from the execution start time, processing cycle, and substitute schedule for closed days.

schedule information file

A text file containing schedule information parameters, entered by command when setting fixed execution registration for a jobnet.

schedule rule

Jobnet information such as execution start time and processing cycle. Up to 144 schedule rules can be defined for a single jobnet.

scheduler service

A service that manages the schedules for jobnet execution, and executes processes according to those schedules. Each scheduler service manages all the units in the root job group whose name matches the scheduler service name.

Multiple scheduler services can be activated in a single manager. This allows root job groups to be managed individually. For example, if you start a separate scheduler service for each application, each scheduler service can run its specific application (jobnet and jobs) in parallel, independently of the other scheduler services.

shift days

A set number of days within which to determine a substitute date when the next execution date falls on a closed day.

shutdown status

A situation in which a jobnet fails to start or end due to an error, and the execution status or the next scheduled execution cannot be verified. If this happens, you must cancel and then re-register the jobnet for execution.

SMTP (Simple Mail Transfer Protocol)

A protocol, generally used in UNIX networks, for transferring ASCII data by TCP/IP between heterogeneous systems.

standard configuration

A system configuration in which JP1/AJS3 information is managed by the embedded database.

Unless otherwise indicated, the descriptions in this manual relate to a system in a standard configuration.

Note that the ISAM database is still used to store some information related to QUEUE jobs and submit jobs.

start condition

A definition of the conditions under which a jobnet starts when the jobnet is driven by a specific event.

subject

A character string written in the subject line of an email message. Non-ASCII characters are supported in JP1/AJS3, but might not be supported in connected email systems.

submit

To request the system to execute a job.

submitted job

A standard job registered using the `jpqjobs` command.

substitute schedule

A means of executing a jobnet on a different day when the next execution date, determined from the jobnet schedule, falls on a closed day.

succeeding job

A job executed immediately after another job or jobnet.

succeeding jobnet

A jobnet executed immediately after another job or jobnet.

suspend

To suppress the execution of the root jobnet and lower units.

When you change a definition under a root jobnet that has been registered for execution, you should suspend the root jobnet to prevent erroneous operation such as the execution control processing running with the old definition. By suspending the root jobnet, the redefinition processing can be synchronized with the execution control processing.

threshold

A value for evaluating the termination status of a job. You can define an abnormal threshold and a warning threshold for each job.

timeout period

A time interval after which an executed job is forcibly terminated if there was no response from the job or if it failed to complete during the specified period.

TP1/Server Base

Software for distributing transaction processing and server processing in an open

system. JP1/AJS2 uses TP1/Server Base transaction processing.

unit

A generic term for any job network element.

unit definition parameter file

A text file containing unit definition parameters, entered by command when defining the units.

unit ID

A unique number allocated to a unit.

warning threshold

A value for evaluating whether a job ended with a warning.

Windows Messaging

A facility that provides an interface for sending and receiving email. Using Windows Messaging, you can manage, access, and share a variety of information such as data received from an online service.

Index

Numerics

24-hour schedule 63
48-hour schedule 63, 71

A

abbreviations defined iv
abnormal end 426
abnormal threshold 426
absolute day 67
absolute time 68
access permission 342
access permission to executable files and script files 245
action 14
action job 42, 426
agent 399
 configuration 398
agent host 5, 426
agent management control 13
AJS3 unit monitored object 426
 statuses 331
AJSPATH 426
AND condition 87
authentication server 340
automatic distribution 358
automation procedures 28

B

backup box 426
backup file 427
base day 59, 427
base time 60, 93, 427
Being held 261
Benefits of deploying JP1/AJS3 9
Bypassed 261

C

calendar information 427

changing the execution schedule for nested jobnet 173
checking suspension status 213
closed day 68, 427
cluster system 427
commands
 specifying generation 197
common user profile 427
concurrent execution 83, 236
concurrent execution of execution generations 95
conventions
 abbreviations iv
 diagrams viii
 fonts and symbols ix
 KB, MB, GB, and TB xii
 mathematical expressions xi
 meaning of folder and directory xii
 meaning of member of Administrators group xii
 version numbers xii
correlation ID 428
custom job 43, 428

D

Daily Schedule window 428
 checking schedules 158
 monitoring 298
database administrator 428
default queue 428
defaults
 installation folders of JP1/AJS3 for Windows xiii
Defining a start condition 86
delay monitor setting
 temporarily changing 184
dependent job 37, 428
dependent jobnet 428
diagram conventions viii
directory

term defined xii
displaying background image in map area 315
distributing jobs among execution hosts (agent hosts) 253, 402
dummy schedule 159, 161

E

embedded database
 administrator, glossary definition 428
 glossary definition 428
 operation commands, glossary definition 429
 service, glossary definition 429
 system administrator, glossary definition 429
embedded database control 14
End delay 266
end delay monitoring 35
end with warning 429
Ended abnormally 262
Ended normally 262
Ended with warning 262
environment setting parameters
 glossary definition 429
environment settings file 429
ERP systems 354
event 14, 429
event job 40, 429
event/action control 14
examples of job flow 54
exclusive control 404
Exclusive schedule 80
execution agent 233, 429
 default 239
execution agent group 238, 430
execution ID 142, 430
execution-locked resource 404, 430
execution order control 13
execution registration numbers 200
execution schedule
 adding to jobnet 172
execution schedule management 13
execution service 34
execution simulation 164
execution start time
 temporarily changing 173

F

Failed to start 265
features 2
file access permission check (Unix jobs) 245
file transfer 356
fixed execution 130
fixed execution registration 430
fixed schedule 430
fixed schedule period 130
folder
 term defined xii
font conventions ix
Functions 10
future generation 130

G

GB meaning xii
generation 45, 142
generation management
 example of jobnet with start condition 148
 example of jobnet with start condition (upgrade installation from version 8 or earlier) 150
 examples 146
generation number 142
Glossary 426

H

hold 34
HP NNM 356, 430
 term defined xvii
HP NonStop Server 354
HP-UX OS user groups 247

I

immediate execution 126
immediate execution registration 430
installation folders
 default for JP1/AJS3 for Windows xiii
Interrupted 264
Interrupted monitoring 266
interval control job 40
ISAM

- compatible ISAM configuration, glossary definition 427
 - database, glossary definition 430
- J**
- job 3, 34, 430
 - changing status 196
 - distributing among execution hosts (agent hosts) 253, 402
 - killing 255
 - status levels 260
 - status monitoring 249
 - job execution
 - file access permission check (Unix jobs) 245
 - system 232
 - job execution control 14
 - stopping JP1/AJS3 services 256
 - job execution environment 231, 430
 - system operation 244
 - job execution environment settings file
 - specification 248
 - using 248
 - job group 49, 431
 - job network element 431
 - job network management hierarchy 33
 - job transfer
 - restricting 242
 - jobnet 4, 44
 - adding execution schedule 172
 - canceling registration 170
 - determining generation automatically 197
 - executing immediately 177
 - interrupting 187
 - killing 187
 - operations 169
 - prohibiting execution 182
 - releasing 202
 - rerunning 188
 - suspending 205
 - temporarily changing delay monitor settings 184
 - temporarily changing hold attribute 184
 - temporarily changing priority 185
 - jobnet connector 48, 431
 - jobnet creation examples 51
 - Jobnet Editor window 431
 - Jobnet Monitor window 431
 - monitoring 296
 - jobnet monitoring 47
 - jobnet with start conditions
 - suppressing further executions after abnormal termination 104
 - jobnets
 - glossary definition 431
 - JP1 event 431
 - JP1 permission level 431, 342
 - JP1 resource group 342, 431
 - JP1 user 432
 - JP1 users 341
 - JP1/AJS2 for Oracle E-Business Suite 354, 432
 - JP1/AJS3 - Agent 16
 - JP1/AJS3 - Definition Assistant 355, 432
 - JP1/AJS3 - Manager 16
 - database 18
 - JP1/AJS3 - View 16
 - displaying hold plan 269
 - monitoring applications 260
 - priorities of status level 277
 - JP1/AJS3 - View windows
 - monitoring 270
 - JP1/AJS3 - Web Operation Assistant 355
 - JP1/AJS3 Console
 - components of JP1/AJS3 Console 308
 - displaying hold plan 328
 - flow of jobnet status notifications 329
 - generations to monitor 316
 - manipulating monitored jobnets 336
 - monitoring applications 302
 - monitoring interval 329
 - monitoring method 315
 - priority of statuses 319
 - starting JP1/AJS3 - View 337
 - status display priority 334
 - JP1/AJS3 Console Agent 432
 - JP1/AJS3 Console control 15
 - JP1/AJS3 Console Manager 432
 - JP1/AJS3 Console View 432
 - JP1/AJS3 for Enterprise Applications 354, 432

Index

JP1/Base 17, 433
 user authentication function 340
JP1/Base user mapping 344
JP1/FTP 356, 433
JP1/IM 357, 433
JP1/NQSEXEC 357, 433
JP1/OJE 358
JP1/OJE for Midrange Computer 433
JP1/OJE for VOS3 433
JP1/Script 358, 433
JP1/Software Distribution 358, 434
judgment condition 38
judgment job 37, 434
judgment value 434

K

KB meaning xii
kill 434
Killed 265
killed processes (UNIX execution agent host) 255
killing job 255
 created by JP1/Script 255
killing job or jobnet 187

L

list file 434
local power control job 42
logical host 434
login shell
 precautions 247
login shell at job startup (Unix jobs and action jobs) 247

M

macro variable 434
 specifying value during registration for execution 134
mail filtering application 434
mail receipt parameter file 435
mail send parameter file 435
mail system linkage 359
mainframe 358
manager host 5, 435

manager job group 435
manager jobnet 435
manager/agent configuration 5
MAPI 435
mathematical expression conventions xi
max. shiftable days 73, 78, 435
maximum number of concurrently executable jobs 435
maximum shiftable days 73
MB meaning xii
member of the Administrators group
 term defined xii
message ID 435
message queue system linkage 359
MIME 436
monitor delay 76
Monitor terminated 266
Monitor-end normal 266
monitoring event log job 40
monitoring files job 40
monitoring log files job 40
monitoring properties 315
Monthly Schedule window 436
 checking schedules 159
 monitoring 298
multi-platform support 8
multi-schedule 84

N

nested jobnet 44, 436
 time format 64
Nested jobnet delayed end 267
Nested jobnet delayed start 267
NNM linkage
 supported products xvii
node switching system 436
normal end 436
Normal end + False 262
Not executed + Ended (not scheduled) 261
Not sched. to exe. 260
Now monitoring 266
Now queuing 261
Now running 261
Number of logs to keep 45

number of logs to keep 145

O

online manual
 contents xiii
 open day 68, 436
 OpenView Status Report job 42
 OR condition 87
 OR job 37
 Oracle E-Business Suite systems 354
 OS user environment for job execution 244

P

passing information by event job type 410
 PC job 36
 physical host 436
 planned execution 126
 planned execution registration 436
 planning group 49, 362, 436
 definition and execution 372
 preceding job 34, 436
 preceding jobnet 436
 priority (jobnet) 46
 processing cycle 71, 437
 prohibiting job or jobnet execution 182

Q

queue 398, 437
 processing 399
 QUEUE job 36, 397
 QUEUE jobs and submit jobs
 execution flow 400
 queue/agent configuration 397
 queueless job 405, 437
 queueless job execution control 15
 queueless job execution environment 437
 queuing job 437

R

receive JP1 event job 40
 receive mail job 40
 receive MQ message job 40
 receive MSMQ message job 40

recovery job 437
 recovery jobnet 437
 refer to a calendar of another job group 80
 registered day 67
 registering JP1 users 341
 registration 124
 regular expressions available in JP1/AJS3 xvii
 relative day 67
 relative time 68
 release 203
 release entry 203
 release ID 203
 release source jobnet 203
 release target jobnet 203
 releasing suspension 213
 remote power control job 42
 rerunning job or jobnet 188
 retain 100
 revisions
 07-00 423
 07-50 420
 08-00 419
 09-00 416
 root jobnet 44
 Running + Abend 261
 Running + Warning 261

S

SAP BW systems 354
 SAP R/3 systems 354
 schedule by days from start 77, 438
 schedule information file 438
 schedule option 83
 schedule rule 438
 defining 65
 schedule rules 63
 schedule setting options 79
 schedule simulation 161
 schedule skip 83
 scheduled generation 142
 scheduler control 13
 scheduler service 33, 438
 scheduling after releasing suspension 222
 send JP1 event job 42

Index

send mail job 42
send MQ message job 42
send MSMQ message job 42
Settings for restricting access to units 347
shift days 438
Shutdown 266
shutdown status 438
Skipped so not exe. 262
SMTP 438
specific date and time 131
standard configuration 438
standard error output file 248
standard job 36
standard output file 248
Start condition 75
start condition 439
start day options 67
Start delay 267
start time, setting 66
Statistical information 166
status levels of jobnets, jobs, and jobnet
connectors 260
status monitoring of registered jobs 249
statuses for business scopes 334
subject 439
submit 397, 439
submit job 397
submit job control 15
submitted job 439
substitute schedule 439
Substitute schedule of closed day job 72
succeeding job 34, 439
succeeding jobnet 439
suspend 205, 439
suspend function
 activating 212
 executing 212
symbol conventions ix
syntax conventions x
system management control 13

T

TB meaning xii
TELstaff 359

temporarily changing job or jobnet hold attribute 184
threshold 439
timeout period 35, 47, 439
TP1/Server Base 439
type 34

U

umask value for standard output file and standard error
output file (UNIX only) 248
unit 440
unit definition parameter file 440
unit ID 440
Unix job 36
Unknown end status 265
Unmonitored + Ended 266
user account for job execution 244
user name 348
User who owns 348
User who registered 348

V

valid range of start condition 90
version number conventions xii
version revisions 416
View control 15

W

Wait for prev. to end 260
Wait for start cond. 266
Wait for start time 260
Waiting to execute 261
warning threshold 440
Windows Messaging 440

Reader's Comment Form

We would appreciate your comments and suggestions on this manual. We will use these comments to improve our manuals. When you send a comment or suggestion, please include the manual name and manual number. You can send your comments by any of the following methods:

- Send email to your local Hitachi representative.
- Send email to the following address:
WWW-mk@itg.hitachi.co.jp
- If you do not have access to email, please fill out the following information and submit this form to your Hitachi representative:

Manual name:	
Manual number:	
Your name:	
Company or organization:	
Street address:	
Comment:	

(For Hitachi use)
