

**JP1 Version 11**

**JP1/Automatic Job Management System 3  
Overview**

**3021-3-B12-20(E)**

## Notices

### ■ Relevant program products

For details about the applicable OS versions, and the service packs and patches required for JP1/Automatic Job Management System 3, see the *Release Notes*.

*JP1/Automatic Job Management System 3 - Manager (For Windows):*

P-2A12-3KBL JP1/Automatic Job Management System 3 - Manager version 11-50

The above product includes the following:

P-CC2A12-4KBL JP1/Automatic Job Management System 3 - Manager version 11-50 (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2)

P-CC2912-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2)

P-CC8212-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7)

P-CC2A12-3NBL JP1/Automatic Job Management System 3 - Print Option Manager version 11-10 (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2)

*JP1/Automatic Job Management System 3 - Manager (For HP-UX):*

P-1J12-3KBL JP1/Automatic Job Management System 3 - Manager version 11-50

The above product includes the following:

P-CC1J12-4KBL JP1/Automatic Job Management System 3 - Manager version 11-50 (For HP-UX(IPF))

P-CC2912-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2)

P-CC8212-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7)

P-CC1J12-3NBL JP1/Automatic Job Management System 3 - Print Option Manager version 11-10 (For HP-UX(IPF))

*JP1/Automatic Job Management System 3 - Manager (For Solaris):*

P-9D12-3KBL JP1/Automatic Job Management System 3 - Manager version 11-50

The above product includes the following:

P-CC9D12-4KBL JP1/Automatic Job Management System 3 - Manager version 11-50 (For Solaris 11(SPARC), Solaris 10(SPARC))

P-CC2912-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2)

P-CC8212-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7)

P-CC9D12-3NBL JP1/Automatic Job Management System 3 - Print Option Manager version 11-10 (For Solaris 11(SPARC), Solaris 10(SPARC))

*JP1/Automatic Job Management System 3 - Manager (For AIX):*

P-1M12-3KBL JP1/Automatic Job Management System 3 - Manager version 11-50

The above product includes the following:

P-CC1M12-4KBL JP1/Automatic Job Management System 3 - Manager version 11-50 (For AIX)  
P-CC2912-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2)  
P-CC8212-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7)  
P-CC1M12-3NBL JP1/Automatic Job Management System 3 - Print Option Manager version 11-10 (For AIX)

*JP1/Automatic Job Management System 3 - Manager (For Linux):*

P-8112-3KBL JP1/Automatic Job Management System 3 - Manager version 11-50

The above product includes the following:

P-CC8112-4KBL JP1/Automatic Job Management System 3 - Manager version 11-50 (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7)  
P-CC2912-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2)  
P-CC8212-39BL JP1/Automatic Job Management System 3 - Web Console version 11-50 (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7)  
P-CC8112-3NBL JP1/Automatic Job Management System 3 - Print Option Manager version 11-10 (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7)

*JP1/Automatic Job Management System 3 - Agent (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2):*

P-2A12-33BL JP1/Automatic Job Management System 3 - Agent version 11-50

*JP1/Automatic Job Management System 3 - Agent (For HP-UX(IPF)):*

P-1J12-33BL JP1/Automatic Job Management System 3 - Agent version 11-50

*JP1/Automatic Job Management System 3 - Agent (For Solaris 11 (SPARC) and Solaris 10 (SPARC)):*

P-9D12-33BL JP1/Automatic Job Management System 3 - Agent version 11-50

*JP1/Automatic Job Management System 3 - Agent (For AIX):*

P-1M12-33BL JP1/Automatic Job Management System 3 - Agent version 11-50

*JP1/Automatic Job Management System 3 - Agent (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7):*

P-8112-33BL JP1/Automatic Job Management System 3 - Agent version 11-50

*JP1/Automatic Job Management System 3 - Agent Minimal Edition (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2):*

P-2A12-38BL JP1/Automatic Job Management System 3 - Agent Minimal Edition version 11-50

*JP1/Automatic Job Management System 3 - Agent Minimal Edition (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7):*

P-8112-38BL JP1/Automatic Job Management System 3 - Agent Minimal Edition version 11-50

*JP1/Automatic Job Management System 3 - View (For Windows):*

P-2A12-34BL JP1/Automatic Job Management System 3 - View version 11-50

The above product includes the following:

P-CC2A12-44BL JP1/Automatic Job Management System 3 - View version 11-10 (For Windows Server 2016, Windows 10, Windows 8.1, Windows 8, Windows Server 2012 R2, Windows Server 2012, Windows 7, Windows Server 2008 R2)

P-CC2A12-3MBL JP1/Automatic Job Management System 3 - Print Option version 11-50 (For Windows Server 2016, Windows 10, Windows 8.1, Windows 8, Windows Server 2012 R2, Windows Server 2012, Windows 7, Windows Server 2008 R2)

*JP1/Base (For Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2):*

P-2A2C-6LBL JP1/Base version 11-50

*JP1/Base (For HP-UX(IPF)):*

P-1J2C-6LBL JP1/Base version 11-50

*JP1/Base (For Solaris 11 (SPARC) and Solaris 10 (SPARC)):*

P-9D2C-6LBL JP1/Base version 11-50

*JP1/Base (For AIX):*

P-1M2C-6LBL JP1/Base version 11-50

*JP1/Base (For Linux 6.1(x64) or later, Linux 7.1, Oracle Linux 6(x64), Oracle Linux 7, SUSE Linux 12, CentOS 6(x64), CentOS 7):*

P-812C-6LBL JP1/Base version 11-50

## ■ Trademarks

HITACHI, JP1, Job Management Partner 1, uCosminexus, HiRDB are either trademarks or registered trademarks of Hitachi, Ltd. in Japan and other countries.

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

IBM, AIX are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

IBM, AS/400 are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

IBM, MQSeries are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

IBM, MVS are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

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.

Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries. Microsoft Exchange server is a product name of Microsoft Corporation in the U.S. and other countries.

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

Microsoft Office and Outlook are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

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

Microsoft and Outlook are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Microsoft, Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

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

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

RSA and BSAFE are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries.

SAP and R/3 and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc., in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

SUSE is a registered trademark or a trademark of SUSE LLC in the United States and 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.

Other company and product names mentioned in this document may be the trademarks of their respective owners.

The following program products contain some parts whose copyrights are reserved by Oracle and/or its affiliates: P-9D12-3KBL, P-9D12-33BL, and P-9D2C-6LBL.

The following program products contain some parts whose copyrights are reserved by UNIX System Laboratories, Inc.: P-9D12-3KBL, P-9D12-33BL, and P-9D2C-6LBL.

This product includes RSA BSAFE Cryptographic software of EMC Corporation.

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

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.

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

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

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

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

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

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

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

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

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

- 2. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)
- 3. This product includes software written by Tim Hudson (tjh@cryptsoft.com)
- 4. This product includes the OpenSSL Toolkit software used under OpenSSL License and Original SSLeay License. OpenSSL License and Original SSLeay License are as follow:

LICENSE ISSUES

=====

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

```
-----
/* =====
* Copyright (c) 1998-2016 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
*/
```

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

\* =====

- \*
- \* This product includes cryptographic software written by Eric Young
- \* (eay@cryptsoft.com). This product includes software written by Tim
- \* Hudson (tjh@cryptsoft.com).

- \*
- \*/

Original SSLeay License

-----

- /\* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
- \* All rights reserved.
- \*
- \* This package is an SSL implementation written
- \* by Eric Young (eay@cryptsoft.com).
- \* The implementation was written so as to conform with Netscapes SSL.
- \*
- \* This library is free for commercial and non-commercial use as long as
- \* the following conditions are aheared to. The following conditions
- \* apply to all code found in this distribution, be it the RC4, RSA,
- \* lhash, DES, etc., code; not just the SSL code. The SSL documentation
- \* included with this distribution is covered by the same copyright terms
- \* except that the holder is Tim Hudson (tjh@cryptsoft.com).
- \*
- \* Copyright remains Eric Young's, and as such any Copyright notices in
- \* the code are not to be removed.

- \* If this package is used in a product, Eric Young should be given attribution
- \* as the author of the parts of the library used.
- \* This can be in the form of a textual message at program startup or
- \* in documentation (online or textual) provided with the package.
- \*
- \* Redistribution and use in source and binary forms, with or without
- \* modification, are permitted provided that the following conditions
- \* are met:
- \* 1. Redistributions of source code must retain the copyright
- \* notice, this list of conditions and the following disclaimer.
- \* 2. Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following disclaimer in the
- \* documentation and/or other materials provided with the distribution.
- \* 3. All advertising materials mentioning features or use of this software
- \* must display the following acknowledgement:
- \* "This product includes cryptographic software written by
- \* Eric Young (eay@cryptsoft.com)"
- \* The word 'cryptographic' can be left out if the routines from the library
- \* being used are not cryptographic related :-).
- \* 4. If you include any Windows specific code (or a derivative thereof) from
- \* the apps directory (application code) you must include an acknowledgement:
- \* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
- \*
- \* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
- \* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- \* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- \* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
- \* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- \* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- \* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- \* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- \* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- \* SUCH DAMAGE.
- \*
- \* The licence and distribution terms for any publically available version or
- \* derivative of this code cannot be changed. i.e. this code cannot simply be
- \* copied and put under another distribution licence
- \* [including the GNU Public Licence.]
- \*/

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



This product includes software developed by Andy Clark.

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.



**HITACHI**  
Inspire the Next

 Hitachi, Ltd.



## ■ Microsoft product name abbreviations

This manual uses the following abbreviations for Microsoft product names.

Abbreviation		Full name or meaning
Excel		Microsoft(R) Excel
		Microsoft(R) Office Excel
Exchange Server		Microsoft(R) Exchange 2000 Enterprise Server
		Microsoft(R) Exchange 2000 Server
		Microsoft(R) Exchange Server
IE	Internet Explorer	Microsoft(R) Internet Explorer(R)
		Windows(R) Internet Explorer(R)
Microsoft SQL Server		Microsoft(R) SQL Server
		Microsoft(R) SQL Server Enterprise Edition
MSMQ		Microsoft(R) Message Queue Server
Outlook	Outlook 2007	Microsoft(R) Office Outlook(R) 2007
	Outlook 2010	Microsoft(R) Office Outlook(R) 2010
	Outlook 2013	Microsoft(R) Office Outlook(R) 2013
Outlook Express		Microsoft(R) Outlook(R) Express
Windows 10		Windows(R) 10 Enterprise
		Windows(R) 10 Pro

Abbreviation		Full name or meaning
Windows 7		Microsoft(R) Windows(R) 7 Enterprise
		Microsoft(R) Windows(R) 7 Professional
		Microsoft(R) Windows(R) 7 Ultimate
Windows 8		Windows(R) 8 Enterprise
		Windows(R) 8 Pro
Windows 8.1		Windows(R) 8.1 Enterprise
		Windows(R) 8.1 Pro
Windows Server 2008	Windows Server 2008 R2	Microsoft(R) Windows Server(R) 2008 R2 Datacenter
		Microsoft(R) Windows Server(R) 2008 R2 Enterprise
		Microsoft(R) Windows Server(R) 2008 R2 Standard
Windows Server 2012	Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Datacenter
		Microsoft(R) Windows Server(R) 2012 Standard
	Windows Server 2012 R2	Microsoft(R) Windows Server(R) 2012 R2 Datacenter
		Microsoft(R) Windows Server(R) 2012 R2 Standard
Windows Server 2016		Microsoft(R) Windows Server(R) 2016 Datacenter
		Microsoft(R) Windows Server(R) 2016 Standard

*Windows* is sometimes used generically, referring to Windows Server 2016, Windows 10, Windows 8.1, Windows 8, Windows Server 2012, Windows 7, Windows Server 2008.

## ■ 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.

## ■ Issued

Nov. 2017: 3021-3-B12-20(E)

## ■ Copyright

Copyright (C) 2016, 2017, Hitachi, Ltd.

Copyright (C) 2016, 2017, Hitachi Solutions, Ltd.

## Summary of amendments

The following table lists changes in the manuals (3021-3-B12-20(E), 3021-3-B13-20(E), 3021-3-B14-20(E), 3021-3-B15-20(E), 3021-3-B16-20(E), 3021-3-B17-20(E), 3021-3-B18-20(E), 3021-3-B19-20(E), 3021-3-B20-20(E), 3021-3-B21-20(E)) and product changes related to these manuals.

Changes	Location
<p>The Web GUI now support the following functions:</p> <ul style="list-style-type: none"> <li>Viewing monthly schedules</li> <li>Adding execution schedules</li> <li>Change plan (change time)</li> <li>Change plan (execute immediately)</li> <li>Change plan (execution prohibited)</li> <li>Change plan (release change)</li> <li>Change delay monitor</li> </ul>	<p>Overview <i>6.3, 8.2.1, 8.2.2</i></p> <p>System Design (Configuration) Guide <i>3.2.1(3), 3.2.1(4), 3.4.7</i></p> <p>Configuration Guide <i>3.4.3(4), 3.4.3(5)(n), 3.4.3(6), 13.3.3(4), 13.3.3(5)(m), 13.3.3(6)</i></p> <p>Troubleshooting <i>C.1</i></p> <p>Operator's Guide <i>A.4, A.5, A.6, A.7, A.10, A.19, A.20, A.22, A.23, A.24, A.25, A.26, A.27</i></p> <p>Messages <i>KNAK2023-E, KNAK2427-Q, KNAK2428-Q, KNAK2429-E, KNAK2430-Q</i></p>
<p>The API now supports the function to acquire unit information.</p>	<p>Command Reference <i>6.6.1(1), 6.6.1(4), 7.1.2, 7.1.18, 7.3.2, 7.4.5(7), 7.4.5(8), D.1, D.2, D.3, D.4</i></p>
<p>The maximum size of the Java heap on the Web Console server can now be changed, and the default value was changed from 512 MB to 2,048 MB.</p>	<p>Configuration Guide <i>3.4.2, 3.4.9, 13.3.2, 13.3.9</i></p> <p>Administration Guide <i>2.2.5(1)(a), 2.2.5(1)(b)</i></p> <p>Troubleshooting <i>A.4(1), A.4(2)</i></p>
<p>A function was added to enable you to change the colors indicating unit statuses displayed in the windows and dialog boxes of the Web GUI.</p>	<p>Overview <i>6.1</i></p> <p>System Design (Configuration) Guide <i>4.6.1(3)</i></p> <p>Configuration Guide <i>3.4.2, 3.4.8, 13.3.2, 13.3.8</i></p> <p>Administration Guide <i>2.2.5(1)(a), 2.2.5(1)(b)</i></p> <p>Troubleshooting <i>A.4(1), A.4(2)</i></p> <p>Operator's Guide <i>B.</i></p>
<p>The map area in the Monitor dialog box and the Monitor screen of the Web GUI can now switch between displaying unit names and comments.</p>	<p>Operator's Guide <i>A.10, A.11</i></p>
<p>The following functions were added to improve job throughput when multiple scheduler services are active:</p>	<p>System Design (Configuration) Guide <i>3.1.1(1), 3.1.1(5)(a), 3.4.2, 3.4.2(1), 3.4.2(2)</i></p>

Changes	Location
<ul style="list-style-type: none"> <li>Function to reduce the number of ports that are used during job execution</li> <li>Function to output trace logs for each scheduler service</li> </ul> <p>In addition to the changes to internal processing, the JP1/AJS3 performance processing that runs when multiple root jobnets start at the same time was improved. As a result, job throughput when a single scheduler service is active was also improved.</p>	<p>Configuration Guide  3.5.1(1), 6.1.1(2), 6.1.1(4)(b), 6.1.5, 6.1.12, 7.1.1, 8.2.9(2), 8.2.9(4)(b), 13.4.4(1), 15.1.1(2), 15.1.1(4)(b), 15.1.5, 15.1.13, 16.1.1, 17.2.9(2), 17.2.9(4)(b), 20.4.1, 20.4.2(4), 20.4.2(116), 20.8.1, 20.8.2(4)</p> <p>Troubleshooting  1.2.3(1), 1.2.4(1), 1.2.4(2)</p> <p>Command Reference  3. <i>ajstrsetsz</i></p> <p>Messages  KAVS0788-E, KAVU0244-E, KAVU2229-E, KAVU2231-E</p>
<p>The maximum number of execution agents that can be registered on one manager host can now be increased to 2,048.</p>	<p>Overview  A.2</p> <p>System Design (Configuration) Guide  2.1.1, 2.5.1, B.1, B.5</p> <p>Configuration Guide  6.2.28, 15.2.25, 20.3.1, 20.3.2(16)</p> <p>Command Reference  3. <i>ajsagtadd</i></p> <p>Messages  1.3.7(2), KNAC1521-E</p>
<p>To enable the comments on units to be viewed in JP1/IM windows, commenting on units was added as an extended attribute (specific information) of a JP1 event whose OBJECT_TYPE is JOBNET or JOB.</p>	<p>Configuration Guide  20.4.1, 20.4.2(114), 20.4.2(115)</p> <p>Administration Guide  A.2(3), A.2(4), A.2(5), A.2(6), A.2(7), A.2(8), A.2(9), A.2(10), A.2(11), A.2(14), A.2(15), A.2(16), A.2(17), A.2(18), A.2(19), A.2(20), A.2(21), A.2(23), A.2(24), A.2(25), A.2(26), A.2(27), A.2(28), A.2(29), A.2(30), A.2(31), A.2(32), A.2(33), A.2(34), A.2(35), A.2(36), A.2(37), A.2(44), A.2(45), A.2(46), A.2(47)</p> <p>Messages  1.3.2, 1.4.1, KAVS0917-E</p>
<p>Commands to specify settings of environment setting parameters were integrated into the <code>jaajs_config</code> command.</p>	<p>Overview  10.5.1(3)(a)</p> <p>System Design (Configuration) Guide  7.2.2</p> <p>Configuration Guide  4.2, 4.2.1, 4.2.2, 6.2, 6.2.18(1), 6.2.21(1), 6.2.22(1), 6.2.24(1), 6.4, 6.4.1(1), 6.4.1(3), 6.4.2(1), 6.4.3(1), 13.4.1(1), 14.2, 14.2.1, 14.2.2, 15.2.20(1), 15.2.22(1), 15.4, 15.4.1(1), 15.4.1(3), 20.7.2(5), 20.10.1, 20.10.2(3)</p> <p>Troubleshooting  1.2.4(1), 1.2.4(2)</p> <p>Command Reference  2. <i>jaajs_config</i>, 4. <i>ajsqsetup</i></p>
<p>When the scheduler service is restarted, a job that meets specific conditions now transitions to <i>Wait for prev. to end</i> or to <i>Not executed + Ended</i> status (and not to <i>Unknown end status</i>). This enables you to determine, from the job status</p>	<p>Overview  A.2</p>

Changes	Location
after the restart of the scheduler service, whether a user program has started.	Administration Guide 6.2.1(3)(a), 6.4, A.1 Command Reference 3. ajsstart Messages KAVS0266-I
Descriptions were added concerning JP1/AJS3 - Agent Minimal Edition, which limits the number of jobs that can be executed per day to 10.	System Design (Configuration) Guide 2.1.1, 2.1.2(1), 4.2.1, 8.1.3, A.1(2), A.2, A.4, A.7(1), A.7(2), A.7(3), G. Configuration Guide 20.5.1, 20.5.2(93), 20.5.2(94), 20.6.1, 20.7.1, 20.11.1 Administration Guide 7.9.3 Troubleshooting 1.2.4(1), 1.2.4(2) Command Reference 1.5.1, 1.5.9(2), 2. ajsatsetup, 4. ajsqlsetup Messages 1.3.4, 1.3.8, KAVS1931-E, KAVS4540-E, KAVS8100-E, KAVU3607-I, KAVU3608-E, KNAD1000-E, KNAD1704-Q

**Legend:**

Overview: *JP1/Automatic Job Management System 3 Overview*

System Design (Configuration) Guide: *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*

System Design (Work Tasks) Guide: *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

Configuration Guide: *JP1/Automatic Job Management System 3 Configuration Guide*

Administration Guide: *JP1/Automatic Job Management System 3 Administration Guide*

Troubleshooting: *JP1/Automatic Job Management System 3 Troubleshooting*

Operator's Guide: *JP1/Automatic Job Management System 3 Operator's Guide*

Command Reference: *JP1/Automatic Job Management System 3 Command Reference*

Linkage Guide: *JP1/Automatic Job Management System 3 Linkage Guide*

Messages: *JP1/Automatic Job Management System 3 Messages*

In addition to the above changes, minor editorial corrections were 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

PART 1 describes the features of JP1/AJS3 and its component programs.

### PART 2: Basic Operations

PART 2 provides JP1/AJS3 basics, including a JP1/AJS3 overview, a workflow of using JP1/AJS3 tasks, and an automated tasks workflow.

### PART 3: Functions (Applications)

PART 3 describes the functions for linking JP1/AJS3 with other products, and the JP1/AJS2 functionality that is supported in JP1/AJS3.

## ■ Organization of JP1/AJS3 manuals and choosing the right manuals

There are 11 JP1/AJS3 manuals. The following table summarizes their contents.

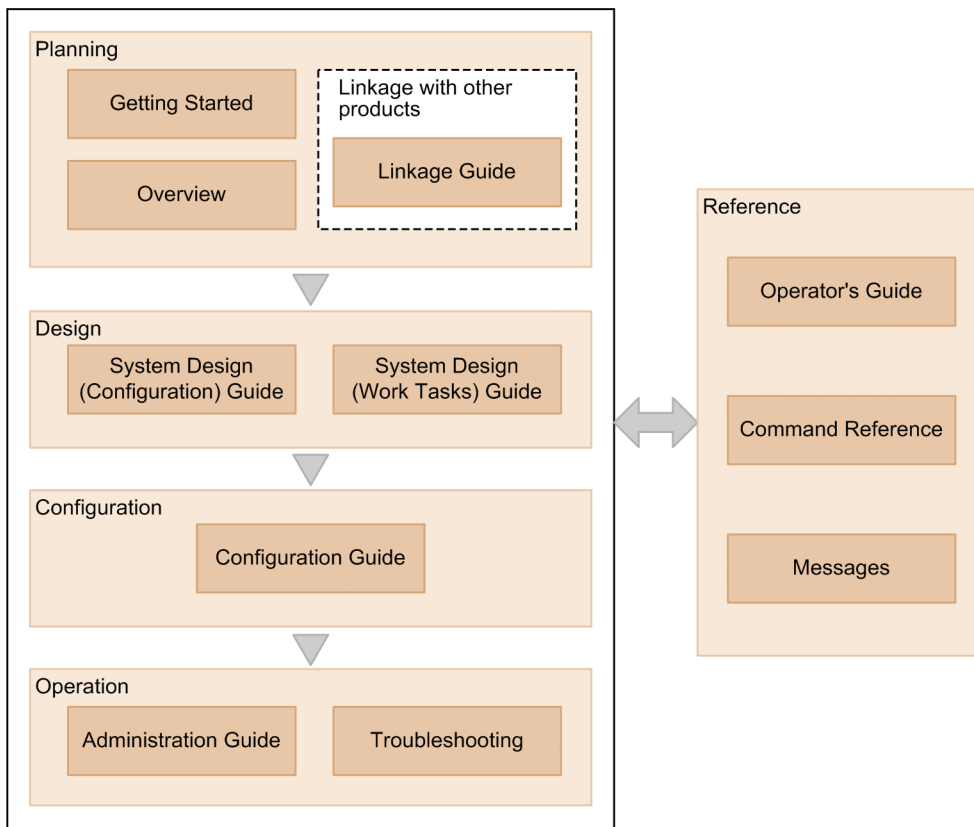
No.	Manual	Contents
1	<i>Getting Started</i> (3021-3-B11(E))	<ul style="list-style-type: none"><li>• Basic JP1/AJS3 operations</li></ul>
2	<i>Overview</i> (3021-3-B12(E))	<ul style="list-style-type: none"><li>• JP1/AJS3 features</li><li>• Description of functions</li></ul>
3	<i>System Design (Configuration) Guide</i> (3021-3-B13(E))	<ul style="list-style-type: none"><li>• Information that must be considered when designing a system</li><li>• Cautionary notes on designing a system</li></ul>
4	<i>System Design (Work Tasks) Guide</i> (3021-3-B14(E))	<ul style="list-style-type: none"><li>• Information that must be considered when constructing jobs and jobnets</li><li>• Cautionary notes on designing jobs and jobnets</li></ul>
5	<i>Configuration Guide</i> (3021-3-B15(E))	<ul style="list-style-type: none"><li>• Installation and setup procedures</li><li>• Environment setup procedure</li><li>• Description of environment setting parameters</li><li>• How to set up functions according to how the system is used</li></ul>

No.	Manual	Contents
6	<i>Administration Guide</i> (3021-3-B16(E))	<ul style="list-style-type: none"> <li>Information required to operate a system</li> <li>Know-how useful for JP1/AJS3 operation</li> </ul>
7	<i>Troubleshooting</i> (3021-3-B17(E))	<ul style="list-style-type: none"> <li>How to troubleshoot errors</li> <li>Data required when an error occurs</li> </ul>
8	<i>Operator's Guide</i> (3021-3-B18(E))	<ul style="list-style-type: none"> <li>How to operate JP1/AJS3 - View</li> <li>Description of windows and dialog boxes</li> </ul>
9	<i>Command Reference</i> (3021-3-B19(E))	<ul style="list-style-type: none"> <li>Command syntax</li> <li>Syntax and coding examples of information definition files</li> <li>API details and syntax</li> </ul>
10	<i>Linkage Guide</i> (3021-3-B20(E))	<ul style="list-style-type: none"> <li>Description of functions that can be used when linked with other products and the setup method</li> </ul>
11	<i>Messages</i> (3021-3-B21(E))	<ul style="list-style-type: none"> <li>Messages output by JP1/AJS3</li> </ul>

*Getting Started* refers to *JP1 Version 11 Job Management: Getting Started (Job Scheduler)*. "JP1 Version 11 JP1/Automatic Job Management System 3" is omitted in other manual names.

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 the basic usage of JP1/AJS3	<ul style="list-style-type: none"> <li>• <i>Getting Started</i> (3021-3-B11(E))</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Operator's Guide</i> (3021-3-B18(E))</li> </ul>
To learn about JP1/AJS3's functionalities	<ul style="list-style-type: none"> <li>• <i>Overview</i> (3021-3-B12(E))</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Linkage Guide</i> (3021-3-B20(E))</li> </ul>
To configure a system (including installation and setup) that automatically runs jobs	<ul style="list-style-type: none"> <li>• <i>System Design (Configuration) Guide</i> (3021-3-B13(E))</li> <li>• <i>Configuration Guide</i> (3021-3-B15(E))</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Linkage Guide</i> (3021-3-B20(E))</li> </ul>
To design work tasks that will be automated (including job definitions and schedule definitions)	<ul style="list-style-type: none"> <li>• <i>System Design (Work Tasks) Guide</i> (3021-3-B14(E))</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Operator's Guide</i> (3021-3-B18(E))</li> </ul>
To learn about monitoring and maintaining a running system	<ul style="list-style-type: none"> <li>• <i>Administration Guide</i> (3021-3-B16(E))</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Troubleshooting</i> (3021-3-B17(E))</li> <li>• <i>Messages</i> (3021-3-B21(E))</li> </ul>
To learn about what action you need to take for problems that occur during operation	<ul style="list-style-type: none"> <li>• <i>Troubleshooting</i> (3021-3-B17(E))</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Messages</i> (3021-3-B21(E))</li> </ul>
To learn about operating JP1/AJS3	<ul style="list-style-type: none"> <li>• <i>Operator's Guide</i> (3021-3-B18(E))</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Command Reference</i> (3021-3-B19(E))</li> </ul>

## ■ Replacing the term JP1/AJS3 - Agent with JP1/AJS3 - Agent Minimal Edition in this manual

This manual uses the term *JP1/AJS3 - Agent* to mean both JP1/AJS3 - Agent and JP1/AJS3 - Agent Minimal Edition. If you use JP1/AJS3 - Agent Minimal Edition, replace the term *JP1/AJS3 - Agent* with the term *JP1/AJS3 - Agent Minimal Edition* while reading this manual. If, however, a difference between JP1/AJS3 - Agent Minimal Edition and JP1/AJS3 - Agent is described, both terms (*JP1/AJS3 - Agent Minimal Edition* and *JP1/AJS3 - Agent*) are used.



# Contents

Notices	2
Summary of amendments	11
Preface	14

## Part 1: Overview

<b>1</b>	<b>Overview of JP1/AJS3</b>	<b>22</b>
1.1	Features of JP1/AJS3	23
1.1.1	Automating an application	23
1.1.2	Centralized management in a manager/agent configuration	25
1.1.3	GUI-based definition and monitoring	25
1.1.4	Support for a wide range of systems	27
1.1.5	Types of applications that can be automated	28
1.1.6	Benefits of deploying JP1/AJS3	29
1.2	Functions provided in JP1/AJS3	30
1.2.1	Overview of JP1/AJS3 functions	30
1.2.2	Function inter-relationships	31
1.2.3	System management control	32
1.2.4	Scheduler control	32
1.2.5	Agent management control	33
1.2.6	Job execution control	33
1.2.7	Event/action control	33
1.2.8	Embedded database control	34
1.2.9	View control	34
1.2.10	Web GUI control	34
1.2.11	API control	34
1.2.12	Other functions	34
1.3	JP1/AJS3 program configuration	36
1.4	JP1/AJS3 database configurations	37
1.4.1	Standard configuration	38
1.4.2	Standard configuration without ISAM	38
1.4.3	Compatible ISAM configuration	39

## Part 2: Basic Operations

<b>2</b>	<b>Automating Applications Using JP1/AJS3</b>	<b>40</b>
2.1	Preparing to automate applications	41

2.1.1	Application assessment	41
2.1.2	Procedures for deploying JP1/AJS3	44
<b>3</b>	<b>Defining Automated Applications</b>	<b>46</b>
3.1	Hierarchical structure of the job network	47
3.1.1	Units	48
3.1.2	Building a jobnet	63
3.1.3	Applications that use macro variables	66
3.1.4	Creating a job flow	66
3.2	Defining a calendar for JP1/AJS3 operation	72
3.2.1	Defining open days and closed days	72
3.2.2	Setting the base day and base time	72
3.3	Defining a schedule	76
3.3.1	Time formats and schedule rules for root jobnets	76
3.3.2	Defining a schedule	78
3.3.3	Concurrent execution and schedule option	93
3.3.4	Checking defined schedules	95
3.4	Defining a start condition	96
3.4.1	Purpose of a start condition	96
3.4.2	Monitoring generation and execution generation	98
3.4.3	Valid range of a start condition	99
3.4.4	Concurrent execution of execution generations	103
3.4.5	Concurrent execution of monitoring generations	105
3.4.6	Retention of execution generations with satisfied start conditions	107
3.4.7	Suppressing further executions of a jobnet with start conditions after it abnormally terminates	110
3.4.8	Operations on a jobnet registered for execution with a start condition	122
3.4.9	Start condition and number of logs to keep	123
<b>4</b>	<b>Executing an Application</b>	<b>125</b>
4.1	Registering a jobnet for execution	126
4.1.1	Methods of registering a jobnet for execution	126
4.1.2	Specifying macro variables during registration for execution	136
4.2	Managing jobnet generations	140
4.2.1	Managing the scheduled generations of a jobnet	140
4.2.2	Jobnet generation number	140
4.2.3	Jobnet generation management	142
4.3	Checking execution registration information	154
4.4	Checking execution schedules	156
4.4.1	Checking a jobnet schedule	156
4.4.2	Simulation	158
4.4.3	Statistical information	162
4.5	Operations on registered jobnets	164

- 4.5.1 Canceling registration of a jobnet 165
- 4.5.2 Adding an execution schedule to a jobnet 166
- 4.5.3 Temporarily changing the execution start time of a jobnet 167
- 4.5.4 Executing a registered jobnet immediately 172
- 4.5.5 Prohibiting execution of a job or jobnet 175
- 4.5.6 Temporarily changing the hold attribute of a job or jobnet 176
- 4.5.7 Temporarily changing the delay monitor settings of a jobnet 177
- 4.5.8 Temporarily changing the priority of a jobnet 178
- 4.5.9 Interrupting a jobnet 179
- 4.5.10 Killing a job or jobnet 179
- 4.5.11 Rerunning a job or jobnet 181
- 4.5.12 Changing the status of a job 188
- 4.5.13 Methods of specifying generations when executing commands 189
- 4.5.14 Automatically switching a jobnet definition at a specified time 193
- 4.5.15 Temporarily changing the wait condition settings for a jobnet or job 195
- 4.5.16 Displaying and re-executing temporary change operations for a job or jobnet 196
- 4.5.17 Changing job and jobnet definitions without unregistering the jobnet 205

## **5 Job Execution Environments 228**

- 5.1 How jobs are executed 229
  - 5.1.1 Execution agents 230
  - 5.1.2 Executing jobs concurrently 233
  - 5.1.3 Execution agent group 234
  - 5.1.4 Default execution agent 236
- 5.2 Restricting job transfer 238
- 5.3 Execution agent restriction function 239
- 5.4 Job execution environments and system operation 240
  - 5.4.1 User account for job execution 240
  - 5.4.2 OS user environment used for job execution 240
  - 5.4.3 Access permission to executable files and script files 240
  - 5.4.4 Checking file access permission at job execution (UNIX only) 241
  - 5.4.5 HP-UX OS user groups 242
  - 5.4.6 Login shell at job startup (Unix jobs, flexible jobs, HTTP connection jobs, and action jobs) 243
  - 5.4.7 Value of umask set for the standard output file and the standard error output file (UNIX only) 243
  - 5.4.8 Monitoring the status of registered jobs 244
  - 5.4.9 Distributing jobs among execution hosts (agent hosts) 247
  - 5.4.10 Killing jobs 248
  - 5.4.11 Stopping JP1/AJS3 services and job execution control 249
  - 5.4.12 Group ID for job execution (UNIX only) 250
  - 5.4.13 Programs that JP1/AJS3 can execute (Windows only) 250

<b>6</b>	<b>Monitoring Applications</b>	<b>251</b>
6.1	Status levels of jobnets, jobs, and jobnet connectors	252
6.2	Using JP1/AJS3 - View to monitor jobs and jobnets	259
6.2.1	Monitoring via the JP1/AJS3 - View windows	259
6.2.2	Monitoring via the Jobnet Monitor window	278
6.2.3	Monitoring via the Daily Schedule window	279
6.2.4	Monitoring via the Monthly Schedule window	279
6.3	Monitoring jobnets by using a web browser	281
6.4	Monitoring jobnets by using user applications	282
6.5	Features of and differences between jobnet monitoring programs	283
<b>7</b>	<b>User Management and Access Control</b>	<b>286</b>
7.1	User management using the functions of JP1/Base	287
7.1.1	User management using the JP1/Base user authentication function	287
7.1.2	User management using the user mapping function of JP1/Base	290
7.1.3	User management using the JP1 administrators group of JP1/Base (UNIX only)	291
7.2	Settings for restricting access to units	294
7.2.1	Unit owner permission	294
7.2.2	Setting the JP1 resource group	294
7.2.3	Setting the Executed by attribute	295
7.2.4	Setting the User name (for PC jobs, Unix jobs, and HTTP connection jobs)	295
7.3	Settings for restricting connection to JP1/AJS3	297
<b>8</b>	<b>Web Interfaces Available in JP1/AJS3</b>	<b>299</b>
8.1	Overview of web interfaces available in JP1/AJS3	300
8.2	Operating JP1/AJS3 by using the Web GUI	301
8.2.1	What you can do with the Web GUI	301
8.2.2	Web GUI screens for monitoring	302
8.2.3	Authentication and operation mechanisms of the Web GUI	306
8.3	Operating JP1/AJS3 by using user applications	308
8.3.1	What you can do with user applications	308
8.3.2	Authentication and operation mechanisms of user applications	309

### Part 3: Functions (Applications)

<b>9</b>	<b>Linking JP1/AJS3 with Other Products</b>	<b>311</b>
9.1	Functions implemented by JP1/AJS3 product linkage	312
<b>10</b>	<b>Supported JP1/AJS2 Functionality</b>	<b>316</b>
10.1	Using a planning group to change the plans for root jobnets	317
10.1.1	Overview of planning group functions and operations	317
10.2	Remote jobnet	337

10.3	Manager jobnet	340
10.4	QUEUE jobs and submit jobs	341
10.4.1	Overview and feature description of QUEUE jobs and submit jobs	341
10.5	Queueless jobs	347
10.5.1	Overview and feature description of queueless jobs	347

## **Appendixes 351**

A	Functions that Cannot Be Used in the Standard Configuration Without ISAM or in the Compatible ISAM Configuration	352
A.1	Functions that cannot be used in the standard configuration without ISAM	352
A.2	Functions that cannot be used in the compatible ISAM configuration	352
B	Version Revisions	354
B.1	Revisions in 11-50	354
B.2	Revisions in 11-10	355
B.3	Revisions in 11-00	356
B.4	Revisions in 10-50	360
B.5	Revisions in 10-10	361
B.6	Revisions in 10-00	362
B.7	Revisions in 09-00	365
C	Reference Material for This Manual	369
C.1	Related publications	369
C.2	Conventions: Abbreviations for product names	370
C.3	Conventions: Acronyms	372
C.4	Conventions: Diagrams	373
C.5	Conventions: Fonts and symbols	374
C.6	Conventions: KB, MB, GB, and TB	376
C.7	Conventions: Meaning of "directory" and "folder"	376
C.8	Conventions: Version numbers	376
C.9	About administrator privileges	376
C.10	About the Start menu in Windows	376
C.11	Default installation folders of JP1/AJS3 for Windows	377
C.12	Online manual	377
C.13	Regular expressions available in JP1/AJS3	377
C.14	About the 3rd level and 4th level Kanji characters added in JIS 2004 (JIS X 0213:2004)	378
C.15	About NNM linkage	378
D	Glossary	379

## **Index 394**

# 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

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.

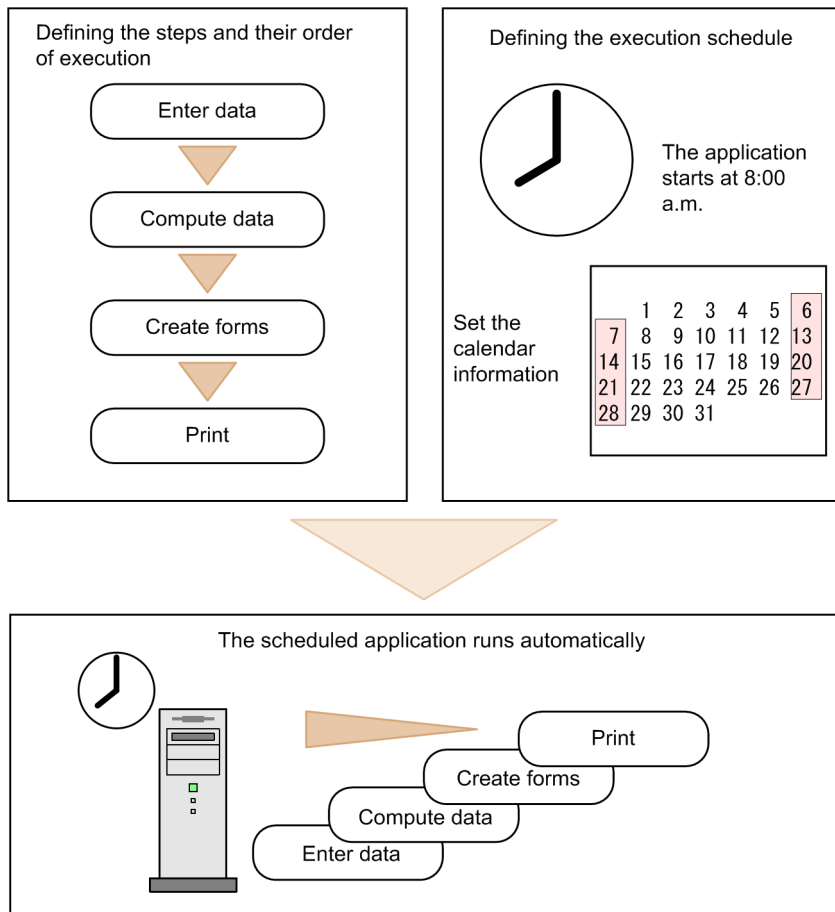
## 1.1.1 Automating an application

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:

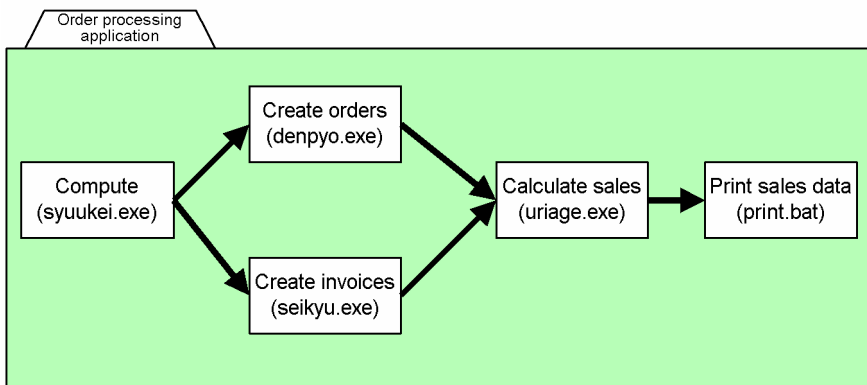
1. Data entry
2. Computation
3. Form creation
4. Printing

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.

JP1/AJS3 provides two GUI functionalities: JP1/AJS3 - View and the Web GUI of JP1/AJS3 - Web Console.

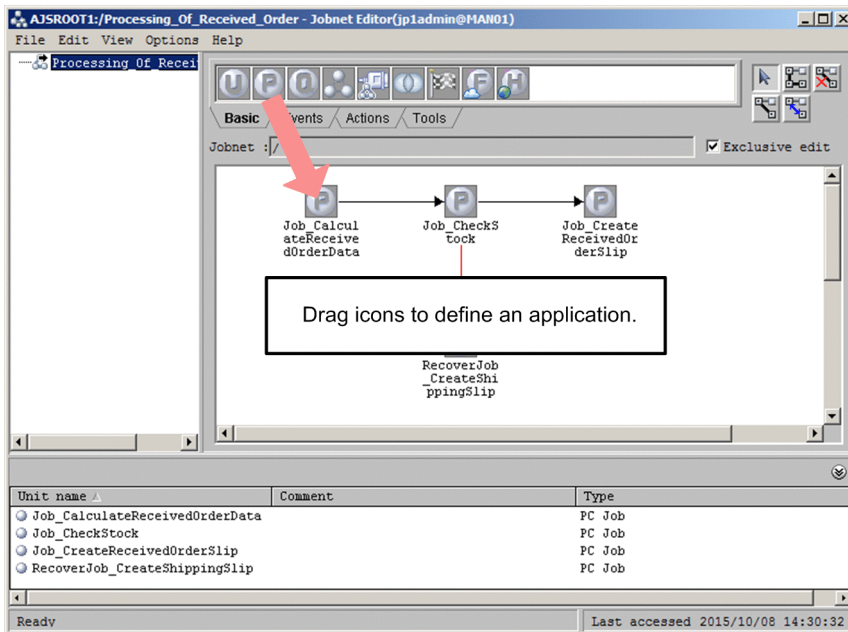
JP1/AJS3 - View is a program that allows users to perform a variety of jobnet-related operations, such as definition, execution, and monitoring.

The Web GUI is a program mainly used to execute and monitor jobnets. You cannot define jobnets or jobs with the Web GUI. To define jobnets or jobs, use JP1/AJS3 - View or JP1/AJS3 - Definition Assistant.

You can define applications by dragging and dropping icons, using the intuitive interface in JP1/AJS3 - View.

The following figure shows an example of defining an application using the JP1/AJS3 - View.

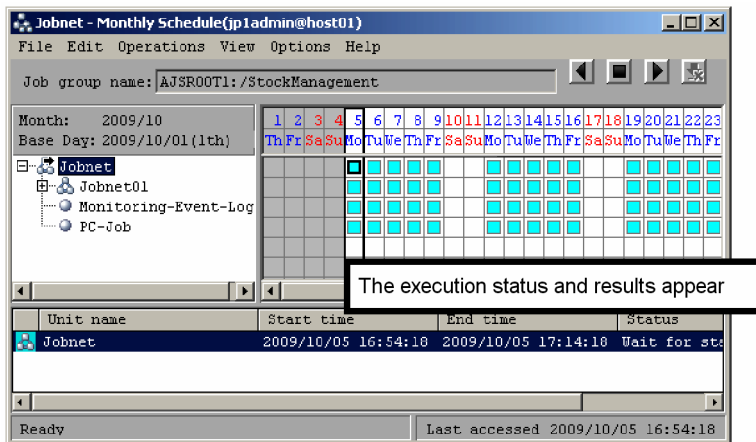
Figure 1–3: Defining an application using the JP1/AJS3 - View



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 JP1/AJS3 - View.

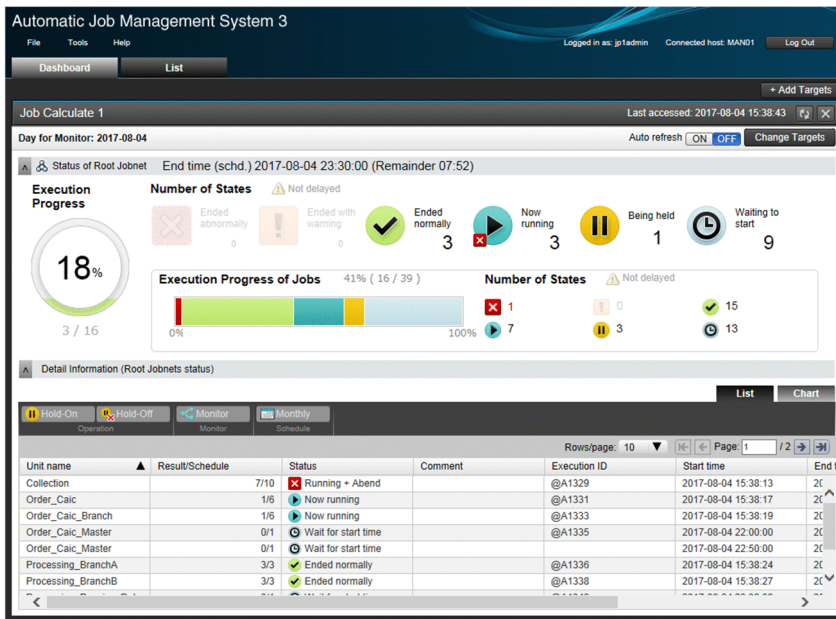
Figure 1–4: Monitoring execution status using the JP1/AJS3 - View



The Web GUI enables you to monitor the progress and execution statuses of applications by using a web browser from an external network. For example, you can check for problems with specific applications while you are away from your office, or monitor JP1/AJS3 applications that are built into a cloud environment.

The following figure shows an example of monitoring applications by using the Web GUI.

Figure 1–5: Example of monitoring applications by using the Web GUI



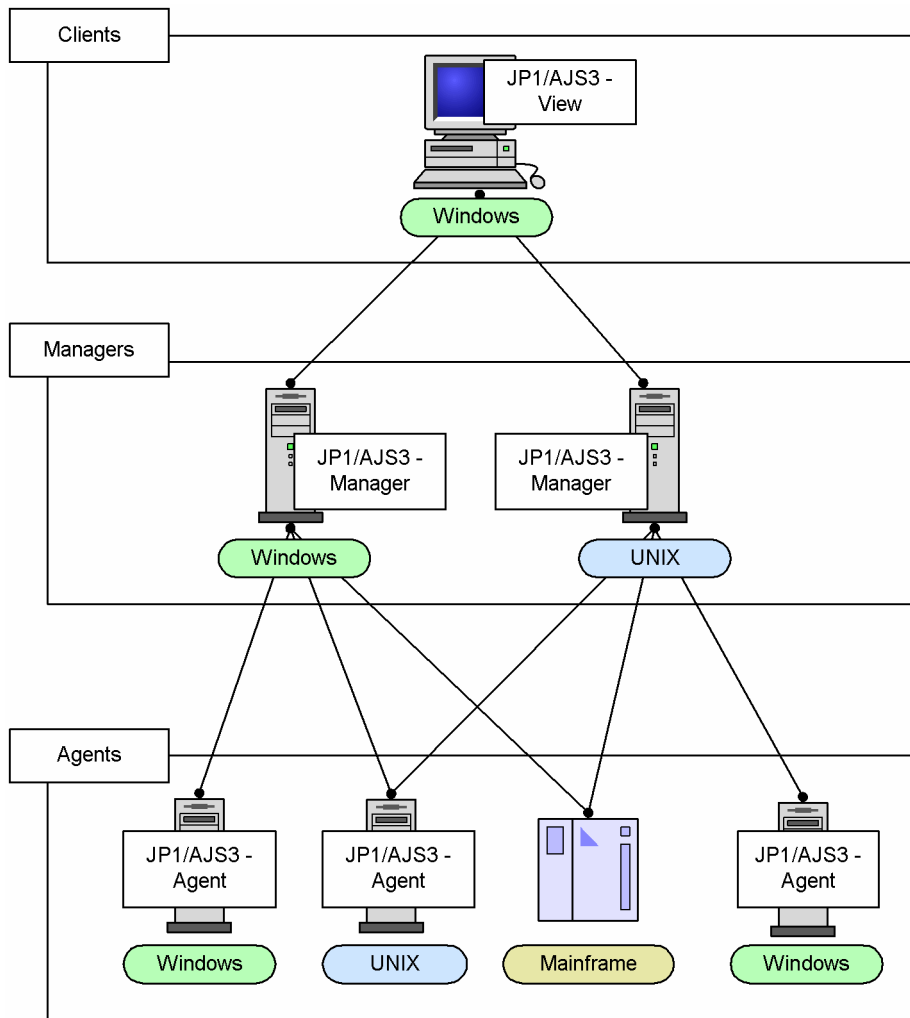
JP1/AJS3 - View and the Web GUI differ from each other in features and role. Understand their differences, and use them appropriately for your purposes. For details about the differences between JP1/AJS3 - View and the Web GUI, see [6.5 Features of and differences between jobnet monitoring programs](#).

### 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. For example, you can build a small-scale system in which only one host is used to define and execute jobs, or you can build a large-scale system in which job execution is distributed across 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

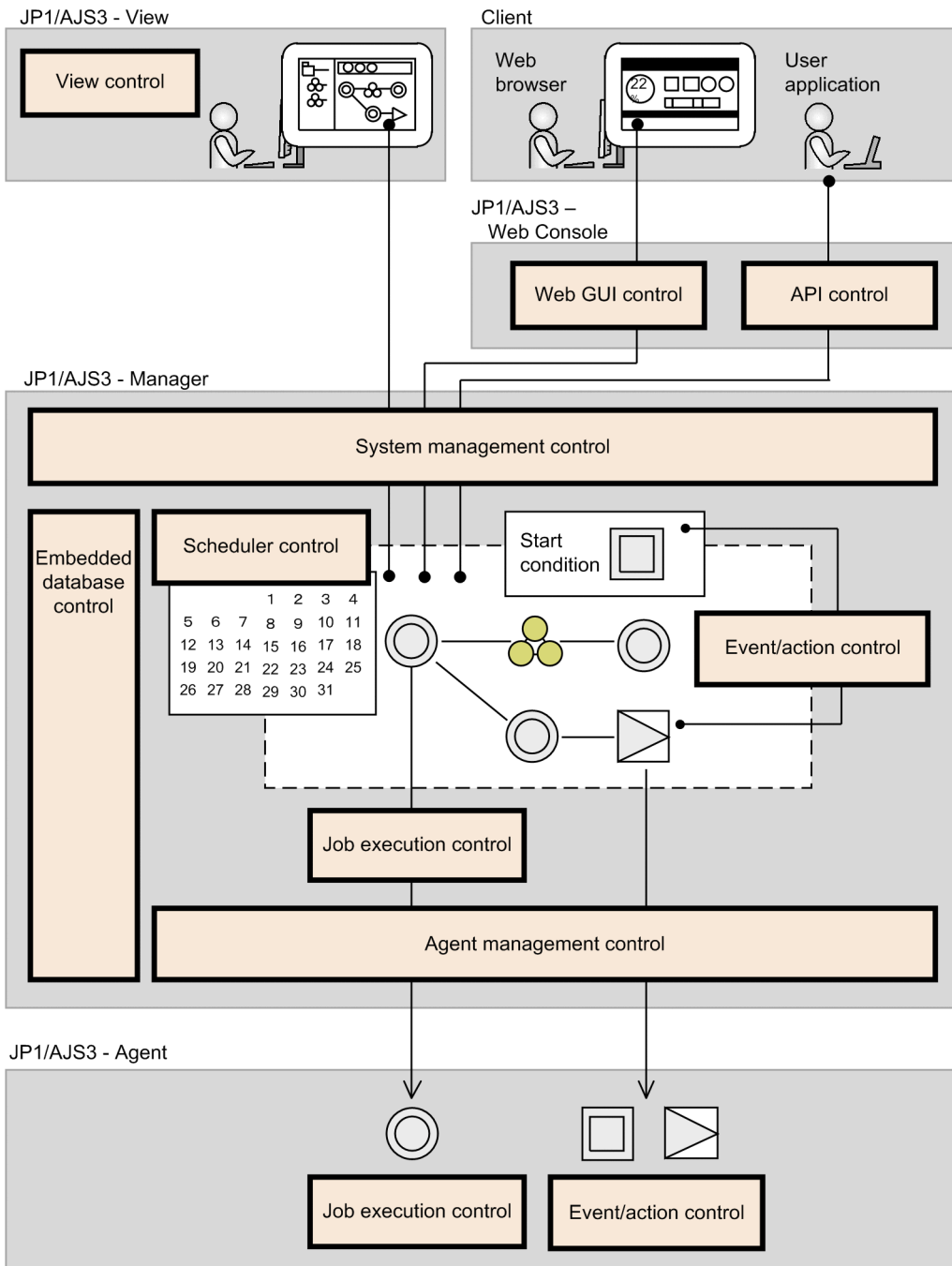
The following table summarizes the programs and functions that constitute JP1/AJS3. JP1/AJS3 uses these functions in combination to control job execution.

Table 1–1: JP1/AJS3 functions

Program	Function	Summary
JP1/AJS3 - Manager	System management control	<ul style="list-style-type: none"> <li>Management of services, processes, and environment settings</li> </ul>
	Scheduler control	<ul style="list-style-type: none"> <li>Execution schedule management</li> <li>Execution sequence control using a jobnet</li> </ul>
	Agent management control	<ul style="list-style-type: none"> <li>Execution agent management</li> </ul>
	Job execution control	<ul style="list-style-type: none"> <li>Job transfer and status management (Manager)</li> <li>Job execution (Agent)</li> </ul>
	Submit job execution control	<ul style="list-style-type: none"> <li>Job management using queuing (Manager)</li> </ul>
	Event/action control	<ul style="list-style-type: none"> <li>Management of event monitoring status (Manager)</li> <li>Event monitoring (Agent)</li> </ul>
	Embedded database control	<ul style="list-style-type: none"> <li>The relational database that stores jobnet definitions, execution agent definitions, and other information</li> </ul>
JP1/AJS3 - Agent	Job execution control	<ul style="list-style-type: none"> <li>Job execution (Agent)</li> </ul>
	Event/action control	<ul style="list-style-type: none"> <li>Event monitoring (Agent)</li> </ul>
JP1/AJS3 - View	View control	<ul style="list-style-type: none"> <li>Job and jobnet definitions</li> <li>Job and jobnet scheduling and execution status display</li> </ul>
JP1/AJS3 - Web Console	Web GUI control	<ul style="list-style-type: none"> <li>Display of job/jobnet execution schedules and execution statuses on a web browser</li> </ul>
	API control	<ul style="list-style-type: none"> <li>Provision of the API to reference and operate jobs and jobnets</li> </ul>

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.

## 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 types of 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 Web GUI control

You can monitor the execution schedules and execution statuses of jobs and jobnets by using a web browser. You can also perform operations such as execution registration, rerun, and temporary change of schedules.

## 1.2.11 API control

JP1/AJS3 provides the HTTP-based API functions to reference and operate jobs and jobnets. With the API functions, you can perform unit information acquisition, execution registration, and other operations from a user application.

## 1.2.12 Other functions

### (1) Submit job execution control

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

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

Submit job execution 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 execution control, is managed in an ISAM database.

For the setup required to use submit job execution control, see *7.1.2 Setup for using QUEUE jobs and submit jobs* in the *JP1/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) Flexible job execution control**

This feature executes jobs called *flexible jobs*. When a flexible job is submitted, the job execution control function on the manager host transfers the job via a relay agent to the destination agent on which the flexible job is to be executed.

## **(4) JP1/AJS3 - Print Option control**

You can use JP1/AJS3 - Print Option to display or print jobnet or schedule information formatted as needed (for example, as jobnet definition information, an execution schedule table, or an execution result confirmation table).

You can also output jobnet or schedule information in CSV format so that you can edit the information.

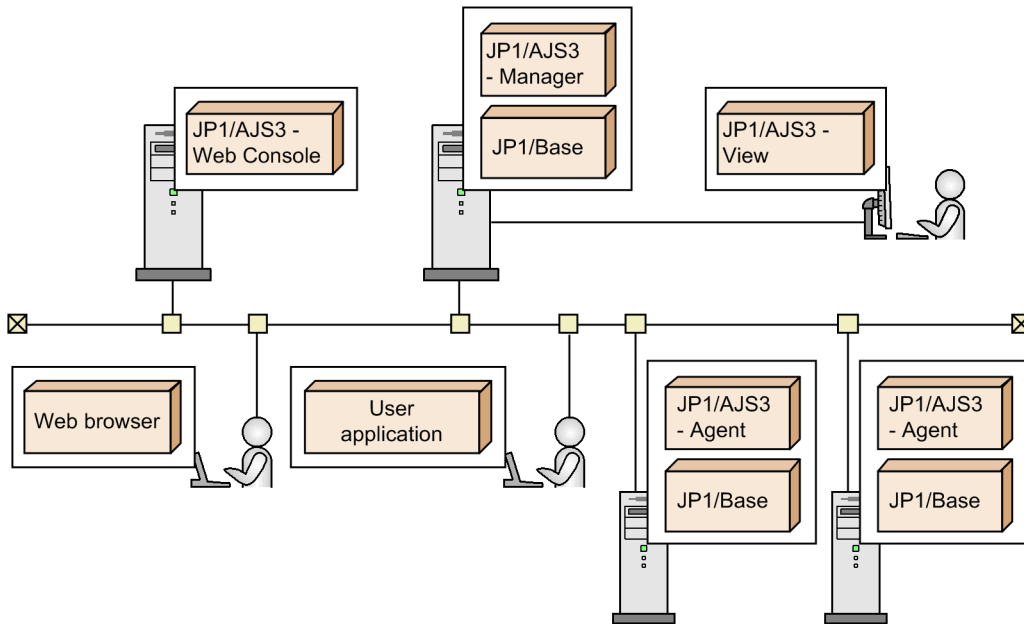
For details on JP1/AJS3 - Print Option, see *JP1/Automatic Job Management System 3 - Print Option Description, User's Guide*.

## 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/AJS3 - Web Console

JP1/AJS3 - Web Console provides the Web GUI and an API. By request from a client, JP1/AJS3 - Web Console connects to JP1/AJS3 - Manager, acquires or manipulates information, and then sends a response to the client.

### 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.

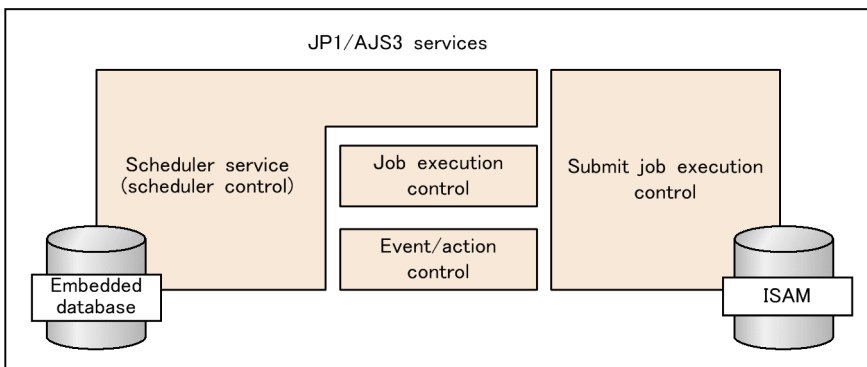
JP1/AJS3 - Manager supports two types of database configurations:

- Standard configuration
- Standard configuration without ISAM

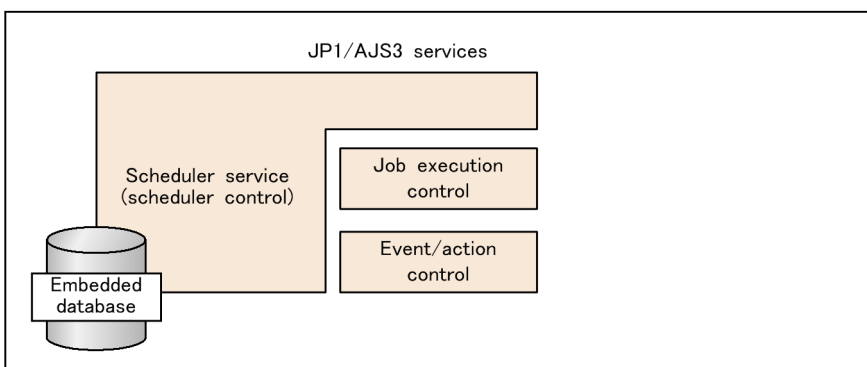
The following figure shows the database configurations supported by JP1/AJS3 - Manager.

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

■ Standard configuration



■ Standard configuration without ISAM



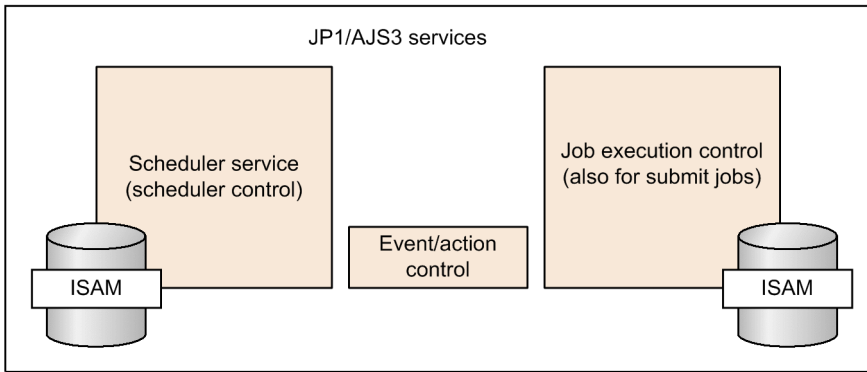
### Supplementary note

In version 11-00 or earlier, a configuration that uses only an ISAM database (compatible ISAM configuration) can be used to maintain compatibility with JP1/AJS2 version 8 or earlier.

In version 11-10 or later, however, the compatible ISAM configuration is not supported.

Figure 1–10: Compatible ISAM configuration

■ Compatible ISAM configuration



Each database configuration is described next.

## 1.4.1 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 Upgrading JP1/AJS2 - Manager to JP1/AJS3 - Manager* in the *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

## 1.4.2 Standard configuration without ISAM

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

In this configuration, some of the functions that can be used in the standard configuration (with ISAM), such as the submit job execution control function, cannot be used. For details, see *A.1 Functions that cannot be used in the standard configuration without ISAM*.

For the setup required to change from a standard configuration to a standard configuration without ISAM, see *6.6.2 Changing the JP1/AJS3 database configuration to the standard configuration without ISAM* in the *JP1/Automatic Job Management System 3 Configuration Guide (Windows)* or see *15.7.2 Changing the JP1/AJS3 database configuration to the standard configuration without ISAM* in the *JP1/Automatic Job Management System 3 Configuration Guide (UNIX)*.

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

### 1.4.3 Compatible ISAM configuration

The compatible ISAM configuration supports the previous JP1/AJS2 - Manager. Note that this configuration is not supported in version 11-10 or later.

In this configuration, some of the functions that can be used in the standard configuration (with or without ISAM) cannot be used. For details, see *A.2 Functions that cannot be used in the compatible ISAM 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. In this case, change the database configuration to the standard configuration. For the procedure to change the database configuration, see *8.5 Upgrading JP1/AJS2 - Manager to JP1/AJS3 - Manager in the JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

The descriptions in this manual assume a standard database configuration unless otherwise stated.

# 2

## Automating Applications Using JP1/AJS3

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



## 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

##### Open days and closed 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 is 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

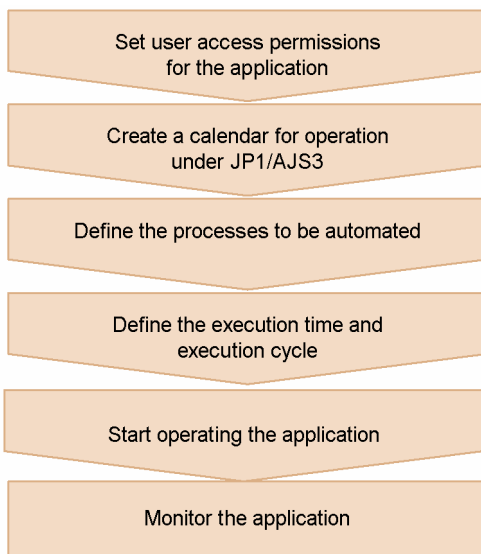
- The work flow controller in the Accounting Department is allowed to define and modify departmental jobs, but not to execute them. The controller has no authority to operate on jobs related to the Personnel Department.
- Operators in the Accounting Department are allowed to execute departmental jobs,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

# 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

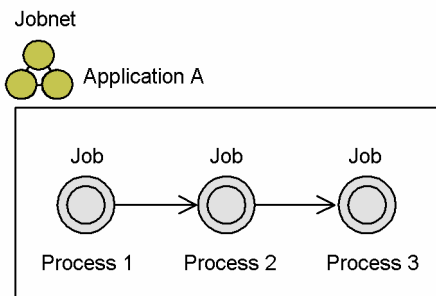
---

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.

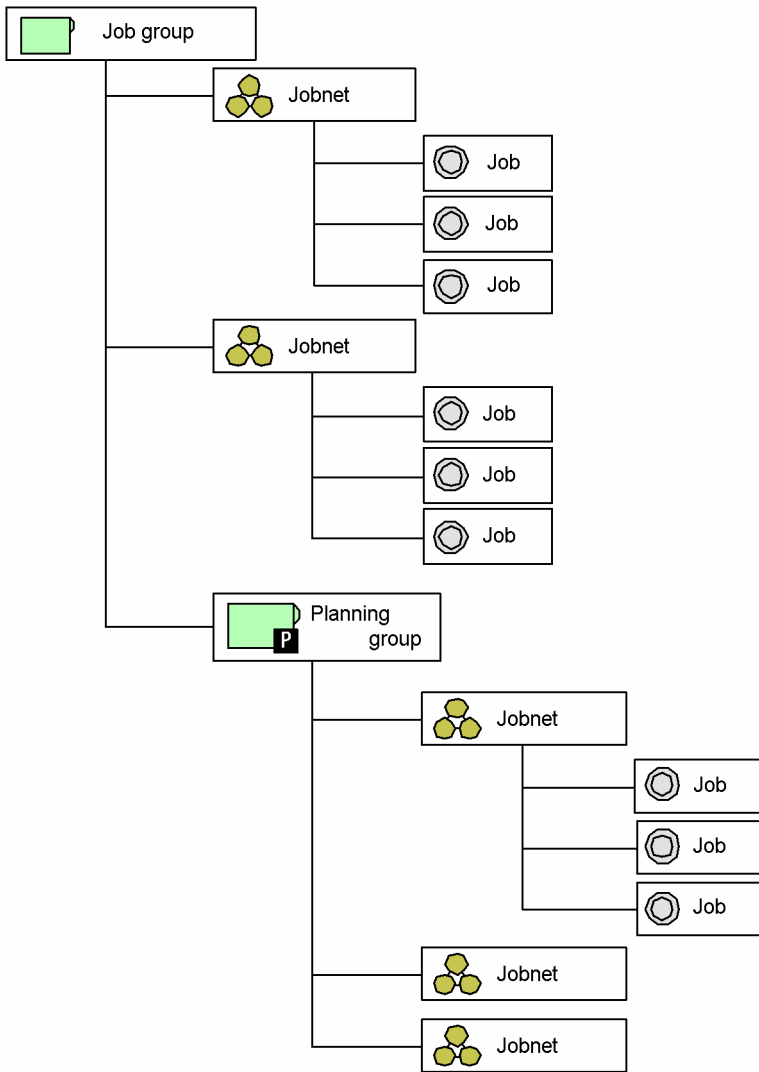
Figure 3–1: Jobnet and jobs



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.

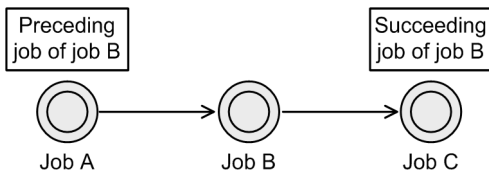
#### (1) Jobs

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 [3.1.4 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 [7.2 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 the *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 [3. Messages beginning with KAVS \(Messages about the scheduler and the common processing\)](#) in the manual *JP1/Automatic Job Management System 3 Messages*. For the output destination and the output conditions, see [1.3.2 Output destinations of messages beginning with KAVS](#) in the manual *JP1/Automatic Job Management System 3 Messages*.  
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 *JP1/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
- Passing information setting job
- HTTP connection 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 four types of standard jobs:

- Unix job
- PC job
- QUEUE job
- Flexible 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 <sup>#1</sup>	Process executed on a UNIX host.	<ul style="list-style-type: none"> <li>• Executable file</li> <li>• Shell script</li> </ul>
2	PC job <sup>#1</sup>	Process executed on a Windows host.	<ul style="list-style-type: none"> <li>• .exe file</li> <li>• .com file</li> <li>• .cmd file</li> <li>• .pif file</li> <li>• .bat file</li> <li>• .spt file<sup>#2</sup> (script file created with JP1/Script)</li> <li>• Data file that has a file type (extension) associated with the application</li> </ul>
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"> <li>• Executable file</li> <li>• Shell script</li> <li>• .exe file</li> <li>• .com file</li> <li>• .cmd file</li> <li>• .pif file</li> <li>• .bat file</li> <li>• .spt file<sup>#2</sup> (script file created with JP1/Script)</li> <li>• Data file that has a file type (extension) associated with the application</li> </ul>

No.	Job type	Description	Definable processes
4	Flexible job <sup>#3</sup>	<ul style="list-style-type: none"> <li>Process executed on an agent host that is not directly managed by the manager host.</li> <li>Job broadcast to multiple execution agents.</li> <li>Flexible jobs can be used to execute processing in a cloud environment. Load balancers are supported.</li> </ul>	<ul style="list-style-type: none"> <li>Executable file</li> <li>Shell script</li> <li>.exe file</li> <li>.com file</li> <li>.cmd file</li> <li>.bat file</li> </ul>

#1

For details on defining a jobnet that uses Unix jobs and PC jobs, see 2.4.1 *Executing a process in a specified file (example of defining a jobnet consisting of standard jobs)* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

#2

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

#3

For details on defining a jobnet that uses flexible jobs, see 2.4.10 *Executing jobs in a cloud environment (example of defining a jobnet that uses flexible jobs)* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*, and 2.4.11 *Executing a job on multiple execution agents simultaneously (example of defining a jobnet that uses flexible jobs)* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

#### Cautionary note

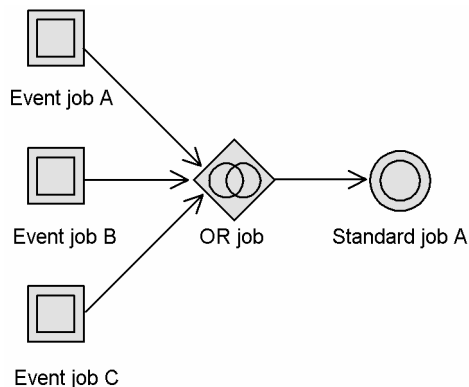
You can define flexible jobs on any manager host regardless of its OS. However, for the execution host of flexible jobs, you can specify only manager or agent hosts that run on Windows or Linux.

## (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.

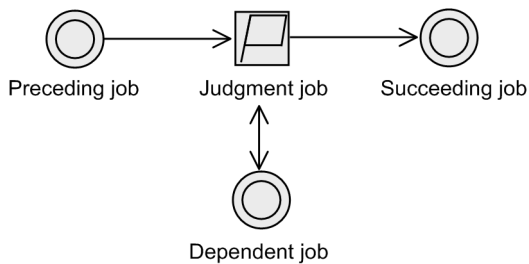
For details on defining a jobnet that uses an OR job, see 2.4.2 *Executing a process when one of multiple conditions is satisfied (example of defining a jobnet that uses an OR job)* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

### (c) Judgment job

A judgment job checks whether a given condition is satisfied. A job that depends on the results of a judgment job is called a *dependent job*, and the results of a judgment determine whether a dependent job is executed. If there are two or more dependent jobs for a judgment job, define these jobs as a nested jobnet. A nested jobnet consisting of multiple dependent jobs is called a *dependent jobnet*. Dependent jobs and dependent jobnets are generically called *dependent units*. In a judgment job, you can set a condition for executing a dependent unit. If the condition is true, the dependent unit is executed, followed by the succeeding unit. If the condition is false, the succeeding unit runs without execution of the dependent unit. However, if the dependent unit ends abnormally, the succeeding unit is not executed.

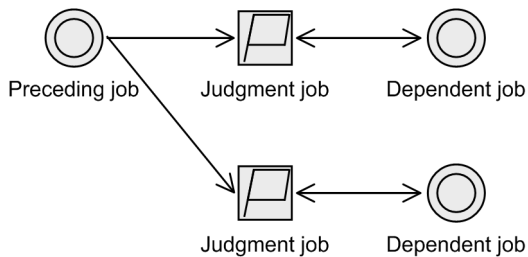
The following figure shows an example of a jobnet that uses a judgment job.

Figure 3–5: Jobnet example 1 (only one judgment job used)



You can also create a jobnet in which multiple judgment jobs run in parallel. If multiple judgment jobs evaluate to true, their dependent jobs are executed individually.

Figure 3–6: Jobnet example 2 (multiple judgment jobs used)



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 unit 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
  - Return code within judgment value range
  - Return code not within judgment value range
- Judgment based on the presence of a file

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

The following conditions can be set:

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

If the specified file is a network file, judgment based on the presence of the file is possible only when the file satisfies the following conditions:

In Windows:

The file can be accessed with an address in UNC format by setting the JP1/AJS3 service account as the user account.

In UNIX:

The file can be accessed by mounting a file system (such as an NFS file system) via a network.

Note, however, that if the network cannot be accessed, the system assumes that the file does not exist. In such cases, judgment based on file presence might not be performed correctly due to a communication error or similar problem.

If you want to determine whether a network file that is not on the manager host is present, we recommend that in the preceding job, you define processing that will copy the file to the manager host if the file exists. This processing will allow the judgment job following the preceding job to correctly judge file presence.

- Judgment based on variable comparison

Determine whether to execute the dependent unit 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 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?
- Is the variable value within the condition value range?
- Is the variable value outside the condition value range?

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 condition value not contained within the variable 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 dependent 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 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

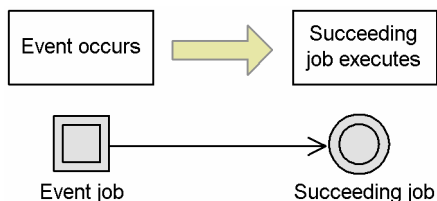
For details on defining a jobnet that uses a judgment job, see *2.4.3 Dynamically changing a process depending on the result of a preceding job (example of defining a jobnet that uses a judgment job)* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

#### (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 3–7: 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 <sup>#</sup>	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 *JP1/Automatic Job Management System 3 Linkage Guide*.

#

Does not apply to Linux.

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 *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

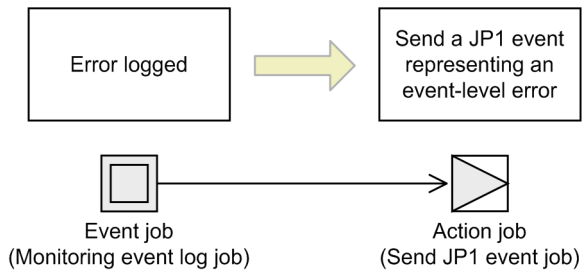
The event information received by an event job can be defined as a variable (macro variable) inherited by the succeeding job. For details about macro variables, see 2.2.6 *Considerations regarding the use of macro variables* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*. For details about passing event information via macro variables, see 2.4.4(6) *Passing information received by an event job* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

**(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 3–8: 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 <sup>#</sup>	Sends a message to TP1/Message Queue or MQSeries.
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 *JP1/Automatic Job Management System 3 Linkage Guide*.

#

Does not apply to Linux.

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 *JP1/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.
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	ADSHPC, ADSHUX, ADSHAPPEXEC	JP1/Advanced Shell	Works in conjunction with JP1/Advanced Shell to execute the script file of JP1/Advanced Shell.
5	DHAJEW	JP1/DH - AJE	Works in conjunction with JP1/DH - AJE to exchange data with JP1/DH - Server.

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 *JP1/Automatic Job Management System 3 Linkage Guide*.

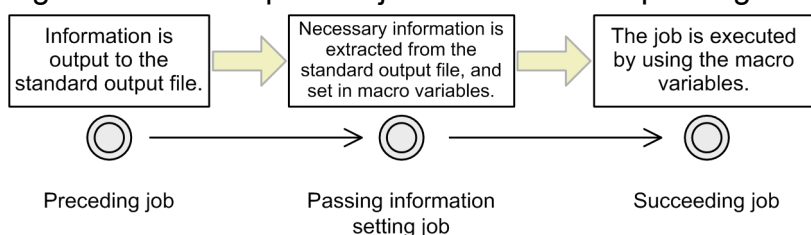
## (g) Passing information setting job

A *passing information setting job* is used between a preceding job and a succeeding job. After the preceding job has output information to the standard output file, the passing information setting job extracts the necessary information from the file, and passes it to the succeeding job. This mechanism allows a succeeding job to receive dynamically changing information that cannot be known when the job is defined from the standard output file of the preceding job.

A passing information setting job uses macro variables to pass information to the succeeding job. For details about macro variables, see 2.2.6 *Considerations regarding the use of macro variables* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

The following figure shows an example of a jobnet that uses a passing information setting job.

Figure 3–9: Example of a jobnet that uses a passing information setting job



For details about defining a jobnet that uses a passing information setting job, see 2.4.9 *Passing information that changes dynamically to a succeeding unit (example of defining a jobnet that uses a passing information setting job)* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

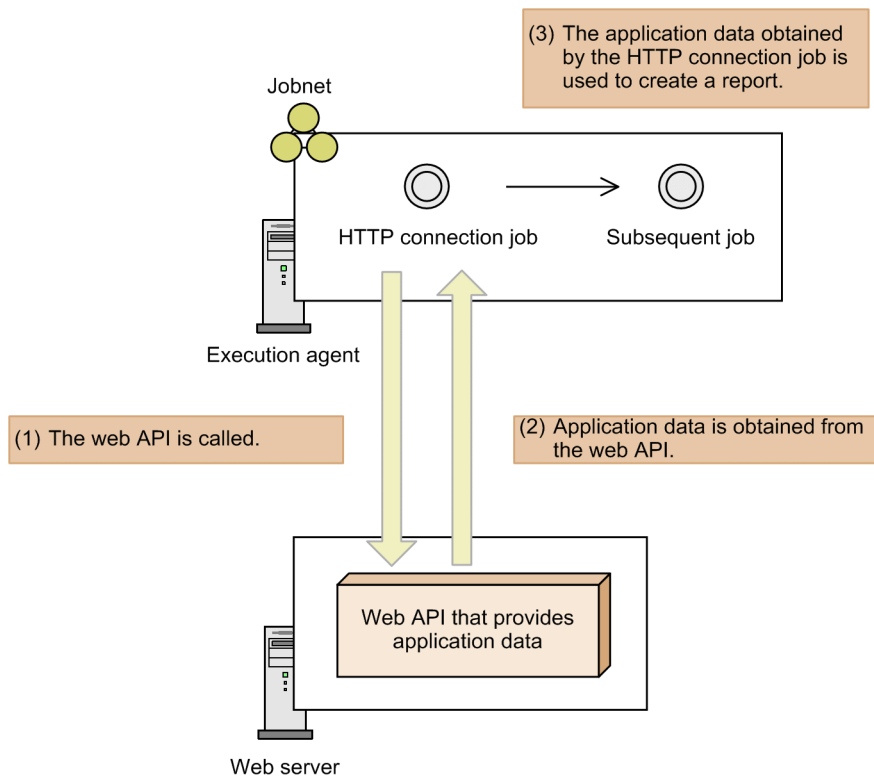
Note that the unit type of passing information setting jobs is *standard custom job*.

## (h) HTTP connection job

*HTTP connection jobs* are jobs that send requests and receive responses via the HTTP protocol. For example, by using an HTTP connection job, you can call the REST API or another web API that is available on a cloud or web server in JP1/AJS3. In other words, HTTP connection jobs allow linkage between a JP1/AJS3-based business system and a business system on the web.

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

Figure 3–10: Example of a jobnet that uses a HTTP connection job



For details about defining a jobnet that uses a HTTP connection job, see 2.4.12 *Linking with a business system on the web (example of defining a jobnet that uses HTTP connection jobs)* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

### Cautionary notes

- You can define HTTP connection jobs on any manager host regardless of its OS. However, for the execution host of HTTP connection jobs, you can specify only manager or agent hosts that run on Windows or Linux.
- The service for executing HTTP connection jobs is fixed to **Standard**. You cannot change the service by using a job definition for HTTP connection jobs.

## (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. Therefore, in each root jobnet, the information necessary for scheduling executions needs to be defined. 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. 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 *JP1/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<sup>#</sup>
- Timeout period
- Schedule option
- Jobnet monitoring
- Execution sequence control
- Hold<sup>#</sup>
- Type<sup>#</sup>

#

Can also be set for a nested jobnet.

The number of logs to keep, priority, timeout period, jobnet monitoring, hold, and type 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 *JP1/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 `ajsshow` command. For the command syntax, see `ajsshow` in [3. Commands Used for Normal Operations](#) in the manual *JP1/Automatic Job Management System 3 Command Reference*.

As the number of logs to keep, you can set a value from 1 to 99. However, the maximum number of logs to keep can be increased to 999 by using the scheduler service environment settings (using the function for increasing the maximum number of logs to keep). For details about the scheduler service environment settings, see the description of the `MAXSAVEGEN` environment setting parameter in [20.4 Setting up the scheduler service environment](#) in the *JP1/Automatic Job Management System 3 Configuration Guide*.

Past execution results are automatically deleted when they exceed the number of generations set in the **number of logs to keep**. For example, if the number of logs to keep for a jobnet is set to 1 and the jobnet is executed more than once a day, only the last execution result can be checked. 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. You can set priority individually for standard jobs (excluding flexible jobs), HTTP connection jobs, and nested jobnets.

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

Note:

You can set **None** for the priority of a job or of a jobnet. If **None** is set, you can specify a default priority to be used for each scheduler service. For details, see [20.4.2\(III\) DEFAULTPRIORITY](#) in the *JP1/Automatic Job Management System 3 Configuration Guide*.

#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 `jajs_spmc` 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 *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

- 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 canceled, 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 a timeout period of **1 day** or **2 days** is set, the *Waiting to execute* or *Being held* status will be canceled when the timeout period elapses, and the status will change to *Skipped so not executed*. If **Unlimited** is set for the timeout period, the status of the root jobnet will not change to *Skipped so not executed* because of the timeout period.

The timeout period corresponding to each timeout setting is as follows:

- **1 day**

23 hours, 59 minutes, and 59 seconds after the base time on the scheduled start day

A timeout period of 2 days is assumed if the root jobnet is operating on a 48-hour basis and the scheduled start time falls between 24:00 and 47:59.

- **2 days**

47 hours, 59 minutes, and 59 seconds after the base time on the scheduled start day

- **Unlimited**

The status of the root jobnet will not change to *Skipped so not executed* because of the timeout period.

For details about the time formats of root jobnets, see [3.3.1 Time formats and schedule rules for root jobnets](#).

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).

- **Hold**

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

For a root jobnet, you can determine whether to hold it based on the previous termination status.

- **Type**

You can specify whether to use a unit as a recovery unit. Select either **Normal** or **Recovery** as the unit type. Note that you must always select **Normal** for a root jobnet. The default is **Normal**.

A jobnet whose type is set to **Recovery** is called a *recovery jobnet*. Recovery jobs and recovery jobnets are collectively called *recovery units*.

### (3) Jobnet connector

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

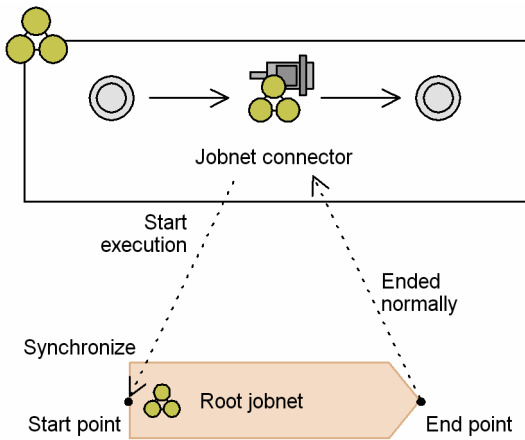
- **Waits for the preceding root jobnet to end**

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 3–11: 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.

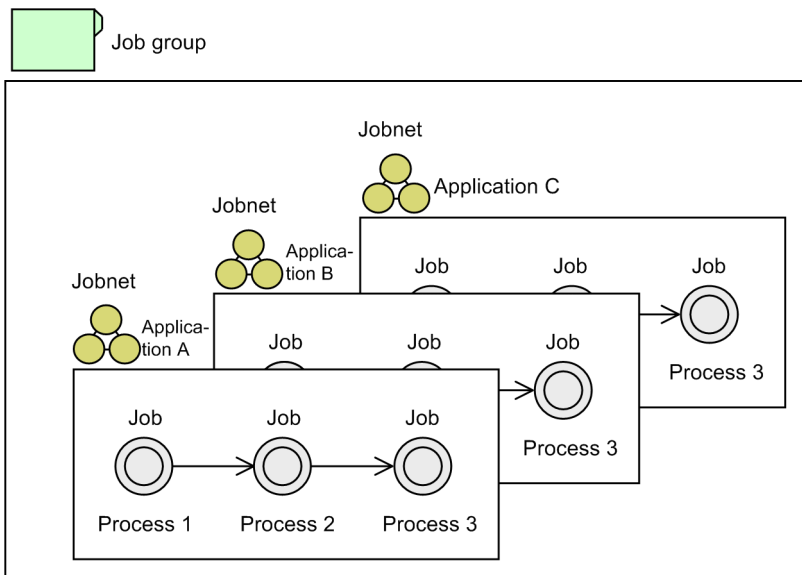
For details about jobnet connectors, see *2.2.4 Using jobnet connectors to control the order of root jobnet execution in the JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

## (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 3–12: 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*.

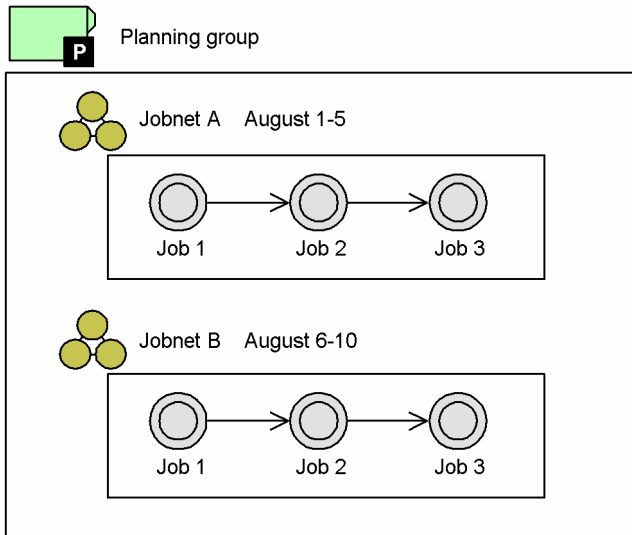
Note that job groups are jobnet management units and cannot be executed in their own right.

## (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 3–13: 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.

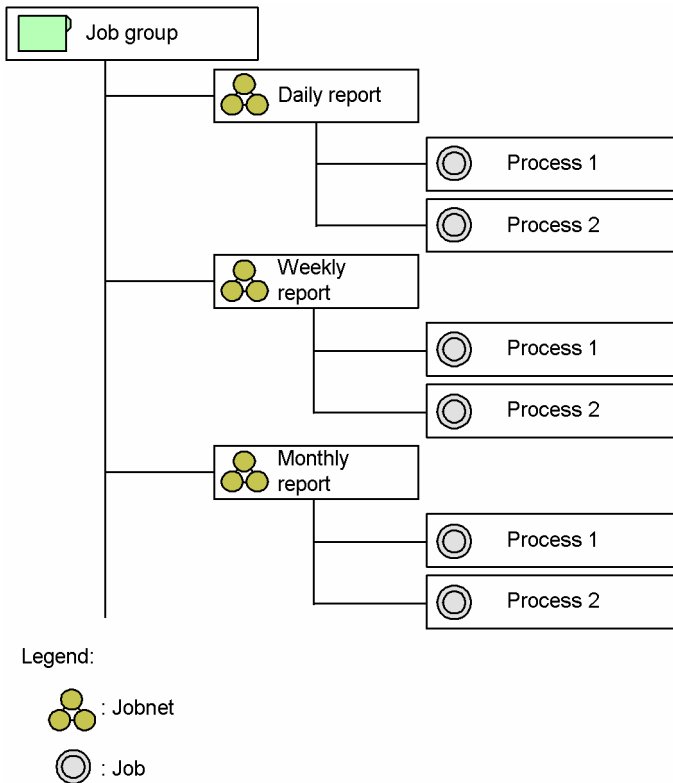
Below, we provide some examples of jobnet creation to illustrate how to group processes into a jobnet.

- 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 3–14: Example of management by application category



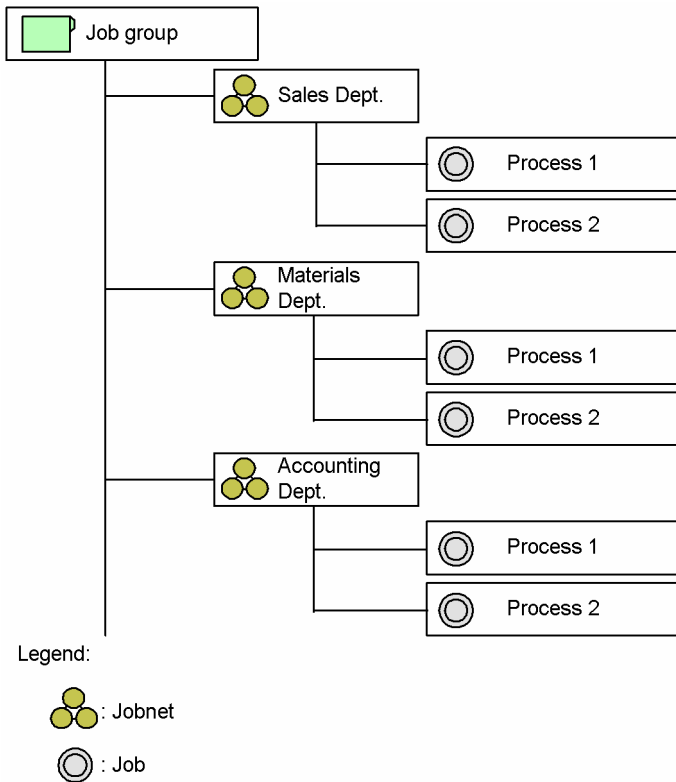
- Example of management by department

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 3–15: Example of management by department

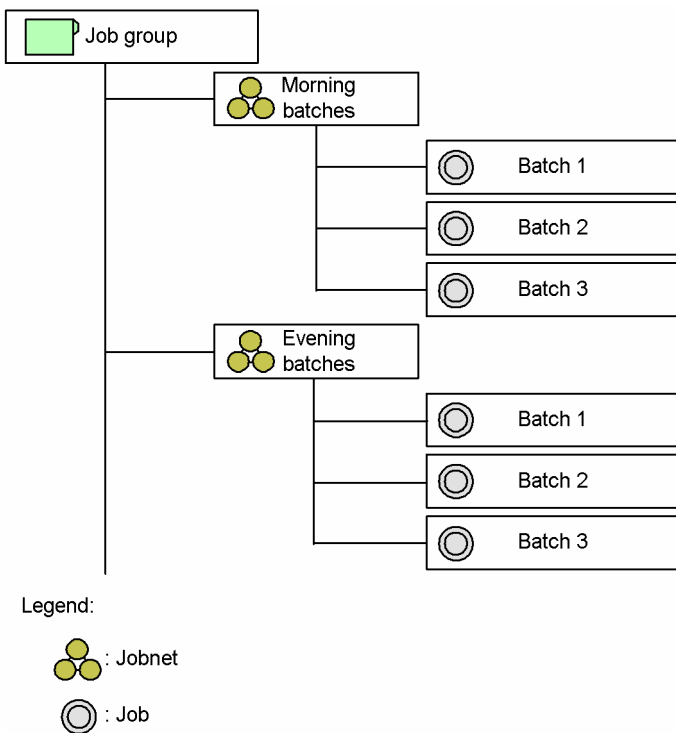


- Example of management by processing category

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 3–16: 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 Applications that use macro variables

Macro variables are variables used to pass information from one job to another job during execution. Using macro variables can automate an application that processes a parameter whose value varies throughout execution of the jobs in the application.

The use of macro variables enables the following:

- Passing event information received by an event job to the succeeding job  
Event information received by an event job can be passed to the succeeding job. For example, if the event job is a file monitoring job, the name of an updated file and the update time can be passed to the succeeding job.
- Extracting necessary information from the standard output file of the preceding job, and passing it to the succeeding job  
A passing information setting job can extract information based on a regular expression from the standard output file that contains the information output by the preceding job, and can pass the extracted information to the succeeding job.
- Passing values specified in a jobnet during registration for execution to subordinate units  
Values specified in a jobnet as passing information when the jobnet is registered for execution can be passed to macro variables defined in units subordinate to the jobnet.

The following table lists topics and the sections in which they are covered in the manual. If you want more information about a topic, see the indicated section.

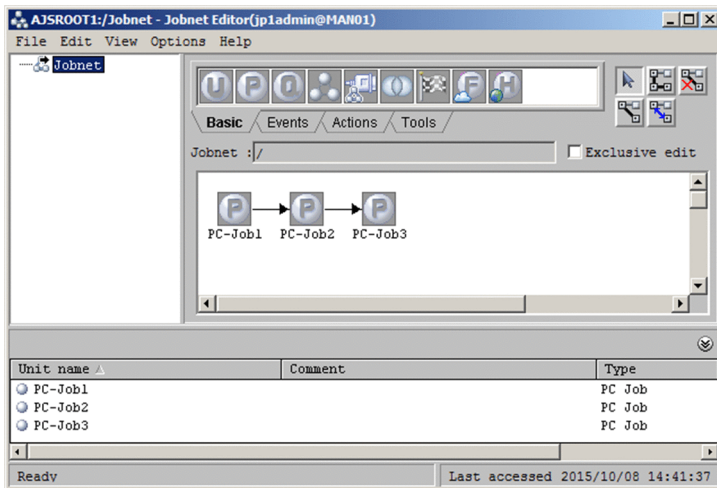
No.	Topic	Section
1	Details of macro variables	<i>2.2.6 Considerations regarding the use of macro variables in the JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide</i>
2	Event jobs and macro variables	<i>2.4.4(6) Passing information received by an event job in the JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide</i>
3	Passing information setting jobs and macro variables	<i>2.4.9 Passing information that changes dynamically to a succeeding unit (example of defining a jobnet that uses a passing information setting job) in the JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide</i>
4	Specification of macro variables during registration for execution	<i>4.1.2 Specifying macro variables during registration for execution</i>

### 3.1.4 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 *JP1/Automatic Job Management System 3 Operator's Guide*.

Figure 3–17: Example of creating a job flow in the Jobnet Editor window



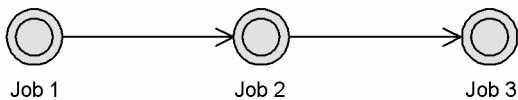
A variety of job flows can be created, depending on how the jobs are sequenced and inter-related. Some examples of job flow creation are provided below.

### (1) 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 3–18: Job flow in a single path

■ One path only

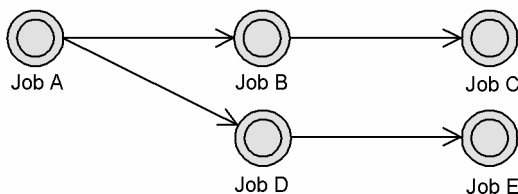


### (2) Multiple processing paths

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

Figure 3–19: Job flow in multiple paths

■ 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.

### (3) Nested jobnet

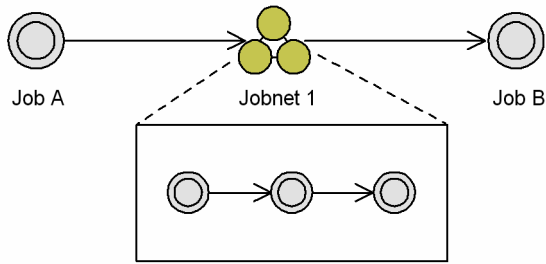
You can use nested jobnets in a job flow. Examples of using nested jobnets are shown below.

Nesting a jobnet

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

Figure 3–20: Example of using nesting a jobnet

■ Nested jobnet included in a job flow



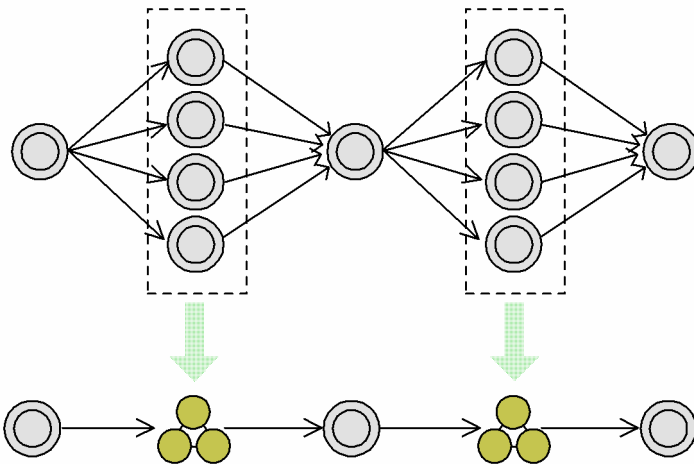
When Job A completes execution, the jobs defined in Jobnet 1 are executed. When processing of Jobnet 1 is finished, Job B is executed.

### Grouping multiple jobs

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

Figure 3–21: Example of grouping multiple jobs into nested jobnets

■ Collections of jobs grouped into two nested jobnets

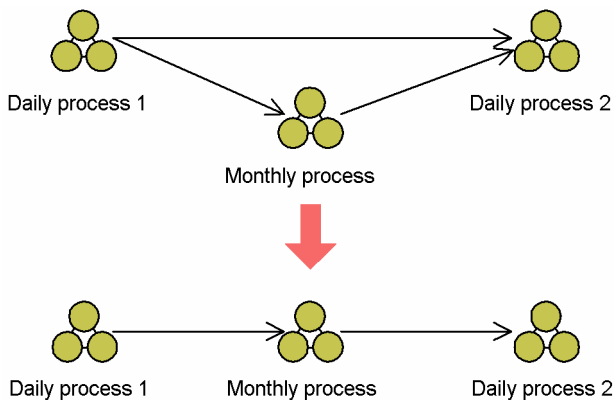


### Joining two paths

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 3–22: 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.

## (4) Sequencing jobs in different jobnets

In JP1/AJS3, you cannot use relation lines to sequence jobs among different jobnets. If you want to sequence jobs among jobnets, create a job flow by using one of the following methods:

- Split a jobnet
- Integrate jobnets
- Use jobnet connectors

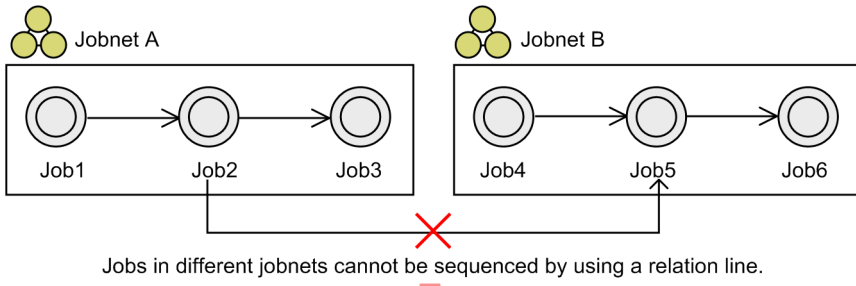
For details about jobnet connectors, see *2.2.4 Using jobnet connectors to control the order of root jobnet execution in the JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

- Use wait conditions

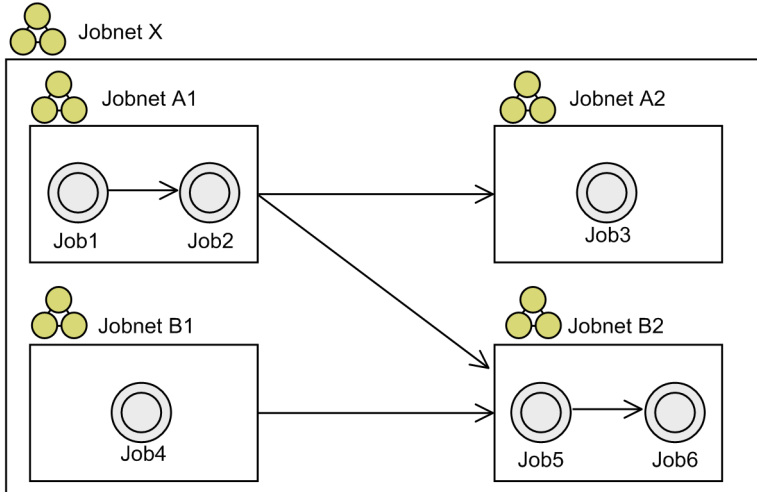
For details about wait conditions, see *2.2.5 Using wait conditions to control the order of unit execution in the JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

The following figure shows examples of sequencing jobs among different jobnets.

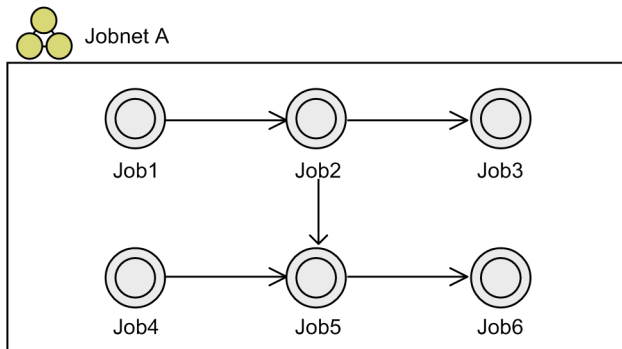
Figure 3–23: Examples of sequencing jobs among different jobnets



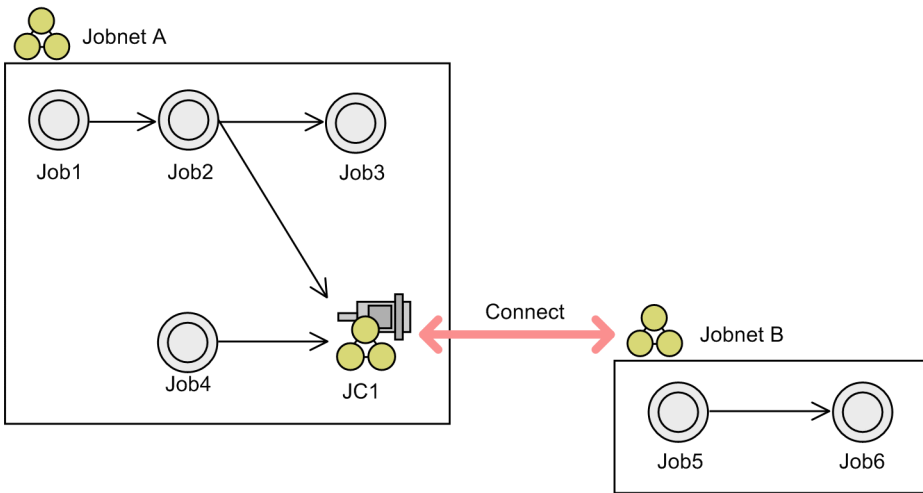
■ Split the jobnets



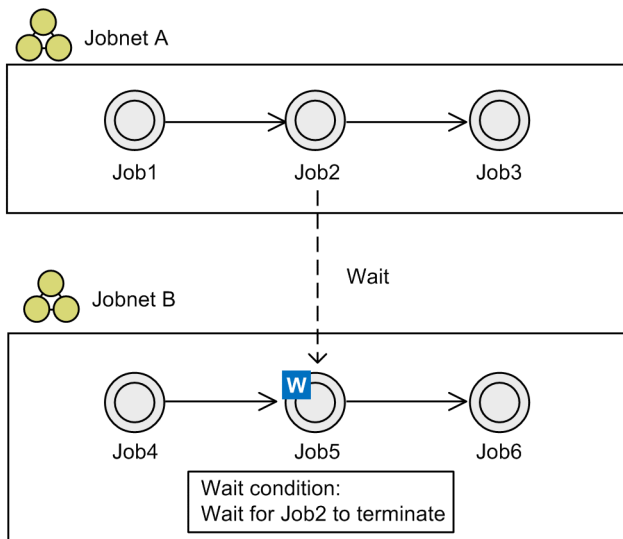
■ Integrate the jobnets



■ Use jobnet connectors



■ Use wait conditions



## 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 *JP1/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 open day and closed 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*.

#### Cautionary note

To change the definition of the base day or the base time, cancel the registration of jobnets registered for execution before you change the definition. If you change the definition of the base day or the base time while a jobnet registered for execution exists, after the change, a jobnet execution schedule might be created for an unintended date or a jobnet might be immediately executed.

For example, if the start time in a schedule rule of a jobnet is defined as 0:00 in relative time and the base time is 0:00, a schedule with a start time of 0:00 will be created. If the base time is 9:00, a schedule with a start time of 9:00 will be created.

Thus, for a jobnet with a scheduled start time of 0:00 in relative time, if the base time is changed to 9:00 after the jobnet is executed at 0:00, the jobnet will be executed again at 9:00. This means that the jobnet will be executed twice in one day based on the same schedule rule.

Moreover, suppose the base time is 9:00 and a jobnet has a schedule rule with a scheduled start time of 0:00 in relative time. If the base time is changed to 0:00 before this jobnet is executed, a schedule with a start time of 0:00 will be created for this jobnet. If the start time has already passed, the schedule will start to execute immediately.



This behavior applies to cases where a jobnet is registered for planned execution. If the base time or the base day is changed, it will affect the time settings, such as the start date and start time for the schedule rules of the jobnet. This could result in the creation of an unintended schedule, such as the one described in the above example. Therefore, to change the defined base day or base time, first cancel the registration of jobnets registered for execution and reexamine the schedule rules of jobnets.

## (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.

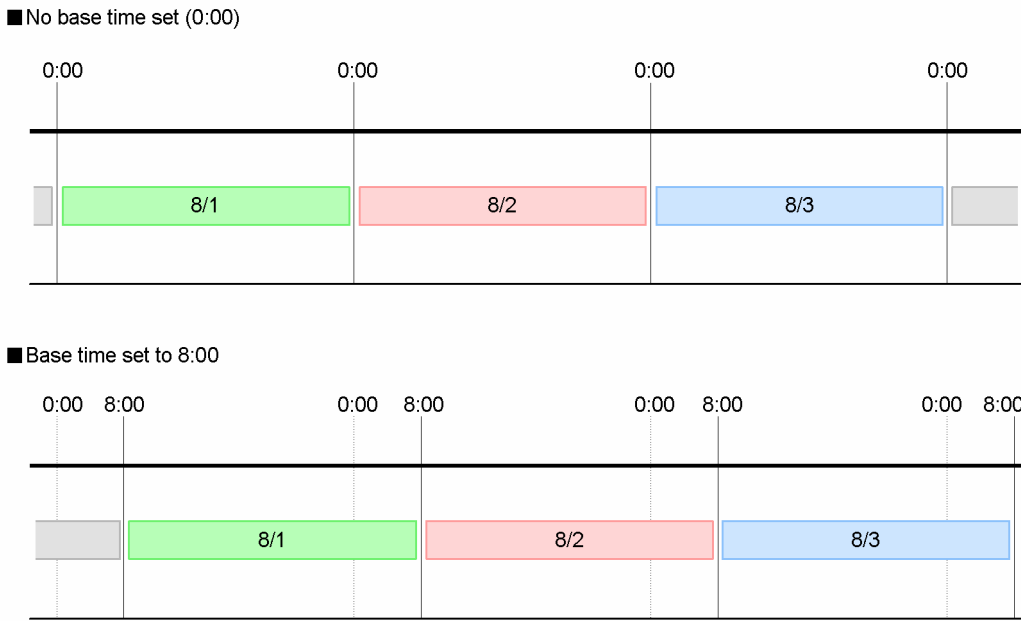
When you set the base day, the month starting from that base day is shown in the Monthly Schedule window, used for checking jobnet schedules and monitoring jobnet status.

## (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 3–24: Base time handling



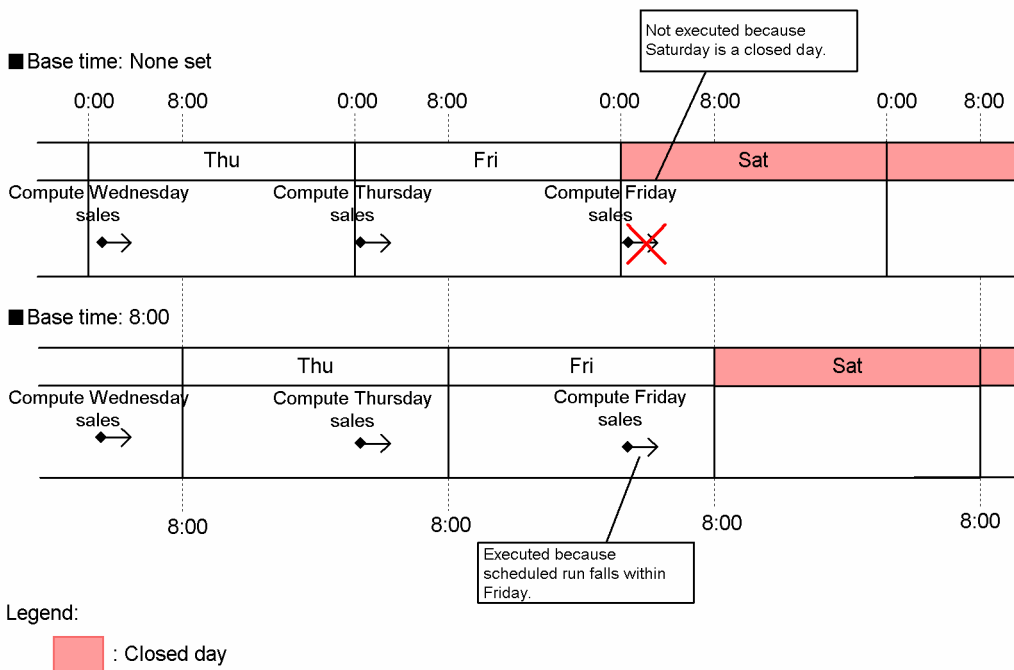
In this example, the base time is set to 8:00. This means that each day starts at 8:00, and lasts until 7:59 the following day.

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 3–25: 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.

#### Supplementary note

In the example in *Figure 3-25*, 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.

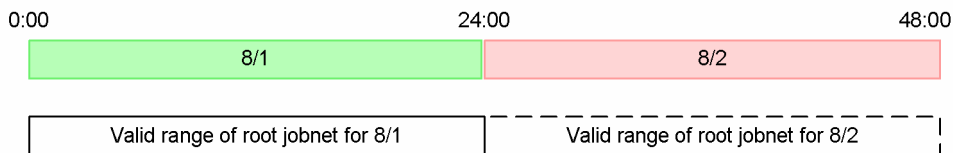
### 3.3.1 Time formats and schedule rules for root jobnets

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 3–26: 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.

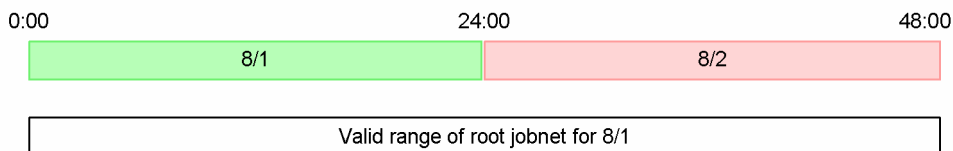
#### Supplementary note

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 3–27: 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 notes

- 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 *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.
- If the root jobnet is operating on a 48-hour basis and *1 day* is defined as the timeout period of the root jobnet, the handling of the timeout period changes according to the scheduled start time as follows:
  - If the scheduled start time is from 0:00 to 23:59 (base time is 0:00):  
The *1 day* in the valid range of the 24-hour-based schedule is used as the timeout period of the schedule.
  - If the scheduled start time is from 24:00 to 47:59 (base time is 0:00):  
The *1 day* in the valid range of the 48-hour-based schedule is used as the timeout period of the schedule.For details, see the description of the timeout period in [3.1.1\(2\) Jobnets](#) and [3.5.1\(1\) Defining an application that extends over two days using a 48-hour schedule](#) in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

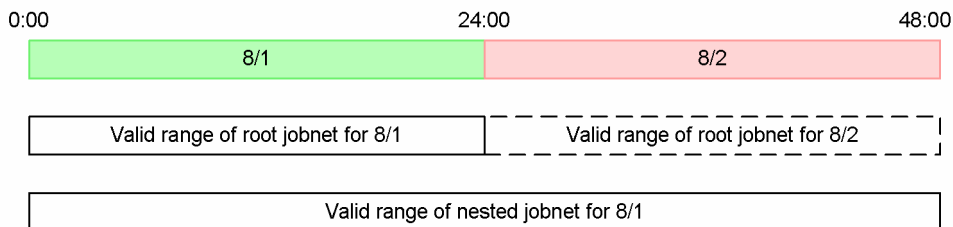
### (3) Time format of nested jobnets

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 3–28: Valid range of a nested jobnet based on a 24-hour schedule (base time 0:00)



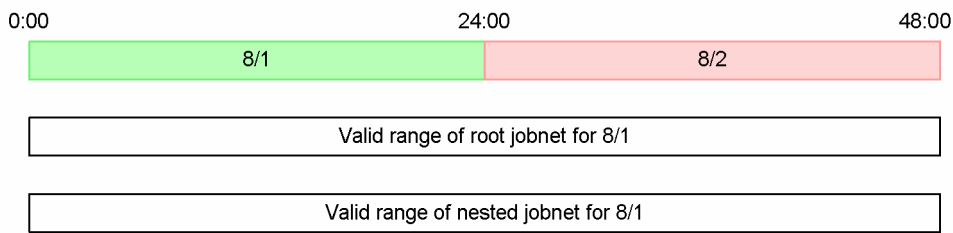
Although the root jobnet has a valid range of 24 hours, the nested jobnet has a valid range of 48 hours. Therefore, if the start time for both jobnets is set to 24:00 or later (between 24:00 and 47:59), their execution dates will differ. For example, if both jobnets are set to start at 25:00 on August 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 3–29: 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 *JPI/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 *JPI/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 *JPI/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
<b>Registered day</b>	--	--
<b>Absolute day</b>	<b>Specify day</b>	Specify a calendar date in the form <i>[ ]th day</i> .
	<b>Specify end of month</b>	Specify the number of days before the end of the calendar month in the form <i>[ ] days before last day</i> .
	<b>Specify day of the week</b>	Specify the day of the week, counting from the start of the calendar month, in the form <i>ordinal number and [ ]th week and day of the week</i> .
<b>Relative day</b>	<b>Specify day</b>	Specify a date, counting from the date set as the base day, in the form <i>[ ]th day</i> .
	<b>Specify end of month</b>	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> .
	<b>Specify day of the week</b>	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> .
<b>Open day</b>	<b>Specify day</b>	Specify an open day, counting only the open days from the date set as the base day, in the form <i>ordinal number</i> .

Type	Start day option	Explanation
<b>Open day</b>	<b>Specify end of month</b>	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> .
<b>Closed day</b>	<b>Specify day</b>	Specify a closed day, counting only the closed days from the date set as the base day, in the form <i>ordinal number</i> .
	<b>Specify end of month</b>	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> .

Legend:

-- : The start day is not specified.

### ■ 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.

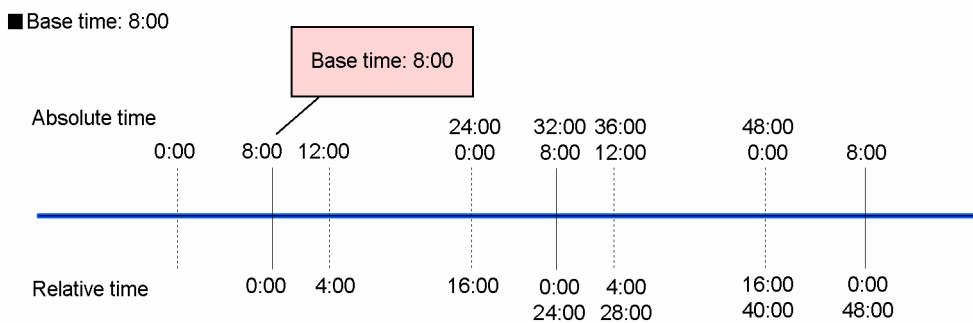
Using absolute time, you can set a schedule by clock time, regardless of whether a base time is set.

**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 3–30: Absolute time and relative time

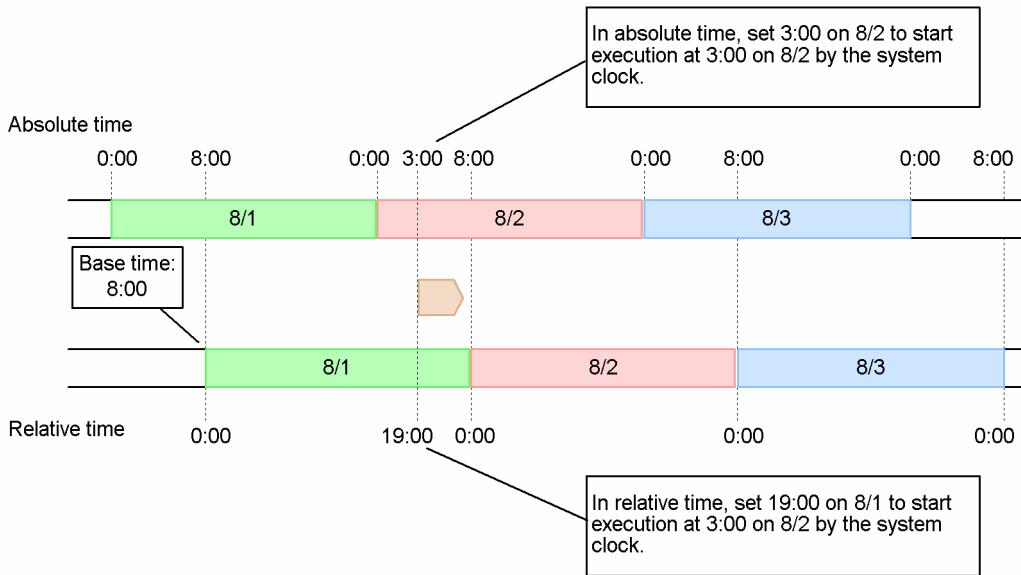


In this example, the base time is set to 8:00. If relative times are specified, the base time of 8:00 in absolute time will be 0:00 in relative time. Thus, 12:00 in absolute time will be 4:00 in relative time, and 24:00 in absolute time will be 16:00 in relative time.

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



Figure 3–31: Difference between settings in absolute time and relative time

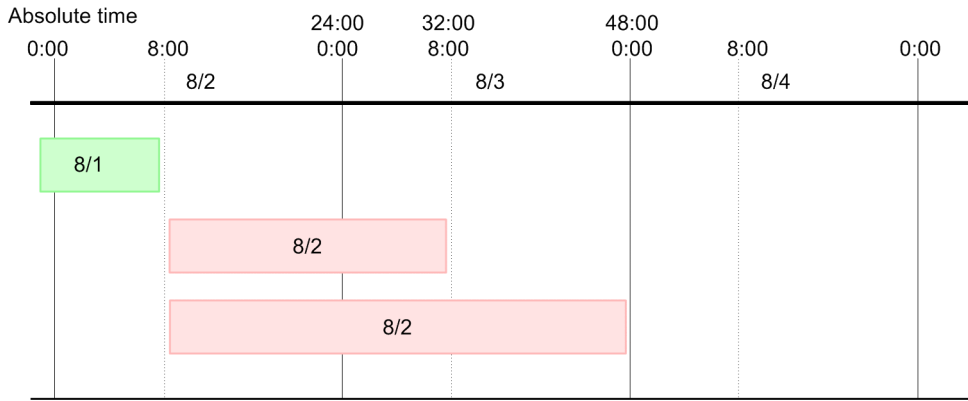


When you use relative time, the specified date matches the date in the JP1/AJS3 calendar. When you use absolute time, however, the date in JP1/AJS3 may differ from the specified date and time because the boundary between one day and the next shifts according to the base time.

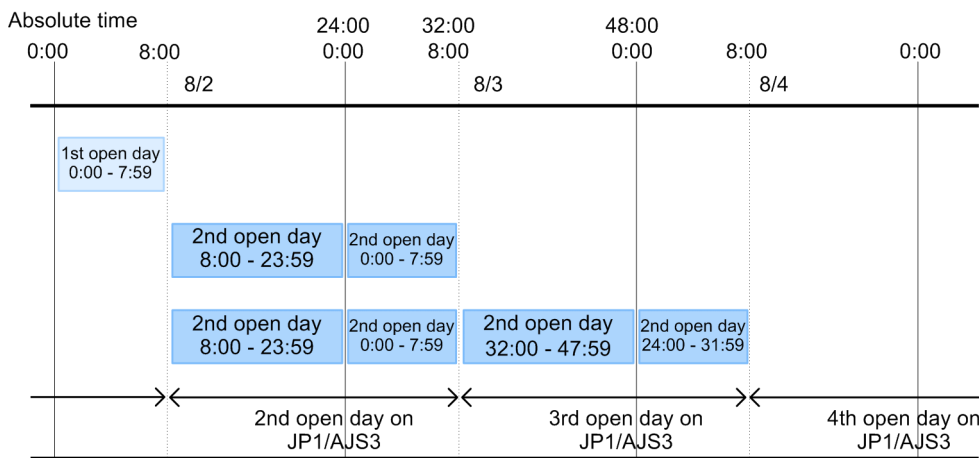
The **Type** option can also affect the behavior of date and time settings. The following figure shows how dates and times are handled in JP1/AJS3 when the time is set in absolute time and a base time is set.

Figure 3–32: Handling of absolute time settings

■ Type set to Absolute day, Relative day, or Registered day (base time: 8:00)



■ Type set to Open day or Closed day (base time: 8:00; start day: 2nd open day (8/2))



- 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.

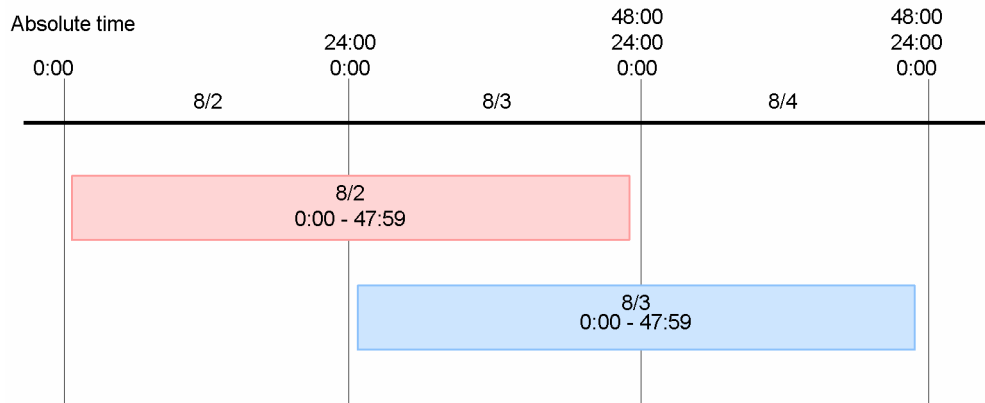
- When **Type** is set to **Open day** or **Closed day**  
Because dates are determined relative to the base time, the date in JP1/AJS3 will be the same as the specified date, but a break occurs in the time continuum.

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 3–33: 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 3–34: Differences between execution dates according to start day option

Execution example 1 Start day: August 1, 20XX

<Aug>							<Sept>							<Oct>						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5						1	2	1	2	3	4	5	6	7
6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14
13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21
20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28
27	28	29	30	31			24	25	26	27	28	29	30	29	30	31				

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

<Aug>							<Sept>							<Oct>						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5						1	2	1	2	3	4	5	6	7
6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14
13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21
20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28
27	28	29	30	31			24	25	26	27	28	29	30	29	30	31				

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

<Aug>							<Sept>							<Oct>						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5						1	2	1	2	3	4	5	6	7
6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14
13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21
20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28
27	28	29	30	31			24	25	26	27	28	29	30	29	30	31				

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**

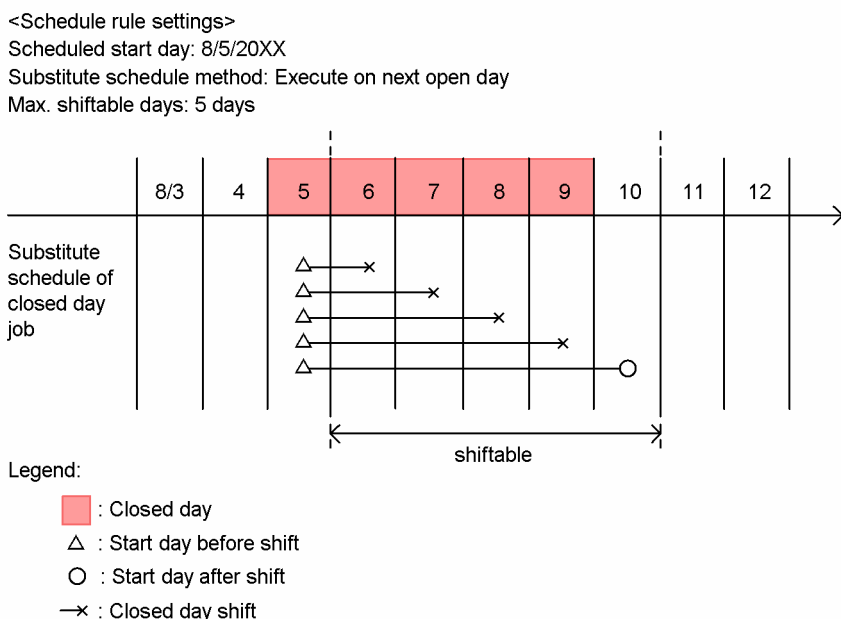
Do not reschedule the jobnet. If the JP1/AJS3 - Manager service is active, execute the jobnet on the closed day. If the JP1/AJS3 - Manager service is inactive, set the jobnet to *Skipped so not executed* status.

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 3–35: Example of a substitute schedule of closed day job



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. If the set duration is exceeded, a delay is reported (a JP1 event is issued and the jobnet status changes in JP1/AJS3 - View or the Web GUI). However, even if a delay occurs, execution of the jobnet is not suspended.

Note that delay monitoring can be set for nested jobnets. 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 (a JP1 event is issued and the jobnet status changes in JP1/AJS3 - View or the Web GUI). 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

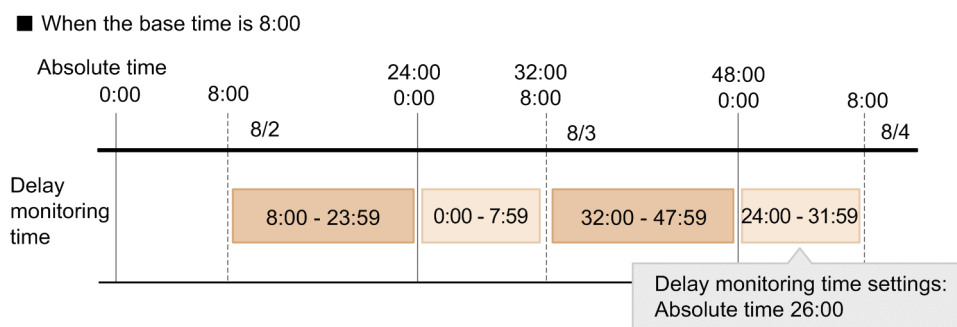
To monitor delays in a jobnet whose execution schedule extends into the next day or starts at 24:00 or later, we recommend that you use a 48-hour schedule without setting a base time (that is, the base time is 0:00). For details, see (a) *Setting the execution start time*. See also 3.5.1(1) *Defining an application that extends over two days using a 48-hour schedule* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

### Supplementary notes

- Be careful if you use an absolute time to specify the start delay or end delay monitoring time when a base time (other than 0:00) is set for the calendar definition used by the jobnet. In this case, although dates are calculated based on the base time, a break occurs in the time continuum. Take special care when you attempt to specify 24:00 or later. In this case, the time that is actually specified might be different from the intended time.

When a base time is set, the delay monitoring time is handled as shown in the following figure.

**Figure 3–36: How delay monitoring time is handled**



As shown above, if the delay monitoring time of a jobnet to be executed on August 2 is set to 26:00 as an absolute time with a base time of 8:00, the actual specified time is 2:00 on August 4.

- 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.
- If you set both of the following types of end delay monitoring, a delay is reported (a JP1 event is issued and the jobnet status changes in JP1/AJS3 - View or the Web GUI) upon the first detection. When a delay is detected and monitored in one type of delay, the delay in the other type of delay monitoring is not monitored.

- End delay monitoring according to the elapsed time from the start of execution of a jobnet
- End delay monitoring starting from the time set as the delay judgment criterion

### (g) Schedule by days from start

*Schedule by days from start* means rescheduling a jobnet by counting 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:

- Open-day basis

When calculating on an open-day basis, the next execution day is calculated from the schedule settings (such as the scheduled start time, processing cycle, and closed-day substitution) and then shifted backward or forward by counting only open days.

- All-day basis

When calculating on an all-day basis, the next execution day is calculated from the schedule settings (such as the scheduled start time, processing cycle, and closed-day substitution) and then shifted backward or forward by counting both open days and closed days.

In other words, you can select one of the following four rescheduling methods:

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

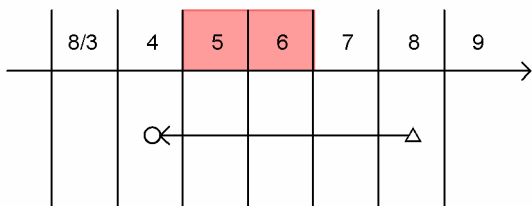
In  $n$ , set the number of days to count.

(Example 1) Execute 2 open days before the scheduled start time

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

Figure 3–37: Example 1 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
- △ : 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.

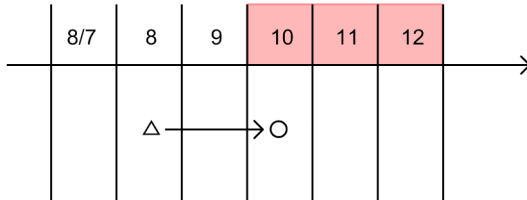


(Example 2) Execute 2 days after the scheduled start time

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

Figure 3–38: Example 2 of rescheduling a jobnet

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



Legend:

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

A jobnet that was scheduled to be executed on August 8 is rescheduled to be executed two days later, regardless of whether the two days include closed days. Therefore, the jobnet will now execute on August 10.

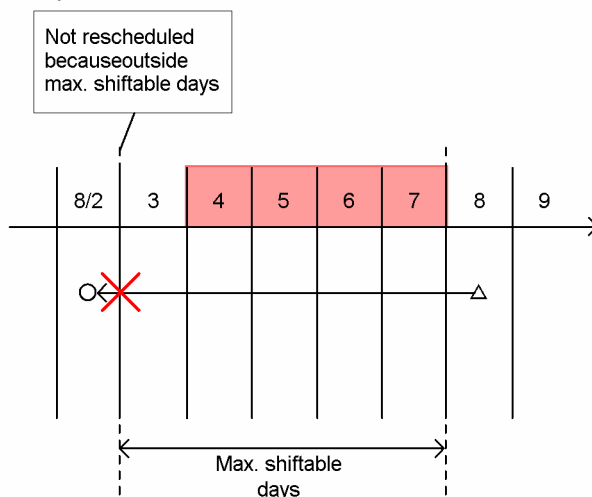
### Max. shiftable days

For rescheduling on an open-day basis, you can specify the maximum number of days by which the next execution day can be shifted (**Max. shiftable days**). The maximum shiftable days includes the number of closed days. If the next rescheduled execution time falls outside the maximum shiftable days, execution is not scheduled for that time.

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 3–39: 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



Legend:

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

Two open days before the start time falls outside the set maximum shiftable days (5 days). Therefore, execution is not scheduled for August 2.

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 *JP1/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 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.

For details on schedule options for a nested jobnet, see also *3.5.3 Defining a different schedule for some jobs in a jobnet* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

### (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 JP1/AJS3 operation*.

#### Cautionary note

If a jobnet is registered for planned execution or for fixed execution by specifying the number of future generations, performing any of the operations below will disable schedule calculations. As a result, the jobnet might be placed in Shutdown status or the execution schedule might not be created. Cancel registration before you execute any of these operations:

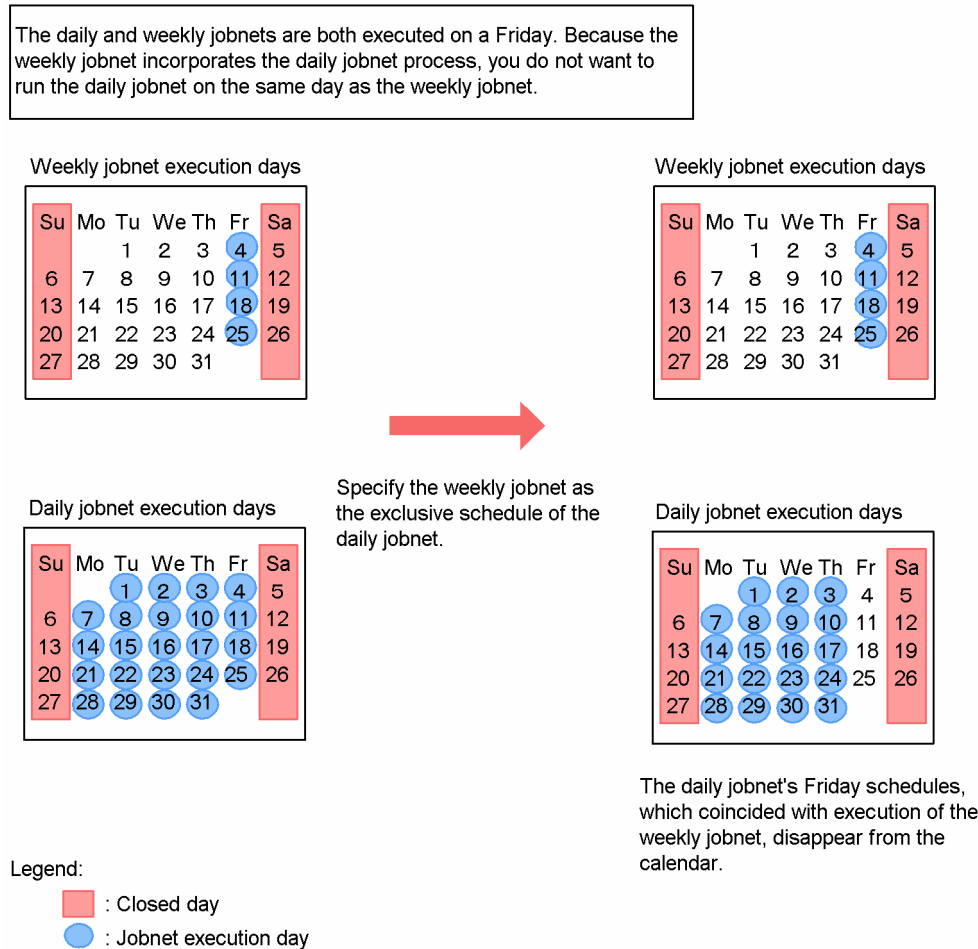
- Change the job group specified in the schedule definition as the job group whose calendar is to be referenced.
- Delete the job group whose calendar is to be referenced.
- Rename the job group whose calendar is to be referenced.
- Execute the `ajsdefine` command with the `-i` or `-f` option specified to update 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, and you do not want them to run on the same day.

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

Figure 3–40: Example of using an exclusive schedule



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 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 3–41: 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\(1\)\(b\) Exclusive schedules for planning groups](#).

Cautionary note

If a jobnet is registered for planned execution or for fixed execution by specifying the number of future generations, performing any of the operations below will disable schedule calculations. As a result, the jobnet might be placed in Shutdown status or the execution schedule might not be created. Cancel 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.
- Execute the `ajsdefine` command with the `-i` or `-f` option specified to update the jobnet specified as an exclusive schedule.

Supplementary notes

- Execution of a jobnet for which other jobnets are specified as exclusive schedules is scheduled based on the schedule definitions of the other jobnets. If the other jobnets have not been registered for execution, the exclusive schedule functionality simulates the execution of these jobnets.
- If the execution start time type is **Registered day** in the schedule definition of a jobnet specified as an exclusive schedule of another jobnet, execution simulation assumes the jobnet starts on the day that the other jobnet is registered for execution. If you do not want the simulation to use this assumption, set the execution start time type of this jobnet to a type other than **Registered day**, and explicitly specify the execution start day.
- If the execution schedule of a jobnet that is specified as an exclusive schedule of another jobnet is changed by **Change plan**, the change is applied but does not affect the schedule of the other jobnet.

### 3.3.3 Concurrent execution and schedule option

Concurrent execution and schedule options are provided as means for determining the JP1/AJS3 behavior if a process does not terminate before the next process starts. 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. When concurrent execution is enabled, each instance can start on time even if the preceding instance has not terminated. 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

*Schedule options* are used to determine how to handle the next execution of a jobnet if execution of the jobnet has not terminated yet when the next execution time arrives and passes.

- 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. Jobnets are not skipped when the previous run was started by a rerun operation. In this case, the execution schedule, which was not skipped, is executed in accordance with the settings for concurrent executions.

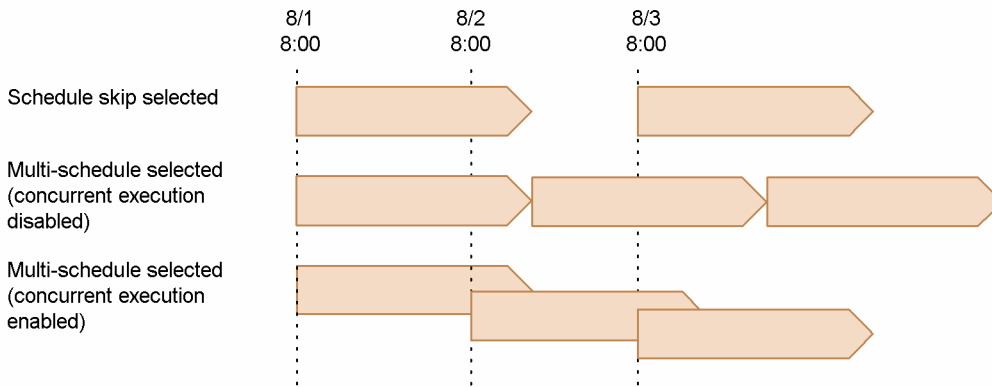
- Multi-schedule

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 3–42: Method of execution with 'Schedule skip' and 'Multi-schedule ' selected

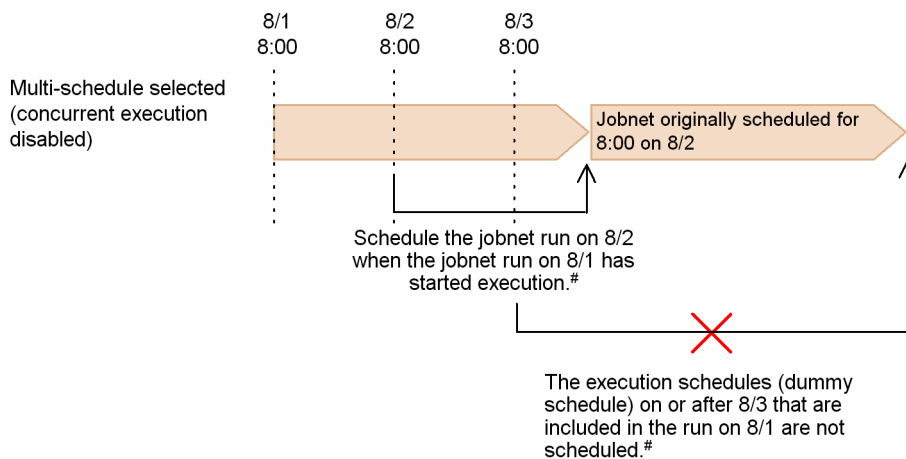
<Jobnet executed daily at 8:00>



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 3–43: Two scheduled jobnet runs held over

Example when daily execution is scheduled for 8:00



# 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.

Cautionary note

If you have set concurrent execution and specified **Multi-schedule**, and the previous execution schedule has not yet started, the next execution schedule does not start when its execution time arrives. The next execution schedule starts only after the previous execution schedule has started.

Supplementary note

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

### 3.3.4 Checking defined schedules

You can check defined schedules in the Monthly Schedule window. For details, see [4.4.1\(2\) Checking schedules in the Monthly Schedule window](#).

## 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*.

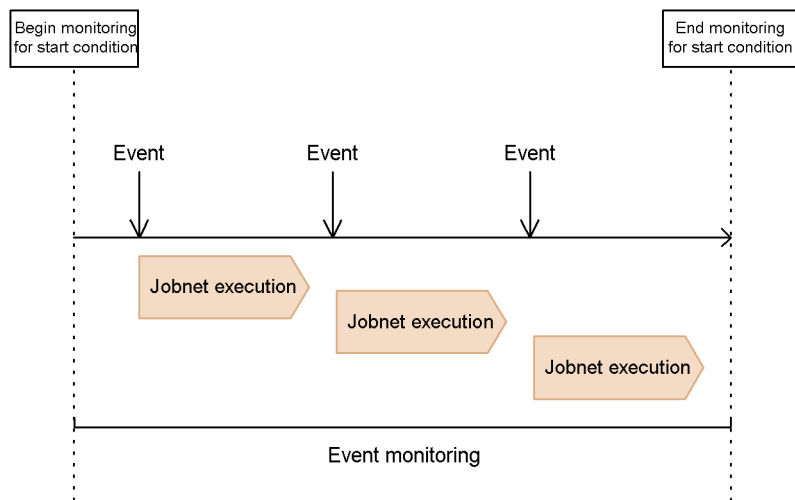
This section describes how to define start conditions and their purpose.

### 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 3–44: 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 *JP1/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.

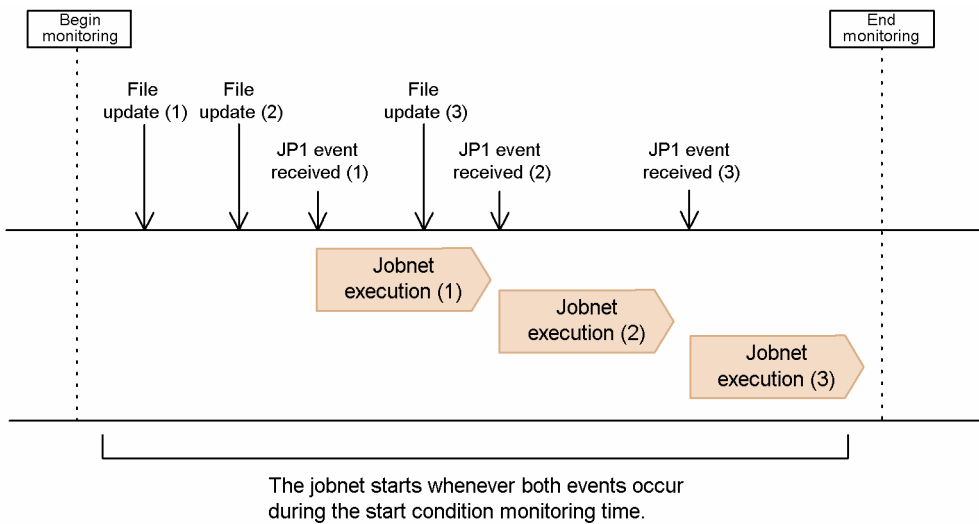
#### AND condition

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 3–45: Example of an AND start condition with multiple events



**Cautionary note**

To monitor multiple conditions by using an AND condition, make sure that same event does not occur in succession. In the example above, make sure that the file update events do not occur in succession.

Occurrence of a specific event in succession might generate a lot of jobnet generations that wait for other events combined by that specific event by the AND operator, which adversely affects performance.

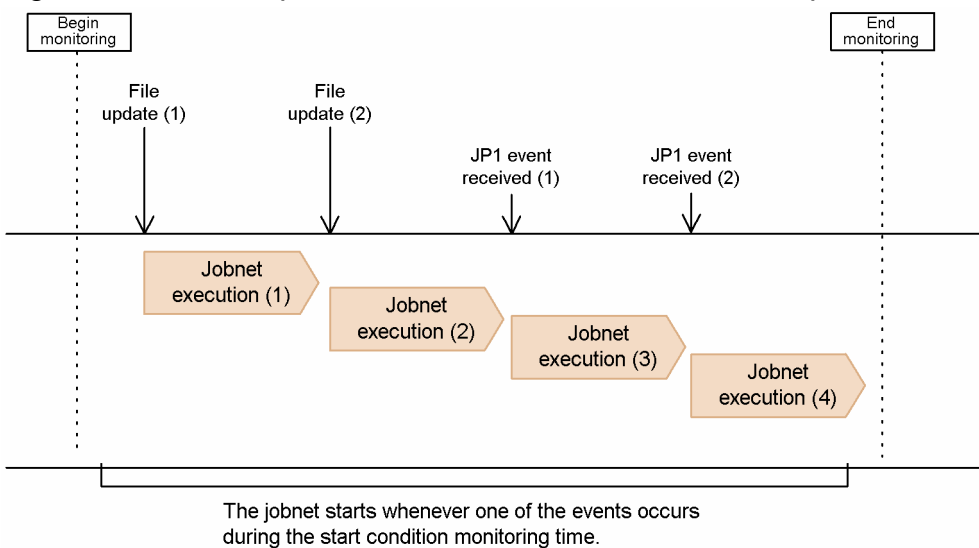
For details about execution generations, see *3.4.2 Monitoring generation and execution generation*.

**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 3–46: 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 *JPI/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.

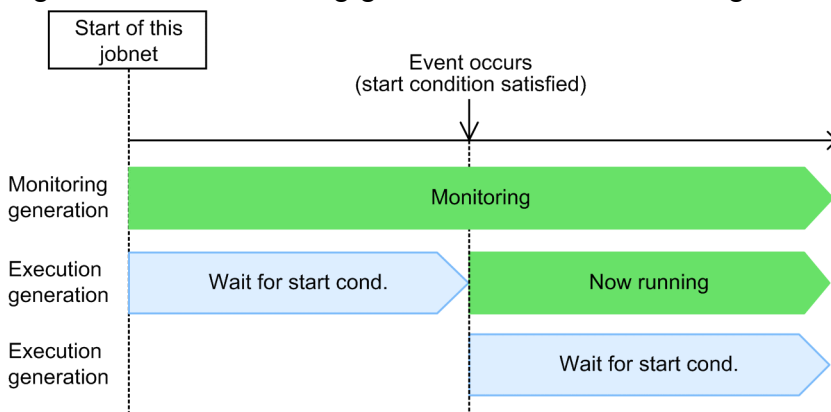
For details on setting the event order option, see 6.3.2 *Setting the event order option* in the *JPI/Automatic Job Management System 3 Configuration Guide* for a Windows host, or 15.3.2 *Setting the event order option* in the *JPI/Automatic Job Management System 3 Configuration Guide* for a UNIX host. For details on estimating the number of events that can be processed with the event order option, see 3.1.5 *Event monitoring performance* in the *JPI/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 3–47: 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

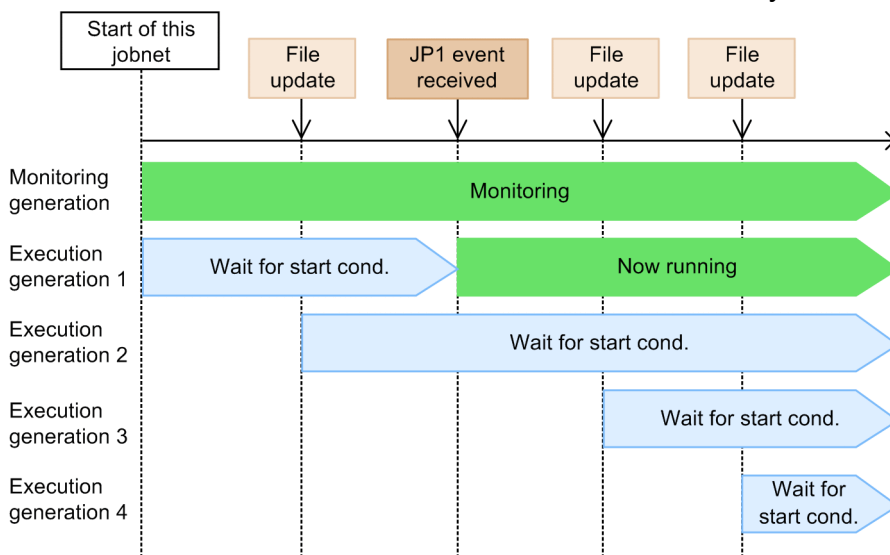
generation is not executed even when the start condition is satisfied, but is instead placed in the *Ended abnormally* status.

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

#### Cautionary note

To monitor multiple conditions by using an AND condition, make sure that only specific events occur in succession. If only particular events occur consecutively, multiple execution generations are generated with the *Wait for start cond* status. For example, assume that you have defined a start condition by combining the file monitoring job and JP1 event reception monitoring job by the AND operator. In this case, if only the file update event occurs in succession, jobnet execution generations waiting for the JP1 event received event in the *Wait for start cond.* status are generated as shown in the following figure.

Figure 3–48: Example when a specific event occurs in succession for a jobnet whose start condition consists of events combined by the AND operator



If a large number of execution generations are generated, performance might be affected.

Additionally, note that the maximum number of execution generations that can be retained by a single monitoring generation is 7,680. If the number of execution generations exceeds 7,680, all monitoring generations are placed in the *Monitor terminated* status.

#### Supplementary note

If you define a hold attribute for a jobnet with start conditions and execute the jobnet, or if you set a hold when registering a jobnet for execution, only monitoring generations of the jobnet are placed in the hold status. When these generations are released from the hold, monitoring starts.

### 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.

#### (1) Setting the valid range

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. However, for planned or fixed execution registration, even if start condition monitoring has terminated, the monitoring starts again within the valid range when the next execution time, determined based on the jobnet's schedule rule definition, is reached.

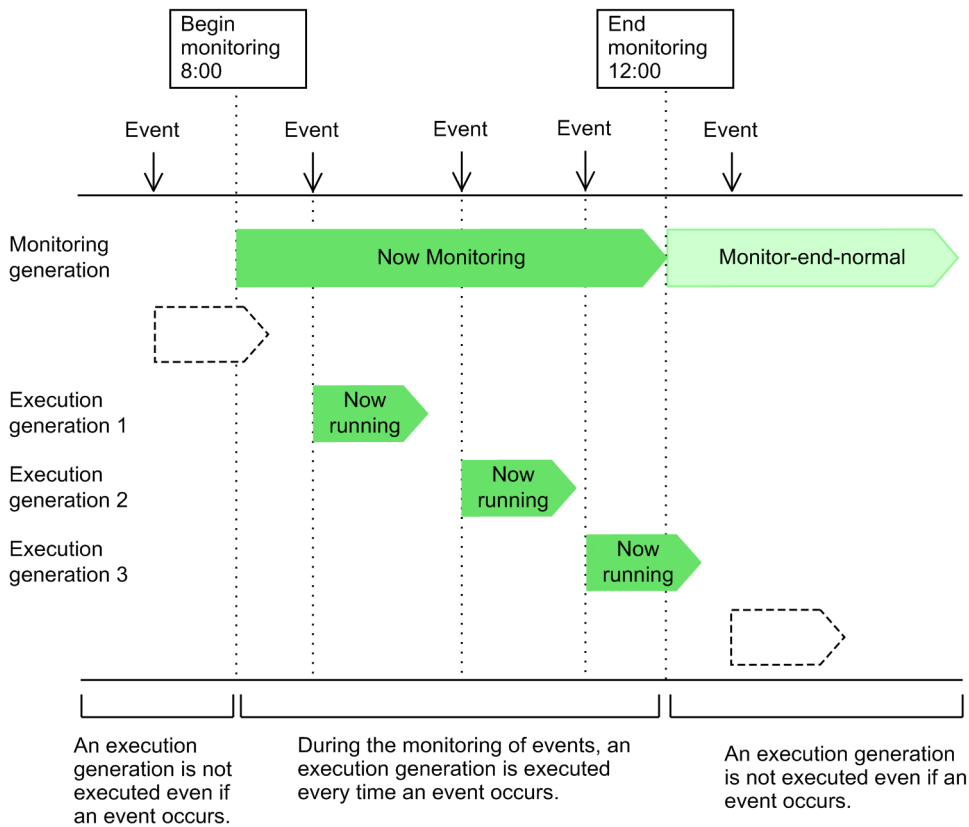
## (2) Valid range specified in absolute time

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 3–49: Example of setting the valid range of a start condition (absolute time)

Start condition monitoring time: 8:00 - 12:00



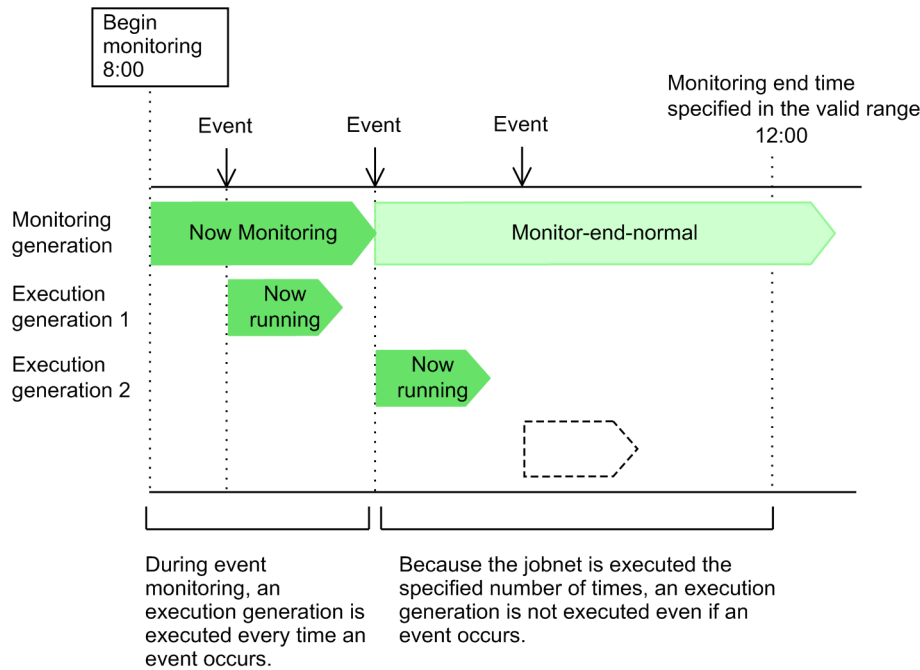
### (3) Valid range specified in absolute time and number of executions

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 3–50: 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



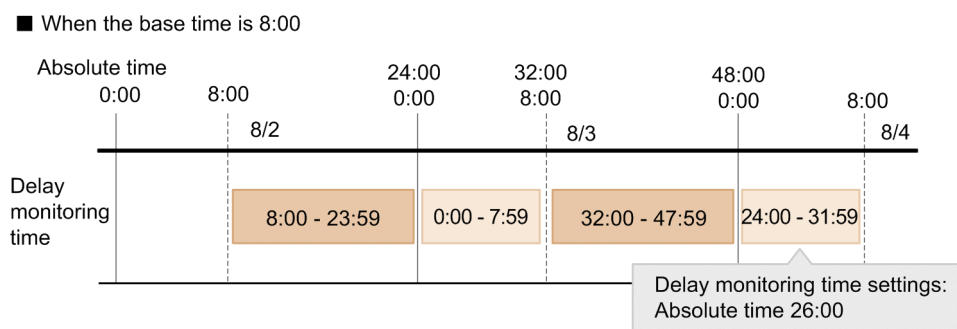
In the above example, when the number of jobnet executions reaches the specified count (2 times), subsequent occurrences of the event are regarded as invalid even though the monitoring end time (12:00) specified in the valid range has not yet arrived.

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 3–51: 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 note

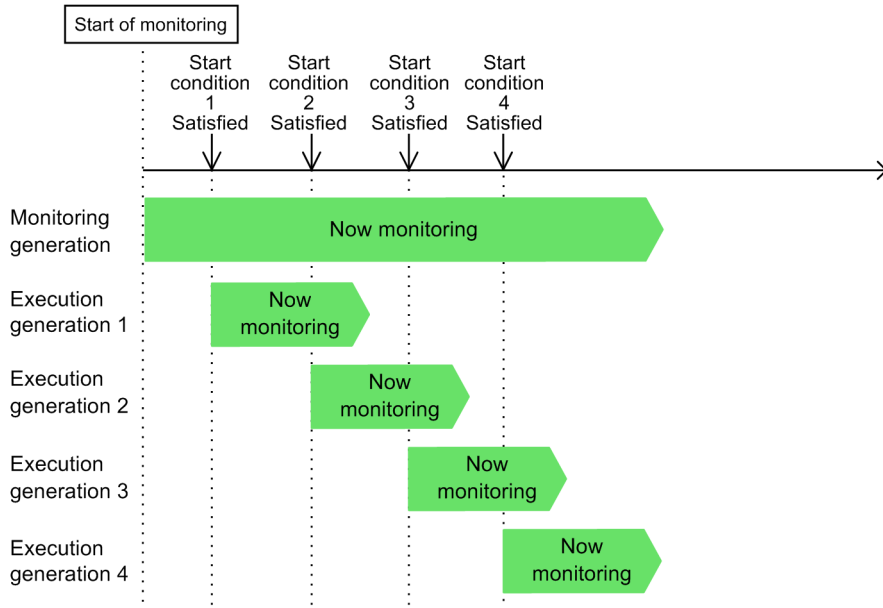
If you specify the valid range in absolute time, make sure that the specified time is later than the execution start time of the jobnet. If you set an earlier time, the monitoring generation will be set to *Monitor terminated* status as soon as it starts execution.

### 3.4.4 Concurrent execution of execution generations

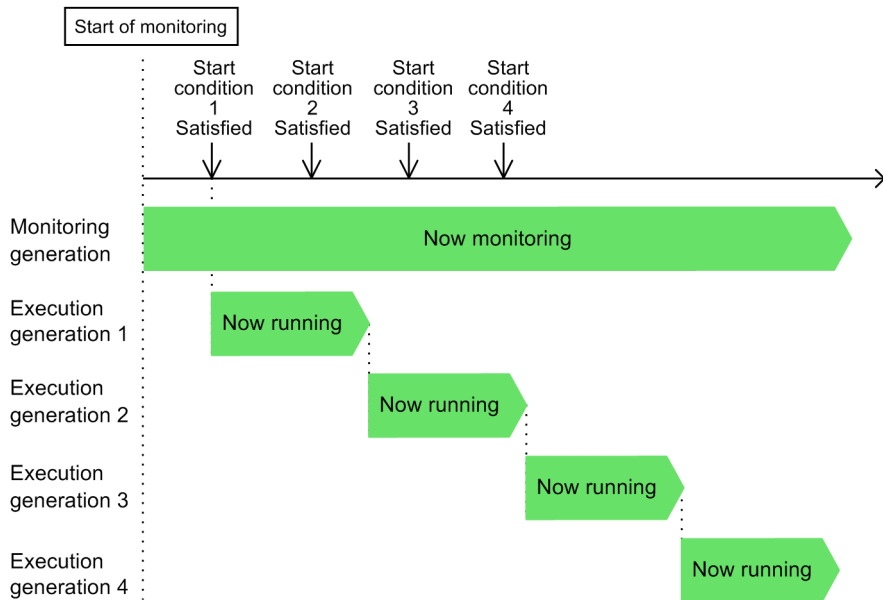
If a start condition is satisfied when a monitoring generation is in the *Now monitoring* status and an execution generation is in the *Now running* status, a new execution generation is generated. The behavior of the execution generation depends on whether concurrent execution is enabled or disabled.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

Figure 3–52: Jobnet behavior according to the concurrent execution setting

■ When concurrent execution is enabled



■ When concurrent execution is 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<sup>#</sup>.

#

When an execution generation is running, you can change the status of the next generation to *Skipped so not exe.* without waiting for the current execution generation to end. To do this, open the Define Details - [Start Condition] dialog box, and select **Does not stay with skip** in the **Pause exec. gen. for start cond.** area. For details, see [3.4.6 Retention of execution generations with satisfied start conditions](#).



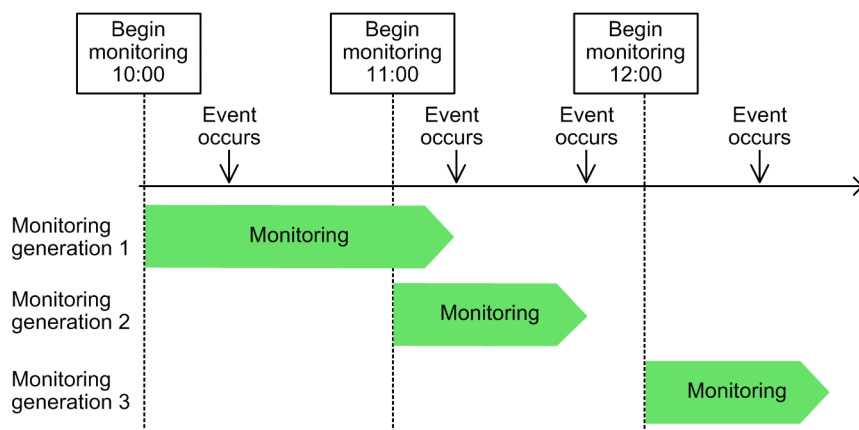
To enable the concurrent execution of execution generations, use the Define Details - [Jobnet] dialog box (for a root jobnet) or the Define Details - [Remote Jobnet] dialog box (for a root remote jobnet). For details, see *12.3.8 Define Details - [Jobnet] dialog box (for a root jobnet)* in the *JP1/Automatic Job Management System 3 Operator's Guide* or *12.3.10 Define Details - [Remote Jobnet] dialog box (for a root remote jobnet)* in the *JP1/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 3–53: Example of the concurrent execution of monitoring generations

Number of executions: 2 times



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.

#### (1) Setting the concurrent execution of monitoring generations

You can select one of 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 *12.4.5 Define Details - [Start Condition] dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

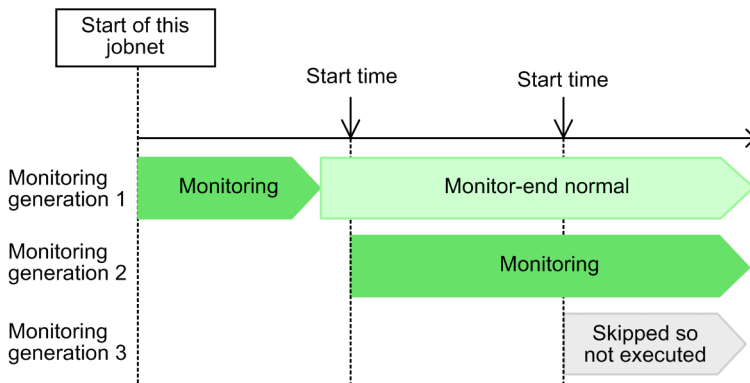
These behaviors are further described below.

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

## (2) 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 3–54: Behavior with "Do not start monitoring" defined in the jobnet's start condition

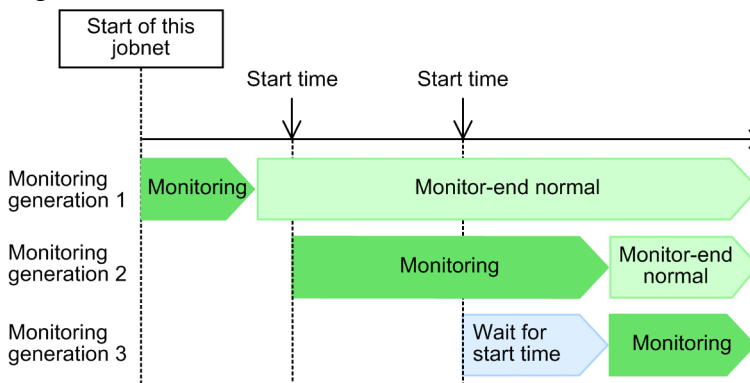


If the next start time arrives while monitoring generation 2 is in *Now monitoring* status, monitoring generation 3 enters *Skipped so not executed* status and does not start. We recommend that you select this behavior for a jobnet that has a set processing cycle and a start condition with an unlimited valid range.

## (3) 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 3–55: Behavior with "Wait for the end of monitoring" defined in the jobnet's start condition



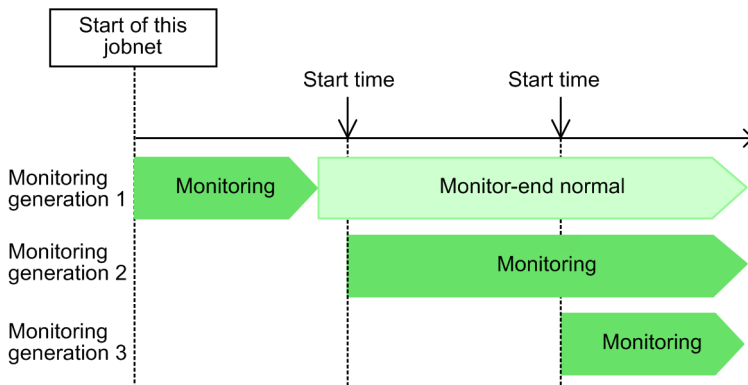
If the next start time arrives while monitoring generation 2 is in *Now monitoring* status, monitoring generation 3 enters *Wait for start time* status. When monitoring generation 2 ends, monitoring generation 3 moves from *Wait for start time* to *Now monitoring* status.

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

## (4) Behavior with the setting "Start monitoring"

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

Figure 3–56: 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.

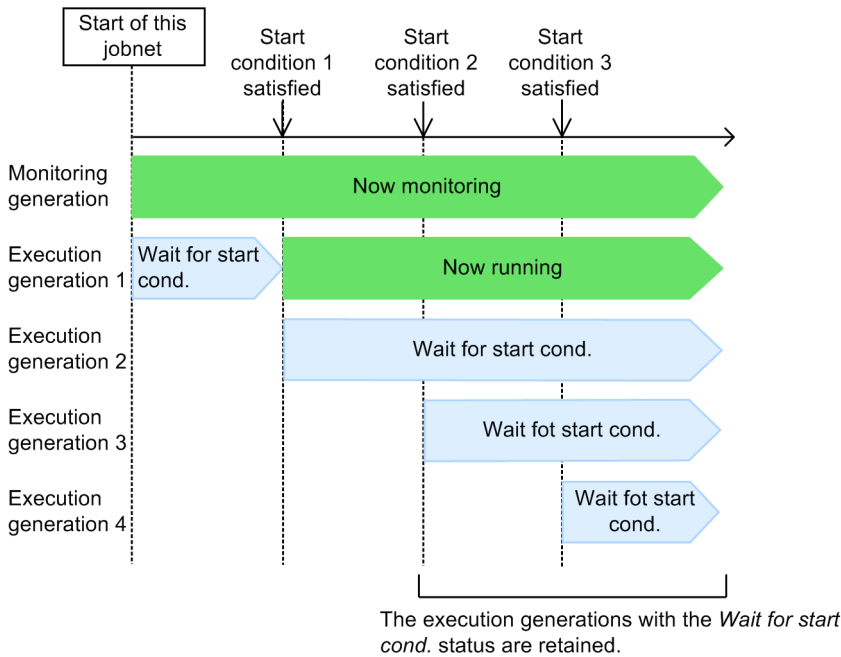
If the monitored event occurs while monitoring generations 2 and 3 are both active, each monitoring generation triggers an execution generation.

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 3–57: 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.

## (1) Setting the retention of execution generations

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 *12.4.5 Define Details - [Start Condition] dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

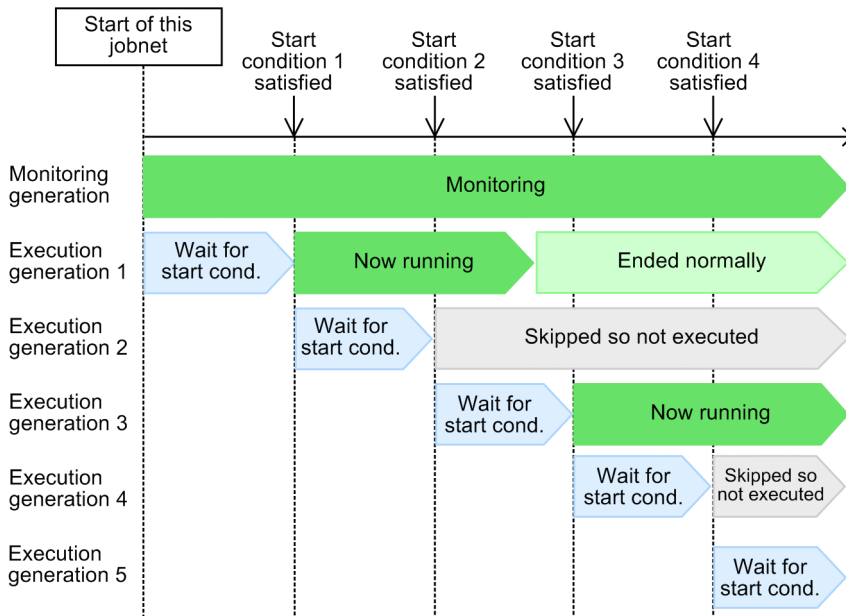
These behaviors are further described below.

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.

## (2) 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 3–58: 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.

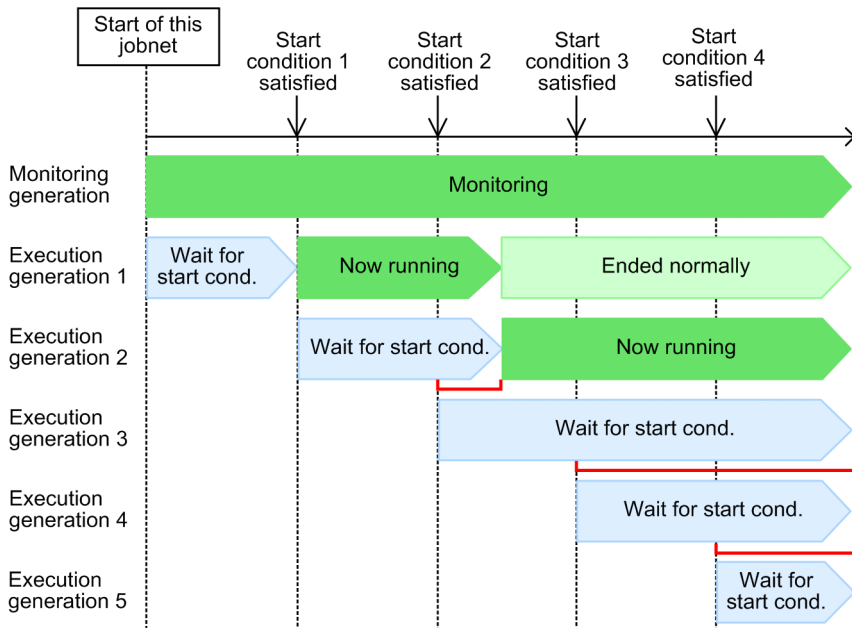
Cautionary notes

- 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.

### (3) 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 3–59: Behavior when "Stay without skip." is selected



Legend:

 : Start condition satisfied

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.

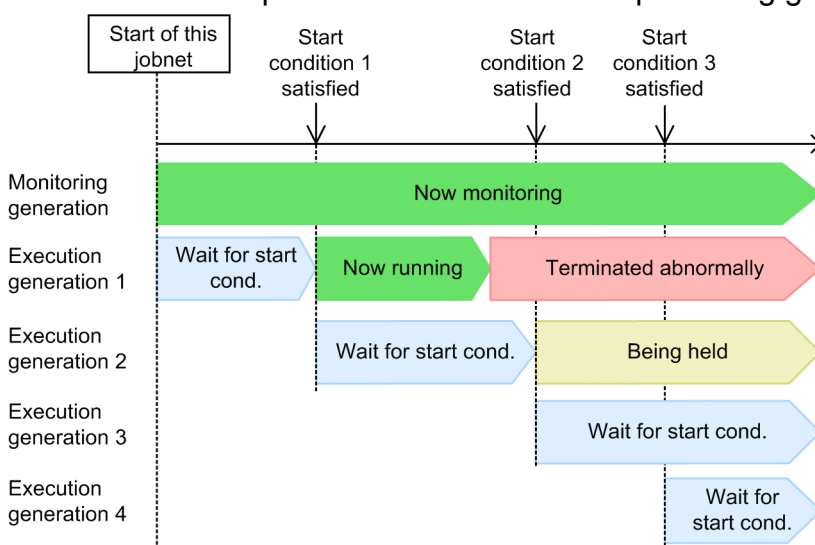
### (a) When the concurrent execution of execution generations is disabled

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 3–60: 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

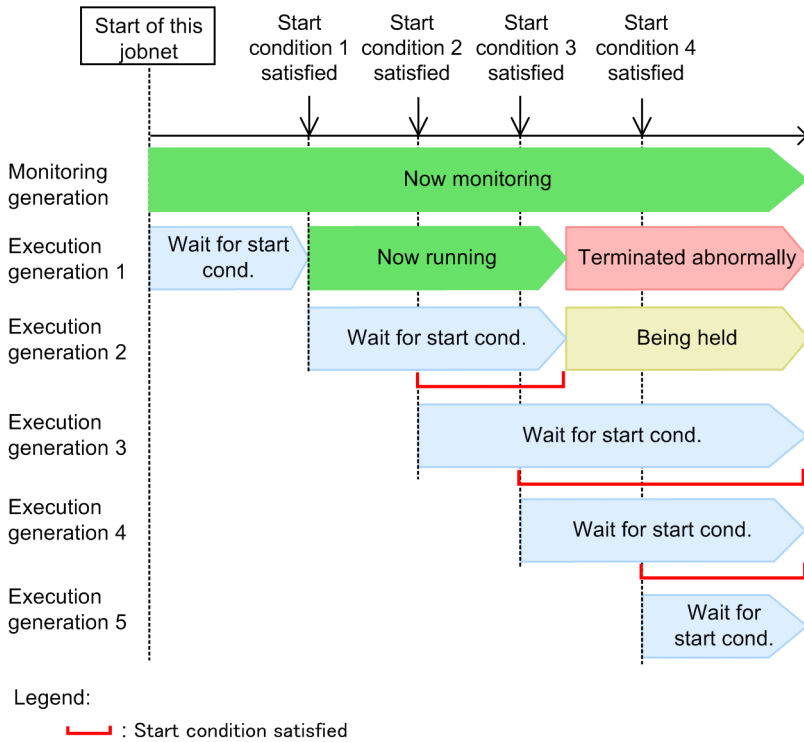


In this example, execution generation 1 terminates abnormally. Subsequently, start condition 2 is satisfied, triggering generation 2, which moves from *Wait for start condition* to *Being held* status. Start condition 3 is subsequently satisfied, but because the preceding generation (generation 2) has 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 3–61: 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



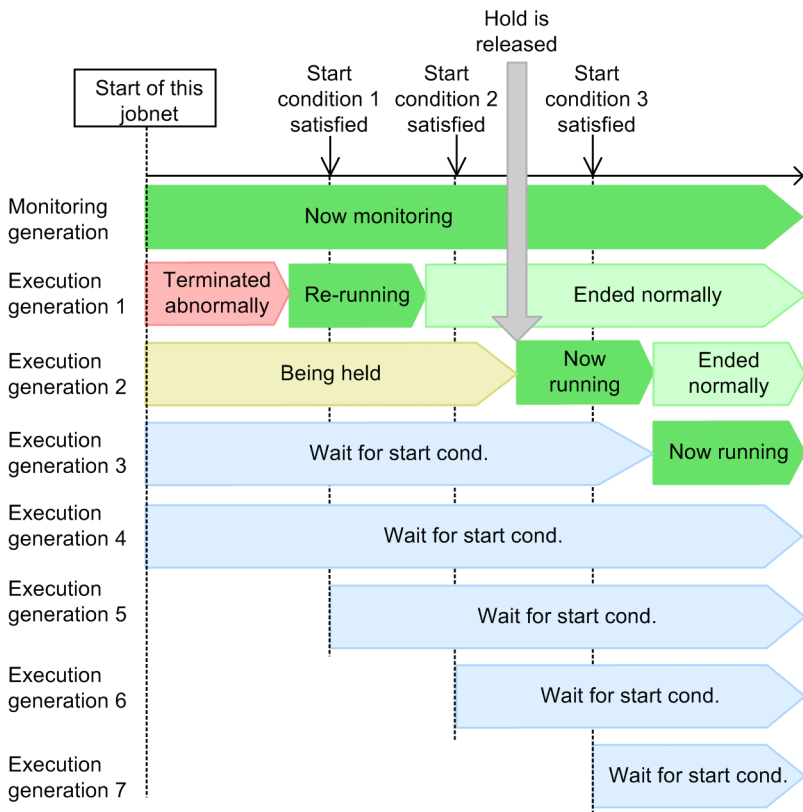
In this example, start conditions 2 and 3 are already satisfied when generation 1 terminates abnormally. Generation 2 is to be executed next. It moves from *Wait for start condition* to *Being held* status. Because the preceding generation (generation 2) has not finished, the succeeding generation 3 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 3–62: 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.

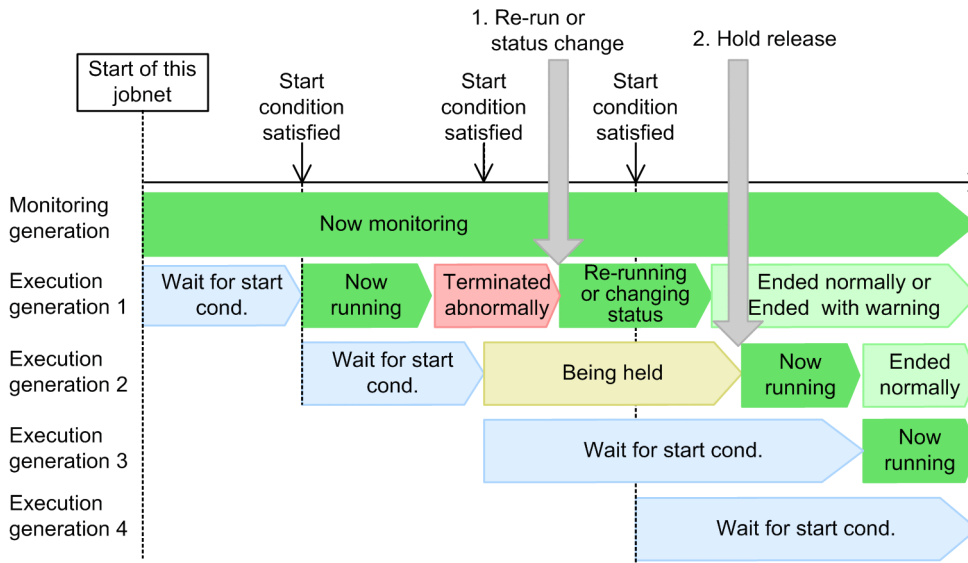
### ■ Procedure to resume operation

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.

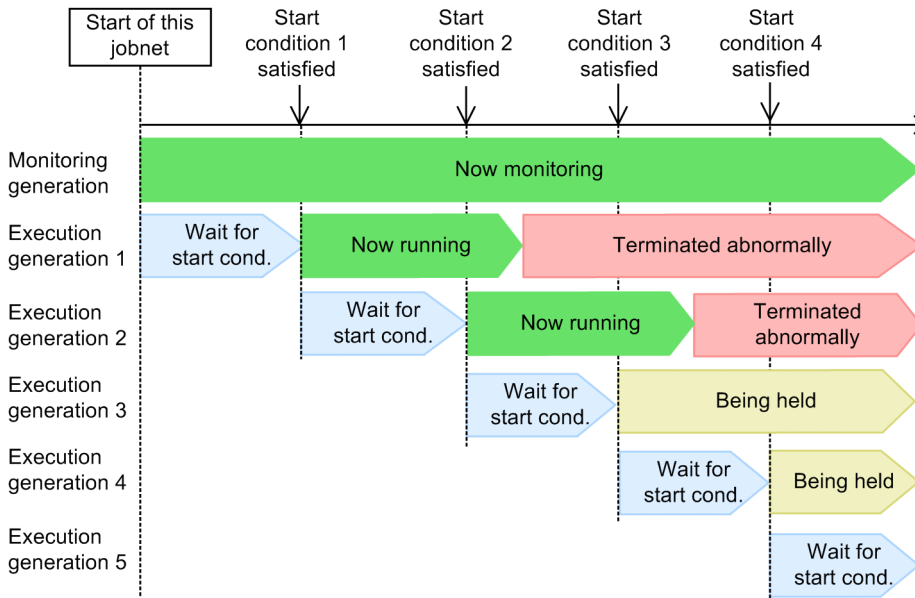
Figure 3–63: Resuming operation when concurrent execution is disabled and "Hold start of jobnet" is set



**(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 3–64: Effect of the "Hold start of jobnet" option when concurrent execution is enabled



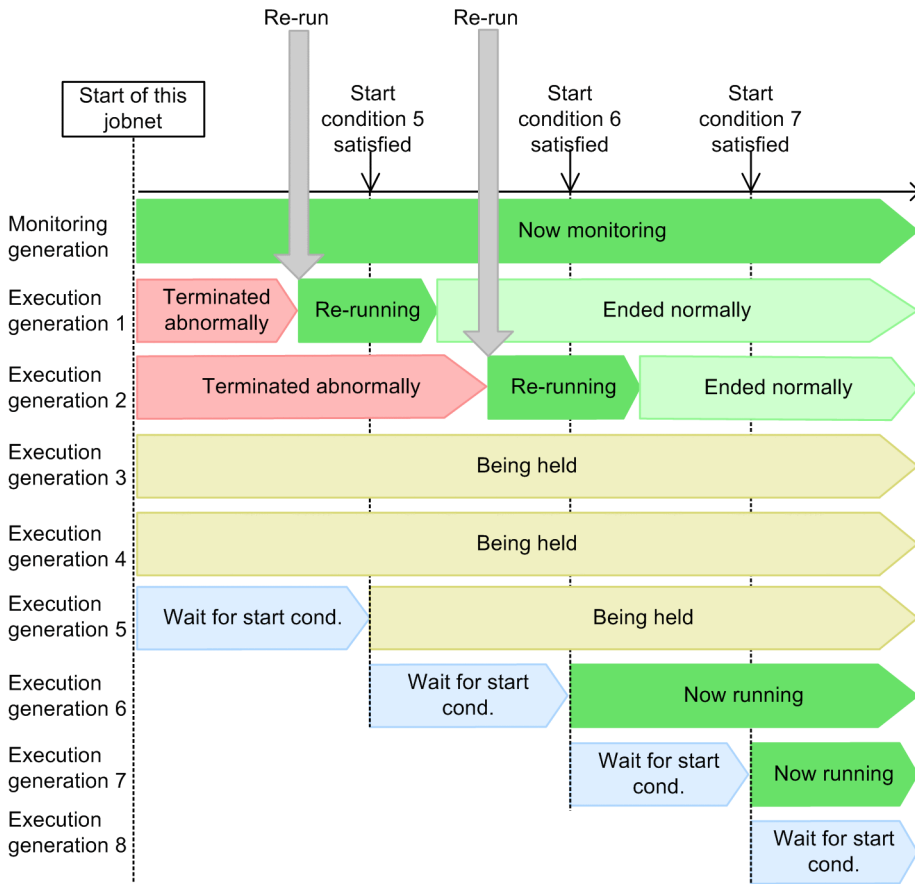
In this example, generation 2 runs concurrently with generation 1 and continues to run when execution generation 1 terminates abnormally. After generation 1 terminates abnormally, start conditions 3 and 4 are satisfied, triggering generations 3 and 4, which move from *Wait for start condition* status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

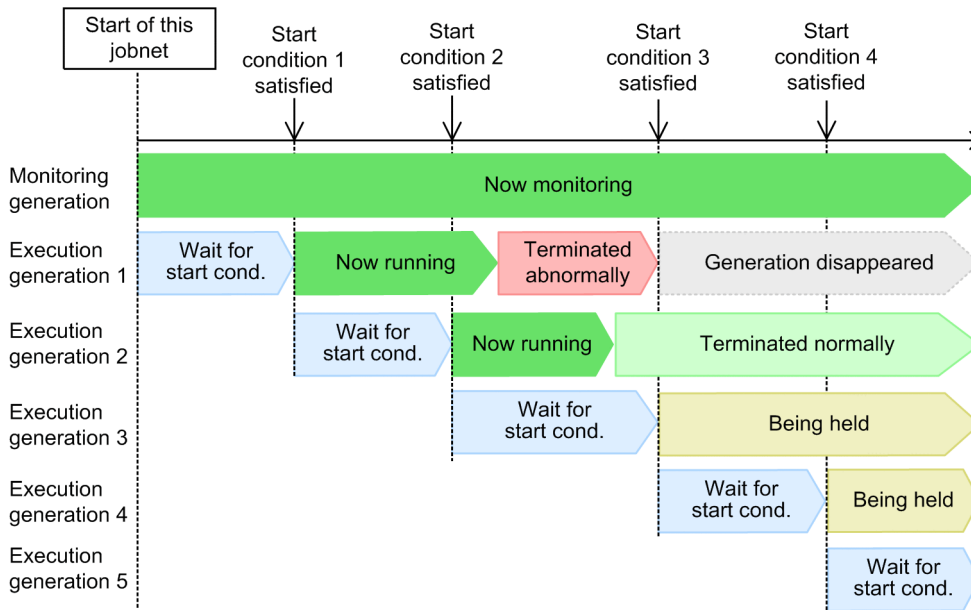
Figure 3–65: Timing of the transition from "Wait for start condition" to "Now running" status



### 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 3–66: Suppression behavior when an abnormally terminated generation is deleted



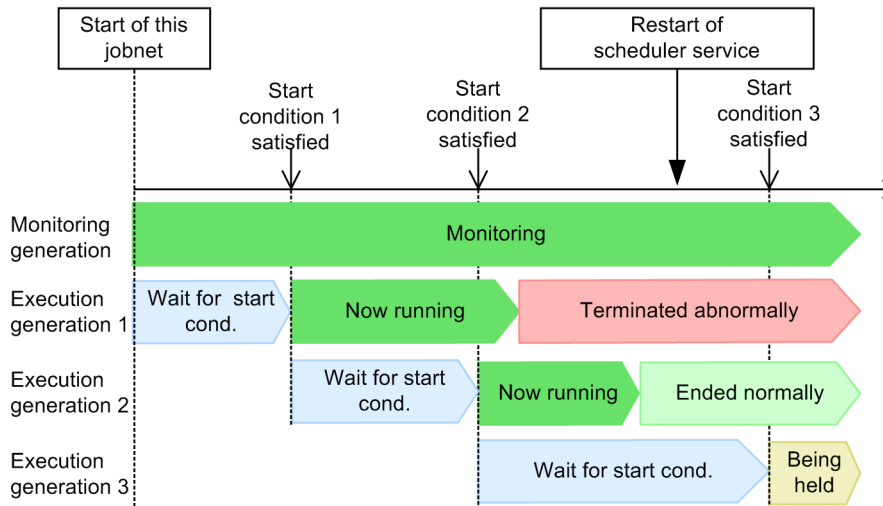
#### When the scheduler service is restarted

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.

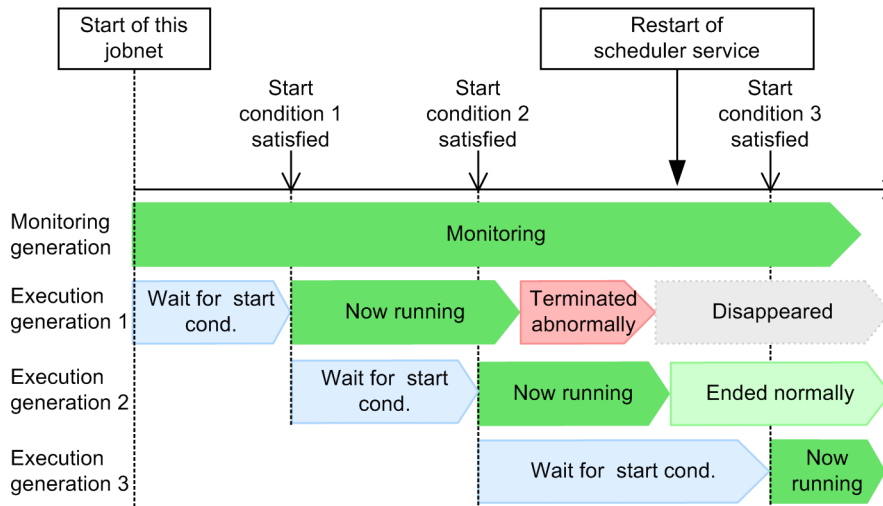
The following figure shows the effect of the **Hold start of jobnet** option when the scheduler service is restarted.

Figure 3–67: 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.

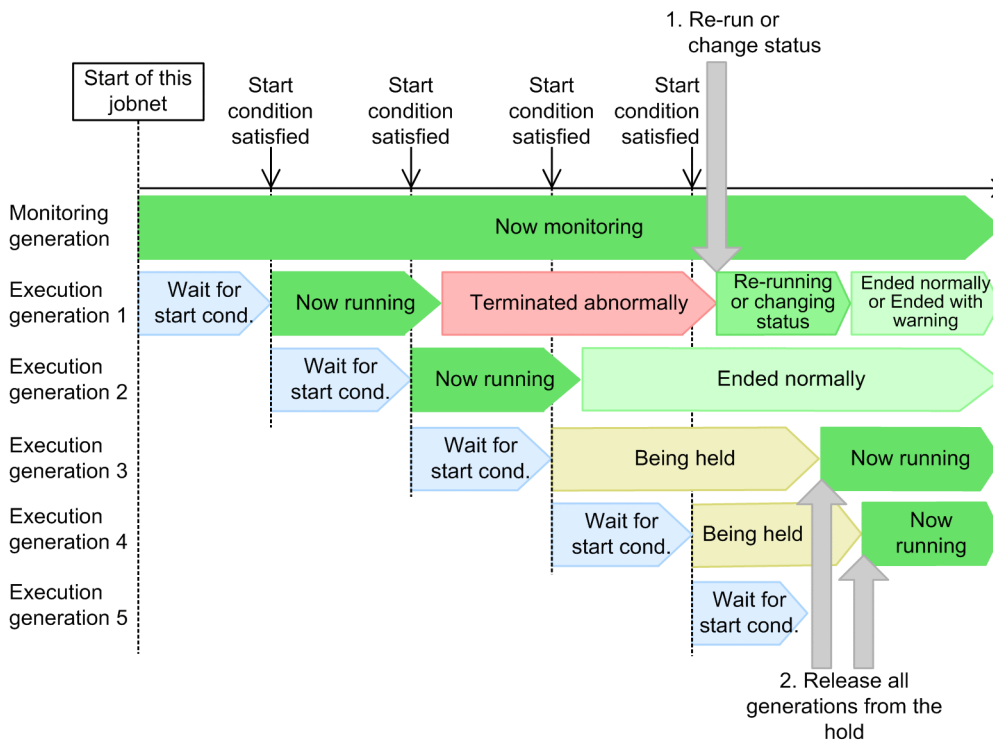
If execution generation 1 disappears because of the specified value for the number of logs to keep when the scheduler service is restarted, execution generation 3 moves from the *Wait for start condition* status to the *Now running* status when start condition 3 is satisfied.

■ 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 3–68: 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.

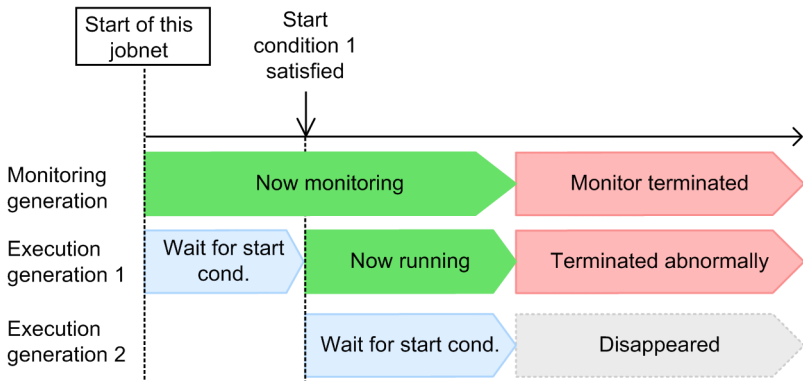
### (a) When the concurrent execution of execution generations is disabled

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 3–69: 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

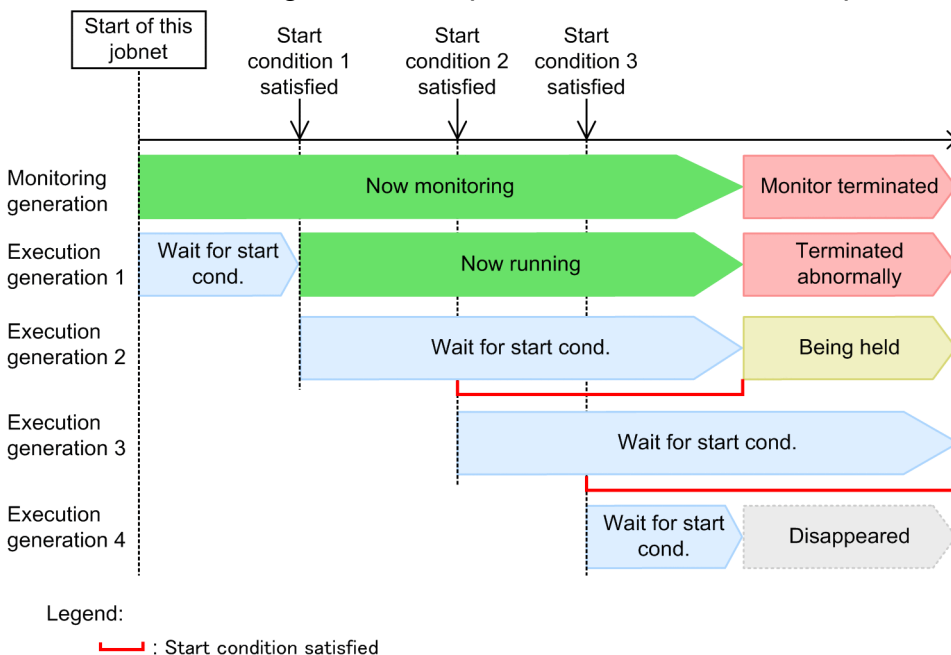


In this example, when execution generation 1 terminates abnormally, the monitoring generation moves from *Now monitoring* to *Monitor terminated* status. The waiting execution generation 2 disappears from the schedule.

■ **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 3–70: 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.

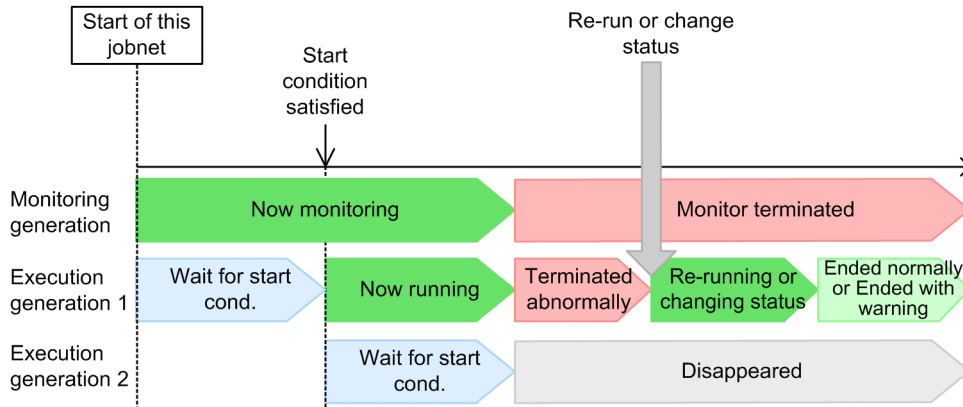
### ■ Procedure to resume operation

To resume operation:

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

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*.

Figure 3–71: 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

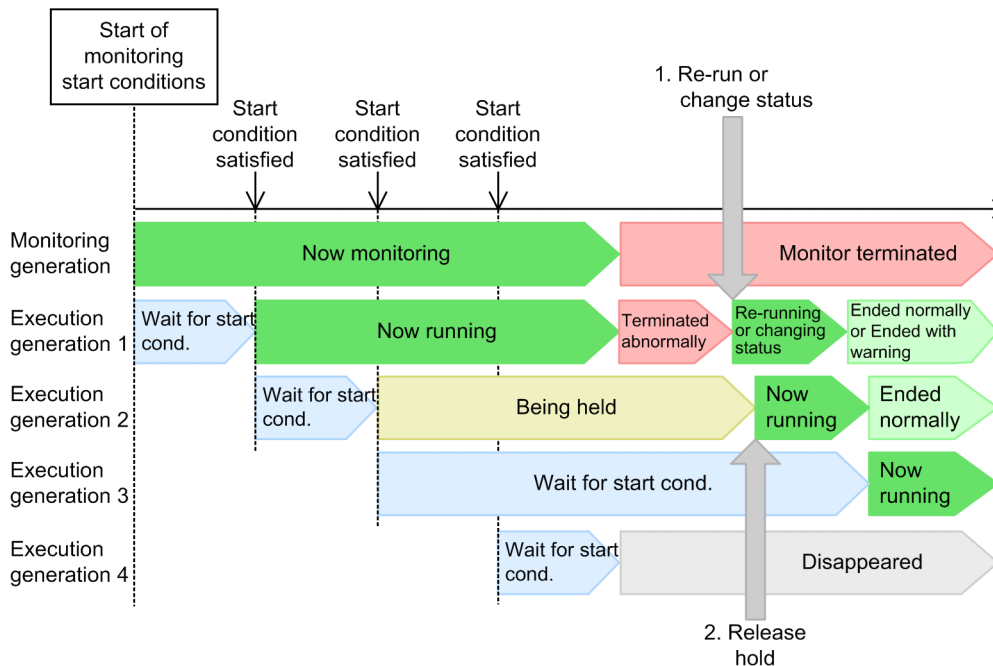
1. Resolve the problem, and then either rerun the abnormally terminated execution generation or change its job status to *Ended normally* or *Ended with warning*.

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.



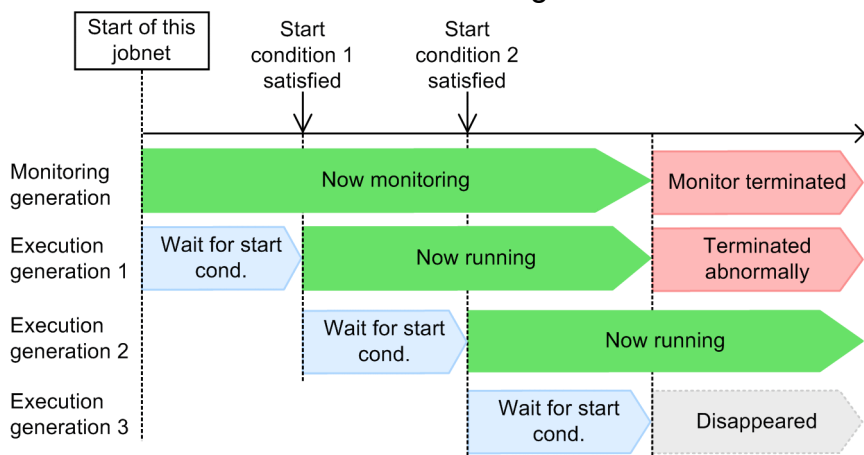
Figure 3–72: Resuming operation when concurrent execution is disabled and "Stop monitoring of start conditions" is set (when succeeding generations are waiting for the completion of execution of the preceding generation)



**(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 3–73: Effect of the Stop execution monitoring of start conditions option when the concurrent execution of execution generations is enabled

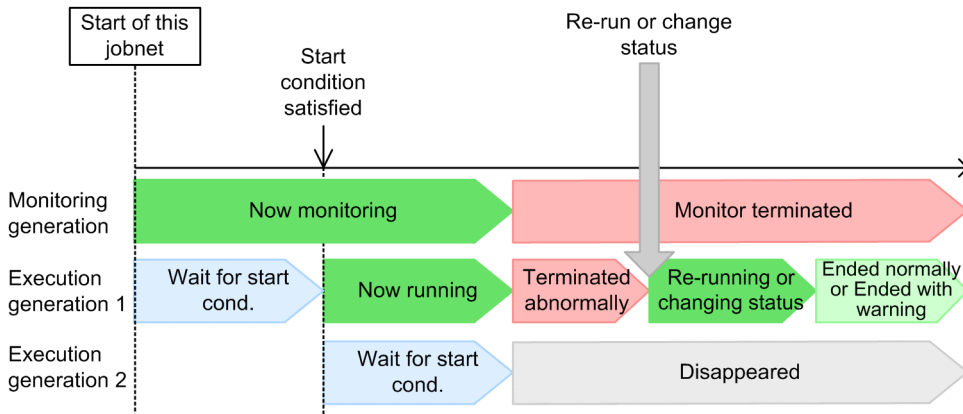


In this example, when execution generation 1 terminates abnormally, the monitoring generation moves from *Now monitoring* to *Monitor terminated* status. Generation 2, which is running concurrently with the abnormally terminated generation, continues execution. The waiting generation 3 disappears from 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 3–74: 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<sup>#</sup> are handled as abnormal termination:

- Ended abnormally
- Invalid-seq.
- Interrupt
- Killed

#

If the result of an execution generation is in *Skipped so not exe.* status, execution of the next execution generation is suppressed. However, the monitoring generation is not placed in *Monitor terminated* status, and continues monitoring.

- When an execution generation 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.

For the monitoring generations of a jobnet whose start conditions are being monitored, you can interrupt or forcibly terminate monitoring. For the execution generations of a jobnet that have start conditions, you can hold or cancel the execution.

Hold, cancellation, and other temporary operations performed on the monitoring generations of a jobnet are applied to the execution generations that will be generated after the operations are performed. These operations have no effect on

any execution generations that have already been generated. If you want to perform these temporary operations for existing execution generations, you need to explicitly specify the execution generations when you perform the operations.

The following shows examples of operations that can be performed for monitoring generations and execution generations.

#### 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 *8.1.1 Stopping a jobnet that is monitoring start conditions* in the *JP1/Automatic Job Management System 3 Administration Guide*.

#### Monitor a start condition again after monitoring has finished

You can restart monitoring of the start condition by performing one of the following:

- Re-registering for execution the jobnet for which monitoring has finished
- Temporarily changing the execution start time of a root jobnet of a scheduled generation
- Executing immediately a jobnet of a scheduled generation

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 change the temporary plans of the scheduled generation, see *9.3.1 Changing the execution start time of a jobnet* in the *JP1/Automatic Job Management System 3 Operator's Guide*, or *9.4 Immediately executing the execution schedule of a jobnet* in the *JP1/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 *JP1/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 *JP1/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 *8.1.2 Changing an event job definition in the start conditions of a jobnet registered for execution* in the *JP1/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 the execution results for a set number of generations in JP1/AJS3 - View or the Web GUI, or by using the API or the `ajsshows` 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\(3\) Examples of managing saved generations for 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 [20.4 Setting up the scheduler service environment in the JP1/Automatic Job Management System 3 Configuration Guide](#).

#### 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\(3\)\(b\) Saved-generation management by using the individual management method](#).

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\(4\) Notes applying when JP1/AJS3 has been upgraded from JP1 Version 8 or an earlier version of JP1/AJS2](#).

# 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

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>
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>

Registration method	Features
Planned execution	<p>Examples of usage:</p> <p>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.</p> <p>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:</p> <p>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:</p> <p>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):</p> <p>Temporary changes can be made to a fixed schedule.</p> <p>Adding execution dates (from a menu or command):</p> <p>Execution dates can be added.</p> <p>Examples of usage:</p> <p>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. To register jobnets for execution, use JP1/AJS3 - View, the Web GUI, the API, or commands. For details about execution registration procedures, see the relevant sections in the following manuals:

- *7. Executing Jobnets* in the *JP1/Automatic Job Management System 3 Operator's Guide*
- *A.18 Register for Execution dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*
- *7.1.4 Planned execution registration API* in the manual *JP1/Automatic Job Management System 3 Command Reference*
- *7.1.5 Fixed execution registration API* in the manual *JP1/Automatic Job Management System 3 Command Reference*
- *7.1.6 Immediate execution registration API* in the manual *JP1/Automatic Job Management System 3 Command Reference*
- *ajsentry* in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*

For details on redefining or rescheduling a registered jobnet, see [4.5 Operations on registered jobnets](#).

#### Supplementary notes

- If a jobnet for which a planned execution was performed has the status *Not sched. to exe.*, an execution schedule with the status *Not sched. to exe.* is created as the next execution schedule. However, if a jobnet for which immediate execution or fixed execution was performed has the status *Not sched. to exe.*, an execution schedule with the status *Not sched. to exe.* is not created.
- When you re-register a jobnet that has been registered for execution, the execution-registration type of the newly-created execution schedule is the type that you select during that re-registration process. However, a jobnet that was registered for planned execution can be re-registered for execution when the status of the jobnet is *Not sched. to exe.* If you do this, the execution schedule with the *Not sched. to exe.* status is deleted, and an execution schedule that has the registration type that you selected during the re-registration process is created.

## (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.

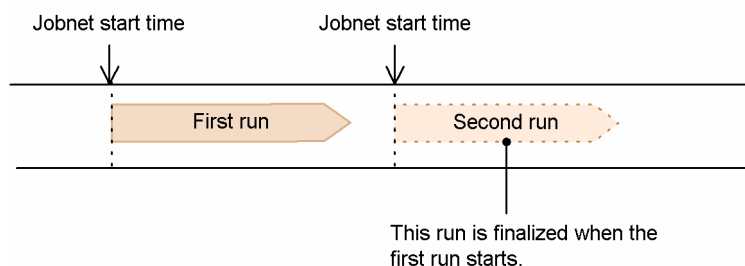
## (2) Planned execution

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

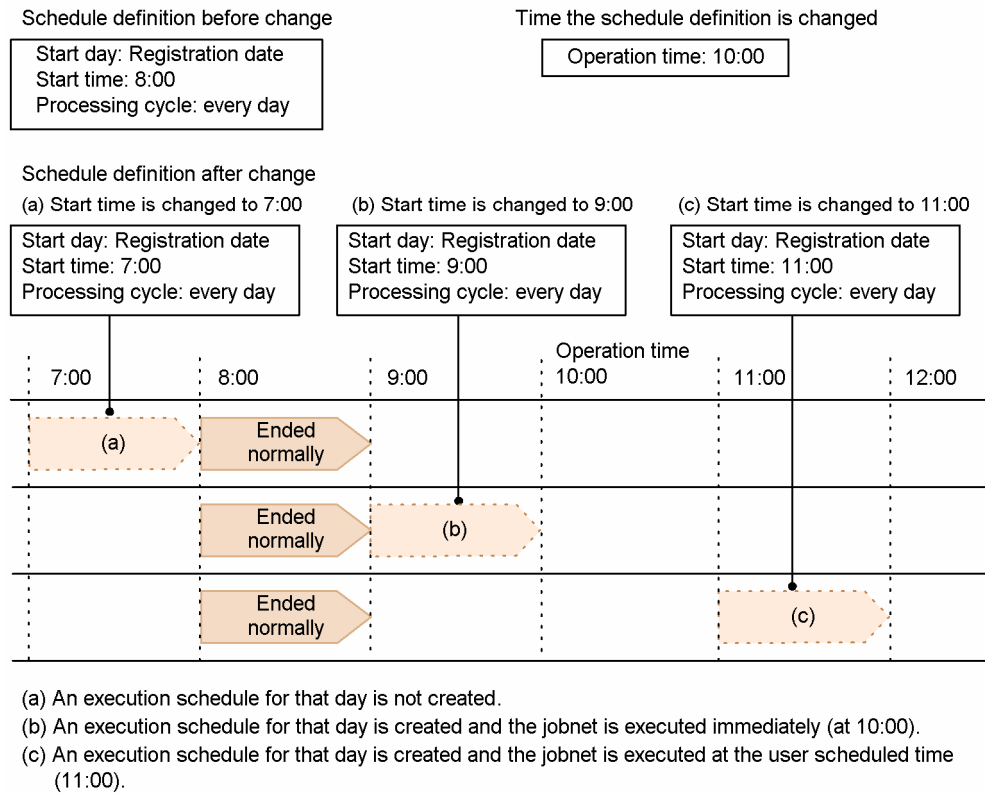
Do not change the base time of a jobnet that has been registered for execution because it would complicate the schedule calculations. Also, depending on what time you set and when you make the change, the jobnet might be scheduled on that day and execute immediately. If you need to change the base time, 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



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.

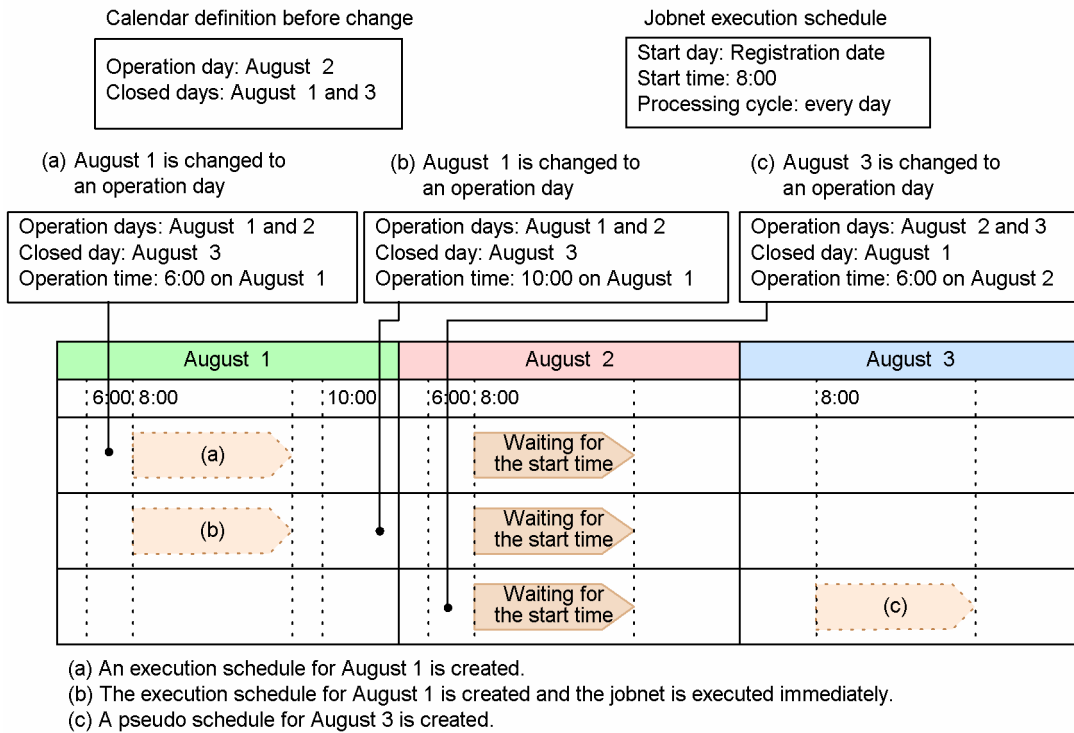
(c) Start time is changed to 11:00

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



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

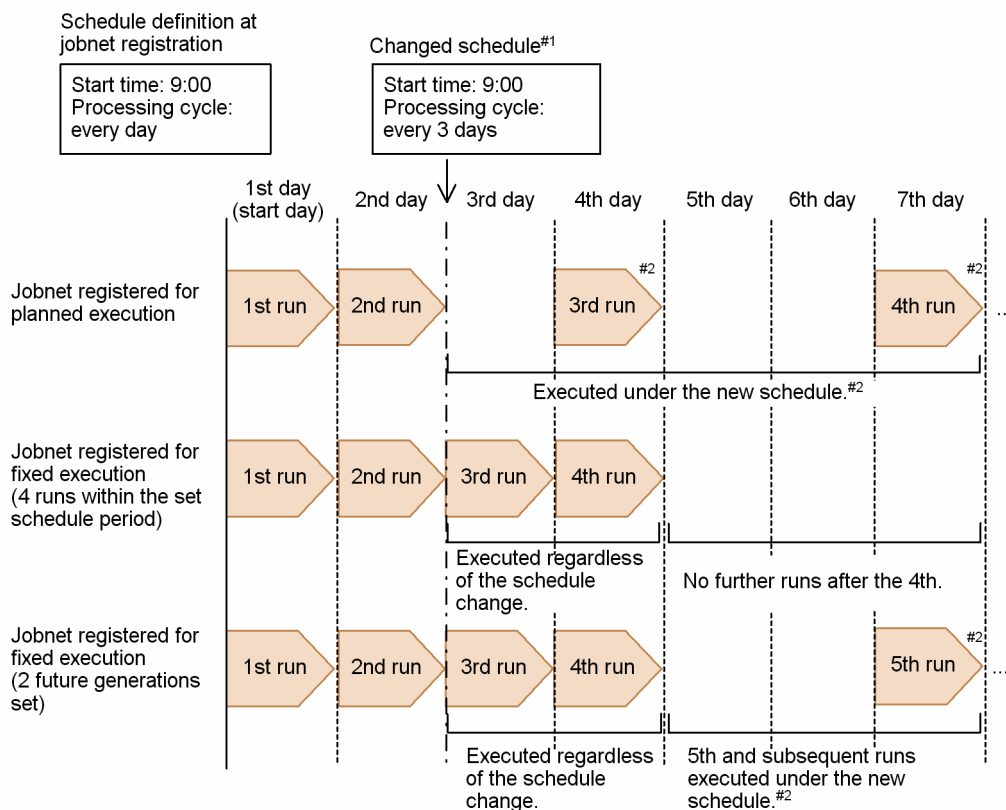
JP1/AJS3 adds a schedule for a jobnet to be executed on the specified date and time, regardless of the schedule definition for the jobnet.

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.

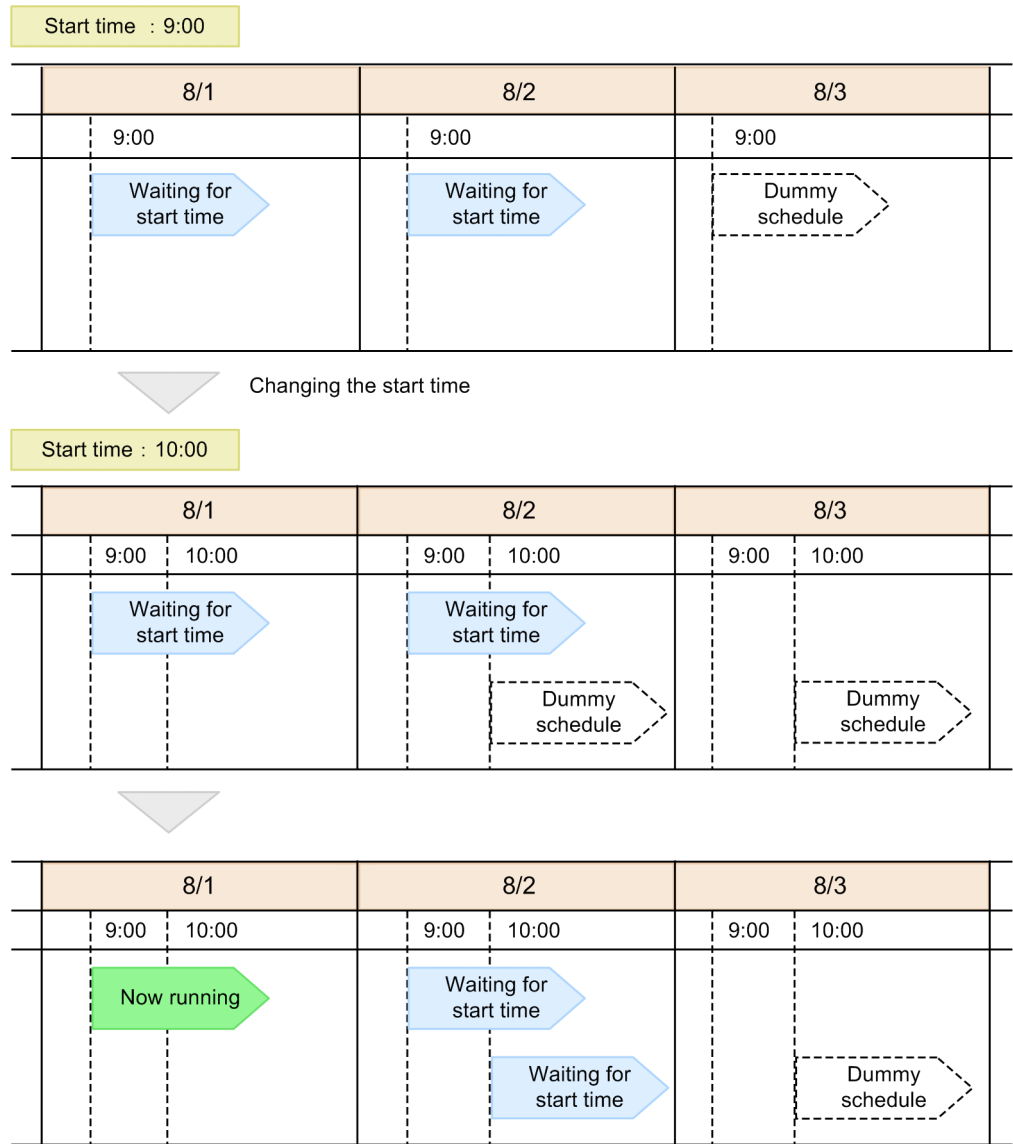
When fixed execution is specified by generation count, JP1/AJS3 continues fixing each succeeding execution schedule up to the specified number. When the first generation starts running, a new execution schedule is generated (fixed). In the above example, 2 is set in **Future generation**. This means 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 notes

- 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.
- Suppose that you register a jobnet for fixed execution by specifying the number of future generations, and then you change the schedule definition without releasing the jobnet registration. Dummy schedules are created with the changed schedule definition starting from a date and time later than the start time of the fixed execution schedule. Accordingly, after you change the start time, both the execution schedule with the pre-change start time and the execution schedule with the post-change start time might be performed on the same day. If you change the start time, also specify the start year, month, and day, and make sure that the execution schedule is specified as you intended. If an execution schedule or dummy schedule that does not need to be executed exists, cancel it.

The following figure shows an example of changing the start time of the schedule definition.

Figure 4–5: Example of changing the start time of the schedule definition



In this example, a jobnet is registered for fixed execution with 2 specified as the number of future generations, and the execution schedules for 9:00 on August 1 and for 9:00 on August 2 are fixed.

Suppose that, before 9:00 on August 1, you change the start time of the jobnet from 9:00 to 10:00. Because the execution schedules for 9:00 on August 1 and 9:00 on August 2 are fixed, the change to the schedule rule is not applied. However, a dummy schedule is created at 10:00 on August 2, which is later than the start time of the fixed execution schedule.

After the schedule for 9:00 on August 1 is executed, the dummy schedule for 10:00 on August 2 becomes a fixed schedule because the number of future generations is set to 2.

When you change the schedule so that the jobnet is executed at 9:00 until August 2 and at 10:00 then from August 3, set **Start year and month** and **Start day** of the schedule rule to August 3, and set **Start time** to 10:00.

- 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, and then re-register the jobnet for fixed execution.
- Use the same JP1 user when registering a jobnet for fixed registration that specifies both number of future generations and then registering a jobnet for fixed registration that specifies a fixed period or time.

When the number of future generations is specified, new execution schedules will be generated when, for example, jobnet execution starts. When a new execution schedule is generated, the registering entity in the

previous execution result becomes the registering entity in the new execution schedule. Accordingly, if different users specify the number of generations and a period or time, the registering entity in a newly generated execution schedule might change depending on who registered the previous execution result.

For details about the registering entity (*user who registered*), see [7.2.3 Setting the Executed by attribute](#).

- After the status of a jobnet that is registered for fixed execution and that has the number of future generations specified changes to *Not sched. to exe.*, if you register the jobnet for fixed execution by specifying a fixed schedule period or by specifying a specific date and time, the specified number of future generations is no longer in effect and the jobnet is treated as a jobnet that is registered for fixed execution with either a fixed schedule period or a specific date and time specified.
- After the status of a jobnet that is registered for fixed execution and that has the number of future generations specified changes to *Not sched. to exe.*, if you change the schedule definition or calendar definition, no execution schedule is created. To change the schedule definition or calendar definition to create an execution schedule, re-register the jobnet for 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, JP1/AJS3 - View, the Web GUI, or the API, a large amount of memory might be needed under certain conditions. For example, a large amount of memory is needed if the jobnet for fixed execution contains many units or 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 reschedule 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.

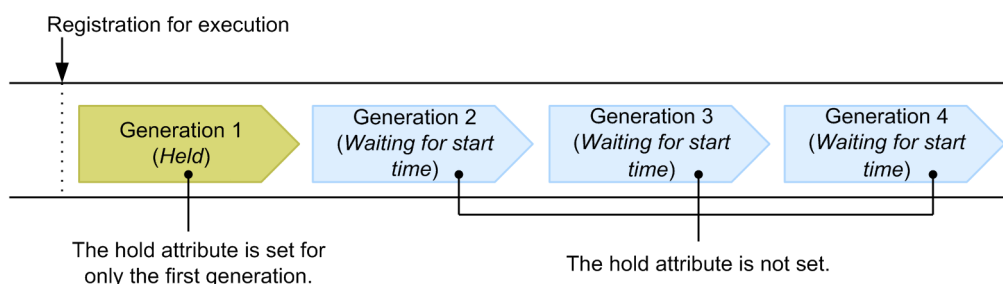
For details on the behavior of a jobnet with a start condition after it is registered, see [3.4 Defining a start condition](#).

## (5) Setting a hold during registration for execution

You can set whether to hold the execution of a root jobnet when registering the root jobnet for execution. If you set a hold during registration for execution, you can hold the execution of the root jobnet that you register, even if a hold is not defined.

If you set a hold during registration for execution, the hold attribute is set for the first of the generations created during registration for execution. The following figure shows the generation for which the hold attribute is set during registration for planned execution or for fixed execution.

Figure 4–6: Generation for which the hold attribute is set during registration for planned execution or for fixed execution

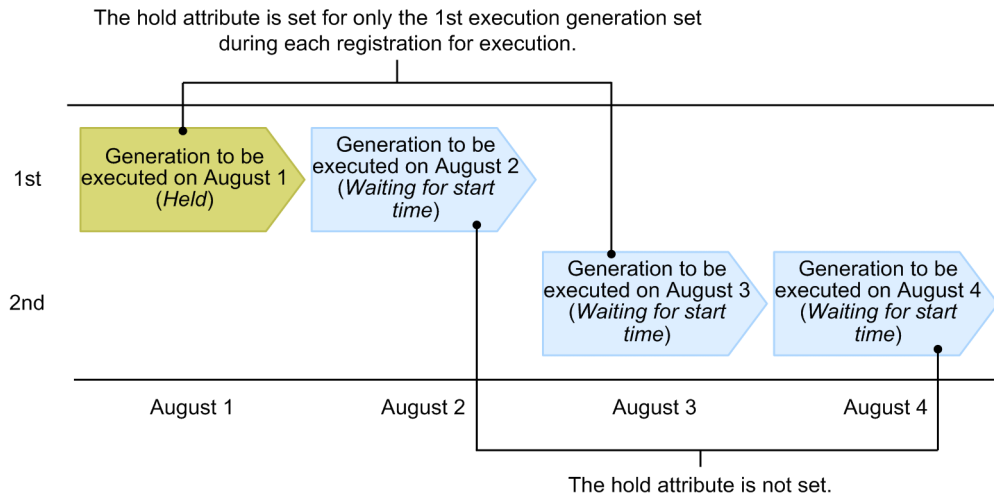


If you perform more than one registration for fixed execution for which the execution period is specified, the hold attribute is set for only the first generation created for each registration for execution. The following figure shows the generations for which the hold attribute is set when more than one registration for fixed execution for which the execution period is specified is performed.

**Figure 4–7: Generations for which the hold attribute is set when more than one registration for fixed execution for which the execution period is set is performed**

This example shows the generations for which the hold attribute is set in the following situation:

- August 1 to 2 is specified as the period during the 1st registration for fixed execution.
- August 3 to 4 is specified as the period during the 2nd registration for fixed execution.

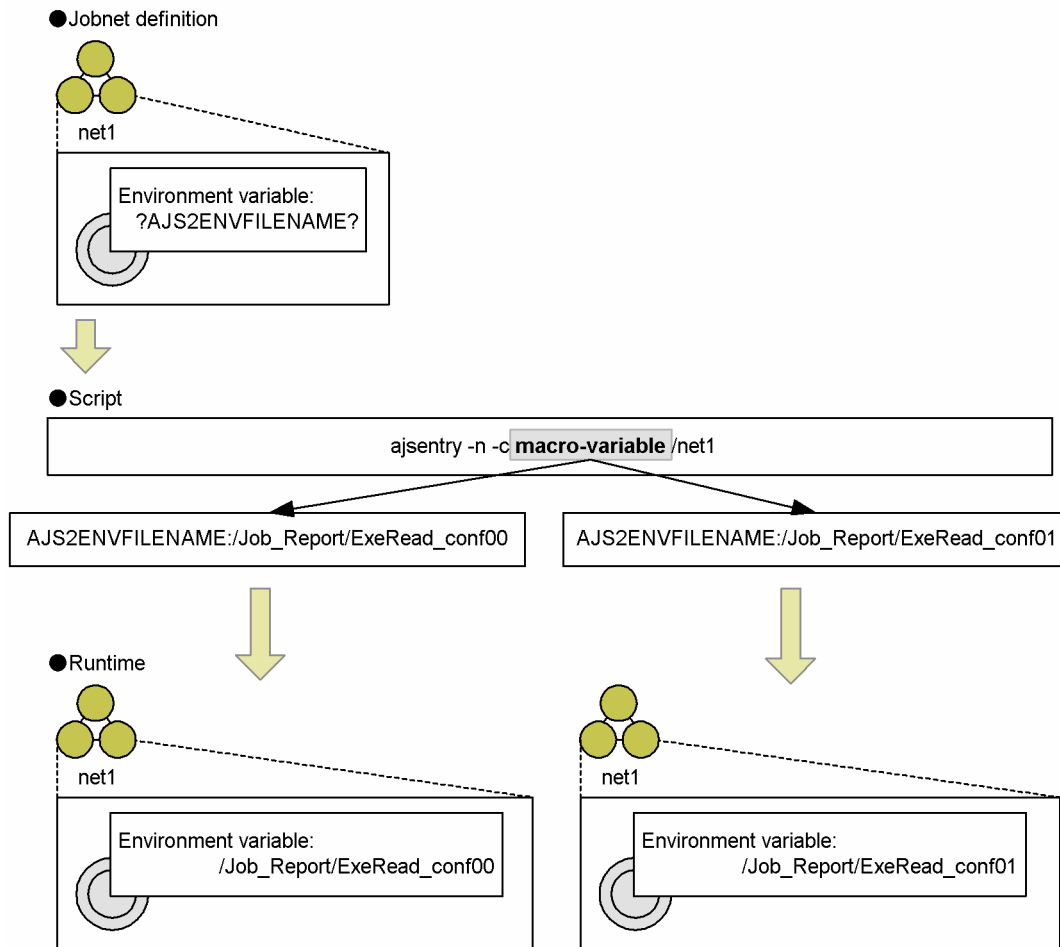


### 4.1.2 Specifying macro variables during registration for execution

During registration for execution, you can specify macro variable names and their values (passing information). If the specified macro variable names 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 (`ajschang` command), and registering the jobnet for execution (`ajsenry` 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 shows how information is passed via a macro variable specified during registration for execution.



Figure 4–8: How information is passed via a macro variable specified during registration for execution



To specifying macro variables during registration for execution, use JP1/AJS3 - View, the Web GUI, the API, and the `ajsentry` commands. For details about how to specify the value of a macro variable, see the relevant sections in the following manuals:

- *7.2 Specifying macro variables during registration for execution in the JP1/Automatic Job Management System 3 Operator's Guide*
- *A.19 Register Passing Information dialog box in the JP1/Automatic Job Management System 3 Operator's Guide*
- *7.1.4 Planned execution registration API in the manual JP1/Automatic Job Management System 3 Command Reference*
- *7.1.5 Fixed execution registration API in the manual JP1/Automatic Job Management System 3 Command Reference*
- *7.1.6 Immediate execution registration API in the manual JP1/Automatic Job Management System 3 Command Reference*
- *ajsentry in 3. Commands Used for Normal Operations in the manual JP1/Automatic Job Management System 3 Command Reference.*

For details about macro variables see *2.2.6 Considerations regarding the use of macro variables in the JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide.*

## (1) Handling of the macro variable specified during registration for execution

The following describes how the macro variable specified during registration for execution is treated.

- Macro variable for each method of registration for execution  
Macro variable names and passing information can be used during registrations for planned execution, fixed execution, and immediate execution. The following table describes how the macro variable is treated for each registration method.

Table 4–2: How the macro variable is treated for each method of registration for execution

Method of registration for execution (option of the <code>ajsentry</code> command)		How the macro variable is treated
Registration for immediate execution ( <code>-n</code> )		The specified macro variable is applied.
Registration for planned execution ( <code>-s</code> )		The specified macro variable is applied to all the schedules. If you want to change the macro variable that was specified during registration for execution, cancel the registration, change the macro variable, and then re-execute registration.
Registration for fixed execution	Specifying a fixed period ( <code>-p</code> (alternatively, <code>-b</code> and <code>-e</code> ))	The specified macro variable 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 for each generation, divide the period and perform registration for each division. If you want to change the macro variable that was specified during registration for execution, cancel the registration for the desired period, change the macro variable, and then re-execute registration.
	Specifying the number of generations ( <code>-g</code> )	The specified macro variable is applied to all of the specified future generations of the jobnet. If you want to change the macro variable that was specified during registration for execution, cancel the registration, change the macro variable, and then re-execute registration.
	Addition ( <code>-d</code> )	The specified macro variable is applied to the execution at the specified date and time. If you want to change the macro variable that was specified during registration for execution, cancel the registration, change the macro variable, and then re-execute registration.

- Macro variables during re-execution  
The macro variable name and passing information specified during registration for execution are applied without change during re-execution also.
- Macro variables during a temporary plan change  
Macro variable names and passing information specified at the time of registration for execution continue to apply even during a temporary plan change.
- Macro variables with identical names  
If the macro variable name specified during registration for execution is specified in the preceding job, the passing information for the preceding job is used wherever information is passed from the preceding job. If the same macro variable name is specified in the start condition, the name specified in the start condition prevails.
- Macro variables when concurrent execution is enabled  
The macro variable specified during registration for execution is applied to all the generations of the jobnet, regardless of whether the jobnet allows concurrent execution.

## (2) Cluster system

Macro variables 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 variable names 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. A jobnet that has not been started is executed according to its schedule, and the macro variable name and value specified during registration for execution are applied. 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 variable names 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 variable names 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 variable names and values specified during registration for execution are applied.

## **(3) Cautionary Notes**

This subsection provides notes on specifying macro variables during registration for execution.

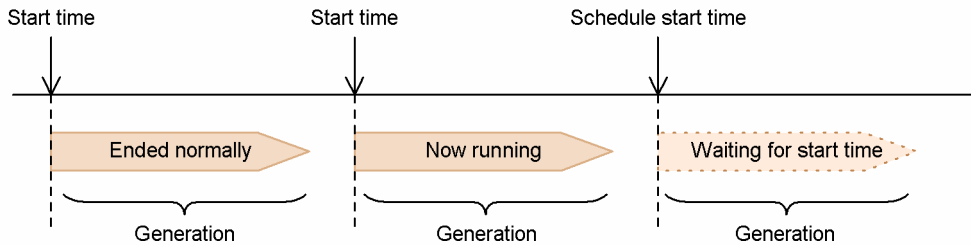
- If a macro variable is specified during registration for execution, the name and value of the macro variable are stored in a file on a root jobnet or generation basis. Since the file must be deleted when the registration for execution is canceled, specification of macro variables during registration 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 6.1.7 *Changing the mode in which unregistration or generation management deletes the generations of a jobnet* in the *JP1/Automatic Job Management System 3 Configuration Guide (Windows)* or 15.1.7 *Changing the mode in which unregistration or generation management deletes the generations of a jobnet* in the *JP1/Automatic Job Management System 3 Configuration Guide (UNIX)*.
- For a root jobnet that is a remote jobnet, macro variables cannot be specified during registration for execution. Also, for a remote jobnet that is a nested jobnet, macro variables 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 4–9: 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\(3\) Examples of managing saved generations for 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 4–10: 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 4–11: 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 4–12: 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

This subsection describes the number of logs to keep and saved-generation management for a jobnet.

#### (1) What is the number of logs to keep?

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.

A jobnet generation is created each time the jobnet is registered for execution, and can be saved as a log when the jobnet is executed. How many executed generations (including the one being executed) can be saved is determined by the number-of-logs-to-keep setting.

The setting applies only to a root jobnet. After you have 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 of JP1/AJS3 - View. You can also view the execution results in the Monthly Schedule window of the Web GUI, or by running the API function to obtain unit information, or by the `ajsshow` command. For details on the `ajsshow` command, see `ajsshow` in 3. *Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

The values you can set as the number of logs to keep are 1 to 99.<sup>#</sup> Because the amount of data saved in the database is proportional to *number of saved generations x number of registered units*, the more generations saved, the greater the load of database access operations is, such as canceling the registration of a jobnet. Before you increase the number of logs to keep, carefully consider the impact on system performance. For a jobnet consisting of many units, we recommend that you set a value less than 10 as the number of logs to keep. If unnecessary generations have been saved, cancel the registration of those generations and delete them manually by specifying dates. For details about canceling the registration of a jobnet, see 4.5.1 *Canceling registration of a jobnet*.

#

Although you can increase the upper limit on the number of logs to keep by using the `MAXSAVEGEN` environment setting parameter of the scheduler service, use of this parameter requires caution. For details about the `MAXSAVEGEN` environment setting parameter, see 20.4 *Setting up the scheduler service environment* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

## (2) Overview of saved-generation management

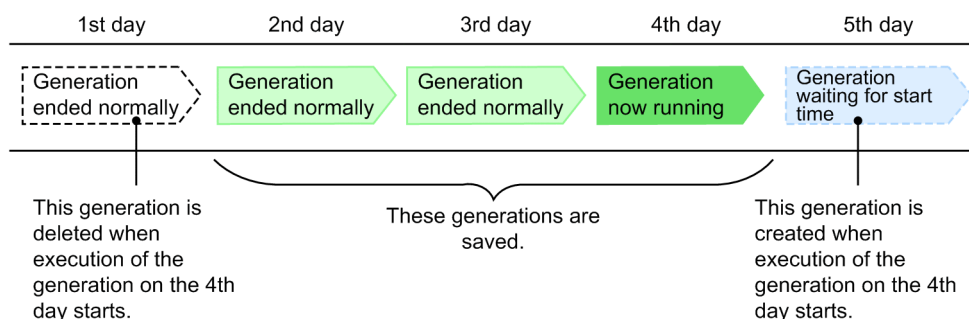
The generations of a jobnet are saved as logs after execution of the jobnet, and the number of saved generations is controlled so that it does not exceed the preset number of logs to keep. Controlling the number of saved generations so that it does not exceed the number of logs to keep is called *saved-generation management*.

Saved-generation management is performed when the number of generations to be managed increases (for example, when the next scheduled jobnet execution starts). If the number of managed generations exceeds the preset number of logs to keep, JP1/AJS3 automatically deletes saved generations starting with the oldest one.

The following figure shows how managed jobnet generations are managed.

Figure 4–13: How managed jobnet generations are managed

- Number of logs to keep: 3



In this example, because the number of logs to keep is set to 3, JP1/AJS3 controls the number of saved generations so that the total number of generations ending normally and the currently running generation does not exceed 3. Therefore,

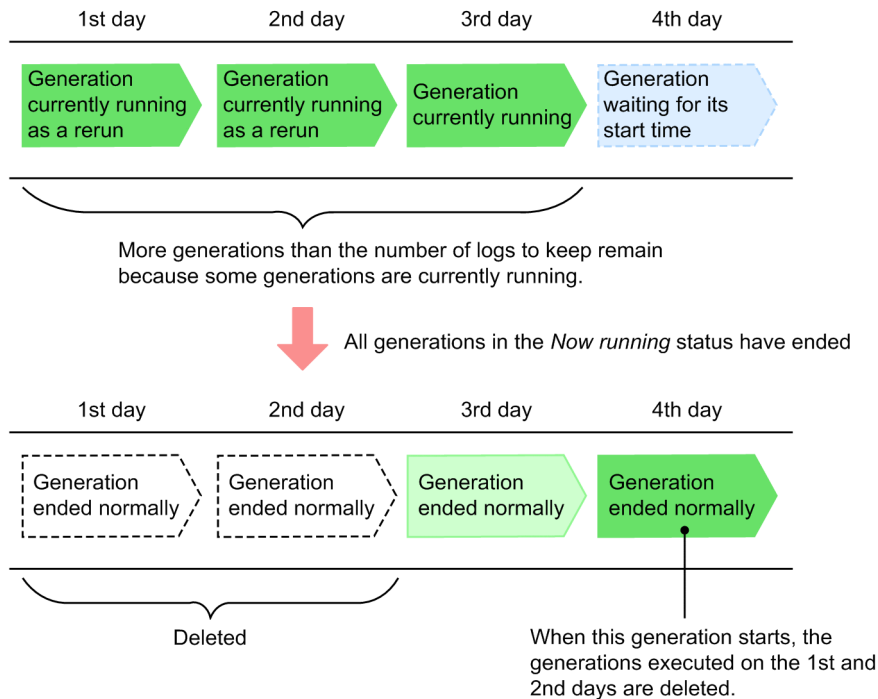
when a new jobnet generation starts on the fourth day, the generation executed on the first day is deleted to maintain the number of saved generations at 3.

Note, however, that because saved-generation management does not delete generations that have not ended (for example, generations in the *Now running* status), more generations than the number of logs to keep might be saved. If this happens, the excess generations are deleted collectively the next time saved generations are managed.

The following figure shows an example of how excess saved generations are deleted.

**Figure 4–14: Example of managing saved generations when more generations than the number of logs to keep are saved due to reruns**

- Number of logs to keep: 2 (Concurrent execution setting: *Enable*)



In this example, although the number of logs to keep is set to 2, generations whose status is in the *Now running* status are not deleted even when the number of logs to keep has been exceeded. Saved generations are managed when the next generation scheduled for execution starts, and the oldest generations are deleted so that two generations, including the one that has just started execution, remain.

### (3) Examples of managing saved generations for a jobnet with a start condition

For a jobnet with a start condition set, the following two methods can be used to manage saved generations:

- **Total management method**  
The setting of the number of logs to keep is applied to the total number of saved monitoring generations and saved execution generations.  
This is the same method that is used for jobnets with no start condition set.
- **Individual management method**  
The number-of-logs-to-keep setting is applied separately to the number of saved monitoring generations and to the number of saved execution generations.



In JP1 Version 8 and earlier, the individual management method is used to manage the number of saved generations.

The figure below shows the difference between the total management method and the individual management method.

In both methods, saved generations are managed when a new managed generation appears (when monitoring of start conditions starts or when the next execution generation is created because the start conditions have been satisfied).

Figure 4–15: Difference between the total management method and the individual management method



For the total management method, monitoring generations and execution generations are saved in such a way that the total number of these generations does not exceed the preset number of logs to keep. Therefore, as shown in the above example, if a jobnet whose start condition is satisfied 50 times a day is executed for 50 days, the number of saved generations is 50. This number is the same for the first day and the 50th day.

For the individual management method, the preset number of logs to keep is applied individually to monitoring generations and to execution generations. Therefore, the number of generations that will actually be saved becomes larger than the preset number of logs to keep. If a jobnet whose start condition is satisfied 50 times a day is executed for 50 days, 2,550 generations are saved.

## Supplementary note

In JP1 Version 8 and earlier versions, the individual management method was used for saved-generation management. However, in JP1 Version 9 and later versions, the total management method has been added and is selected by default.

If you want to set the number of logs to keep to 10 or more, we recommend that you use the total management method.

The following describes the processing performed for each method when the number of saved generations exceeds the number of logs to keep.

### (a) Saved-generation management by using the total management method

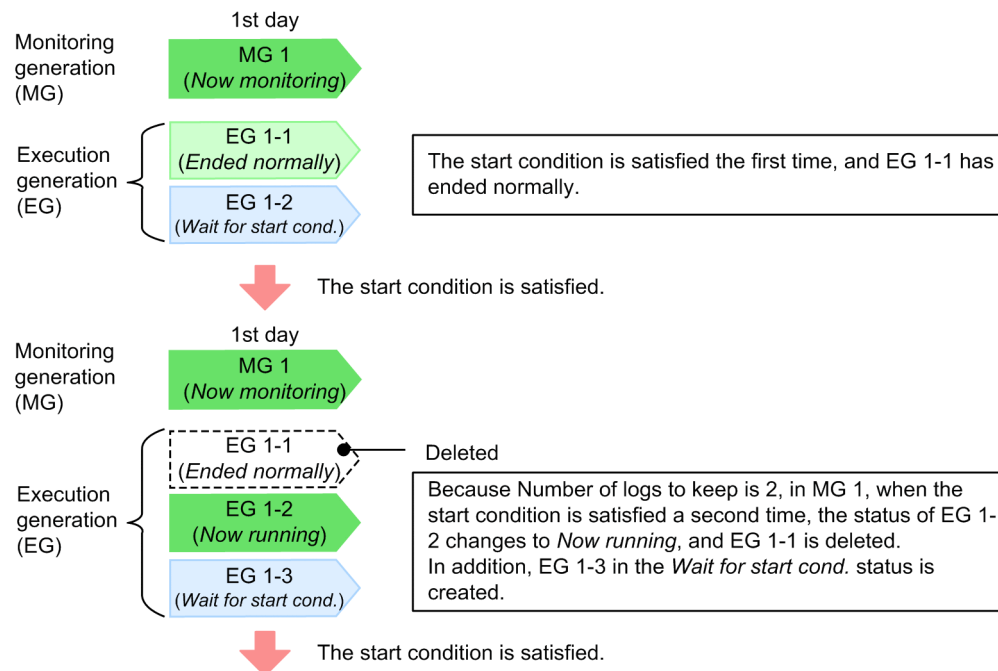
With the total management method, if the number of saved generations exceeds the preset number of logs to keep, saved generations are deleted starting with the oldest generation. For monitoring generations, when the last execution generation in a monitoring generation is deleted, the monitoring generation is also deleted.

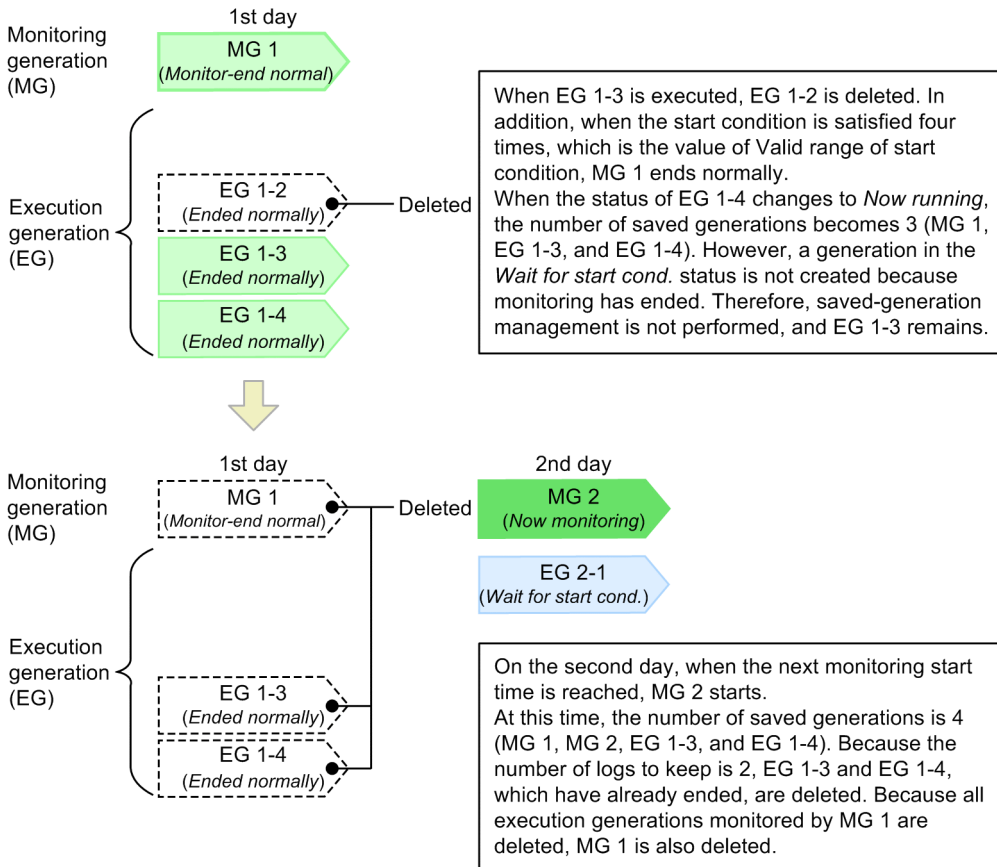
Note that if you use the total management method, you must make sure that you set the number of logs to keep to at least 2. For example, if you want at least one monitoring generation to be saved and at least two execution generations to be saved, you need to set the number of logs to keep to at least 3.

The following figure shows an example of total-management-method processing when *Number of logs to keep* is 2 and *Valid range of start condition* is 4 (number of times satisfied).

Figure 4–16: Example of total-management-method processing (when "Number of logs to keep" is 2, and "Valid range of start condition" is 4 times)

- Number of logs to keep: 2, Valid range of start condition: 4 times





## (b) Saved-generation management by using the individual management method

With the individual management method, the number-of-logs-to-keep setting is applied separately to monitoring generations and to execution generations. More execution generations than the number of logs to keep are saved so you must consider how many execution generations will be created by the specified start condition. If JP1/AJS3 has been upgraded from JP1 Version 8 or an earlier version of JP1/AJS2, the individual management method is still used for the management of saved generations. If you use a jobnet with a start condition, take into consideration that you will be switching to the total management method.

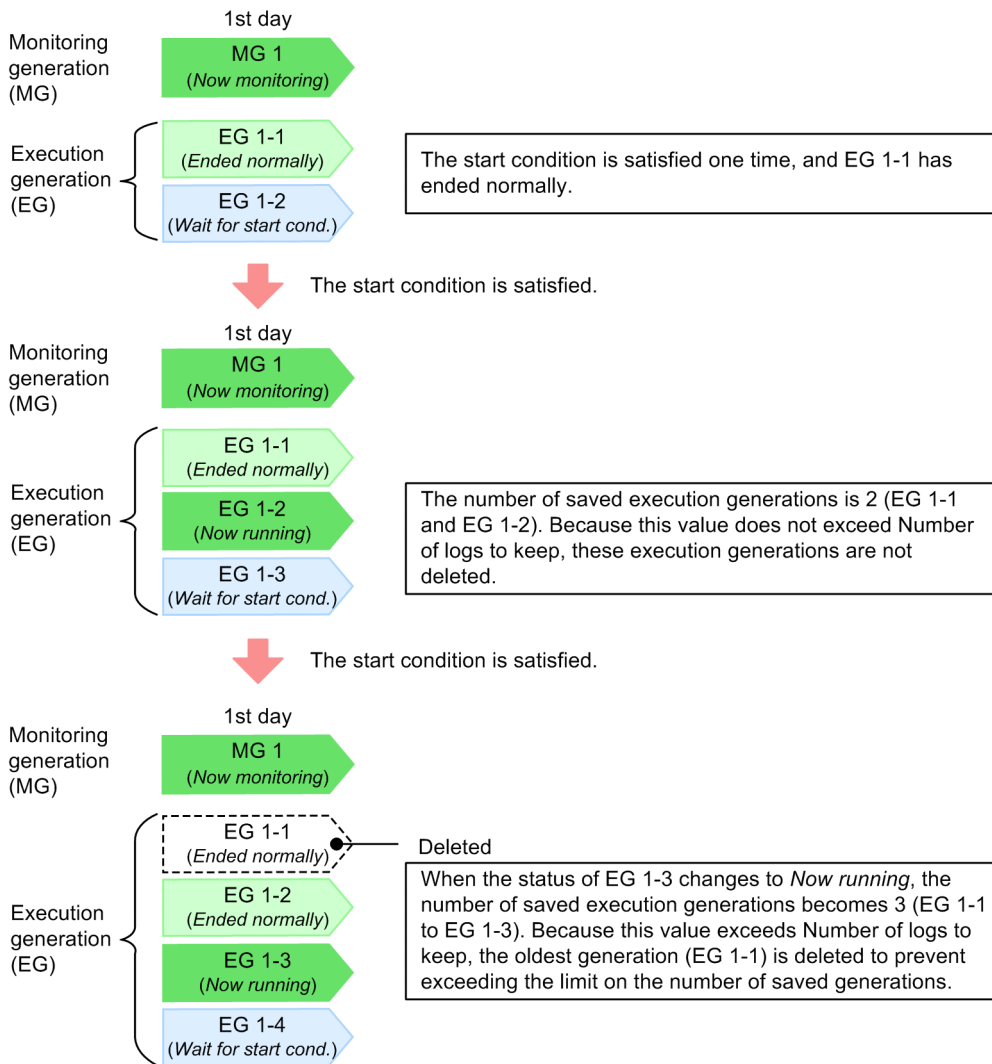
Note that cautionary notes apply when you increase the number of generations that will be saved for a jobnet with a start condition. These notes can be found in [\(4\) Notes applying when JP1/AJS3 has been upgraded from JP1 Version 8 or an earlier version of JP1/AJS2](#).

The following shows examples of the individual management method. These examples assume that *Number of logs to keep* is 2, and *Valid range of start condition* is 4 (number of times satisfied).

- If the number of saved generations exceeds the number of logs to keep on the first day  
The following figure shows an example of processing when the number of saved execution generations exceeds the value of *Number of logs to keep* on the first day.

Figure 4–17: Example of the individual management method when the number of logs to keep is exceeded on the first day

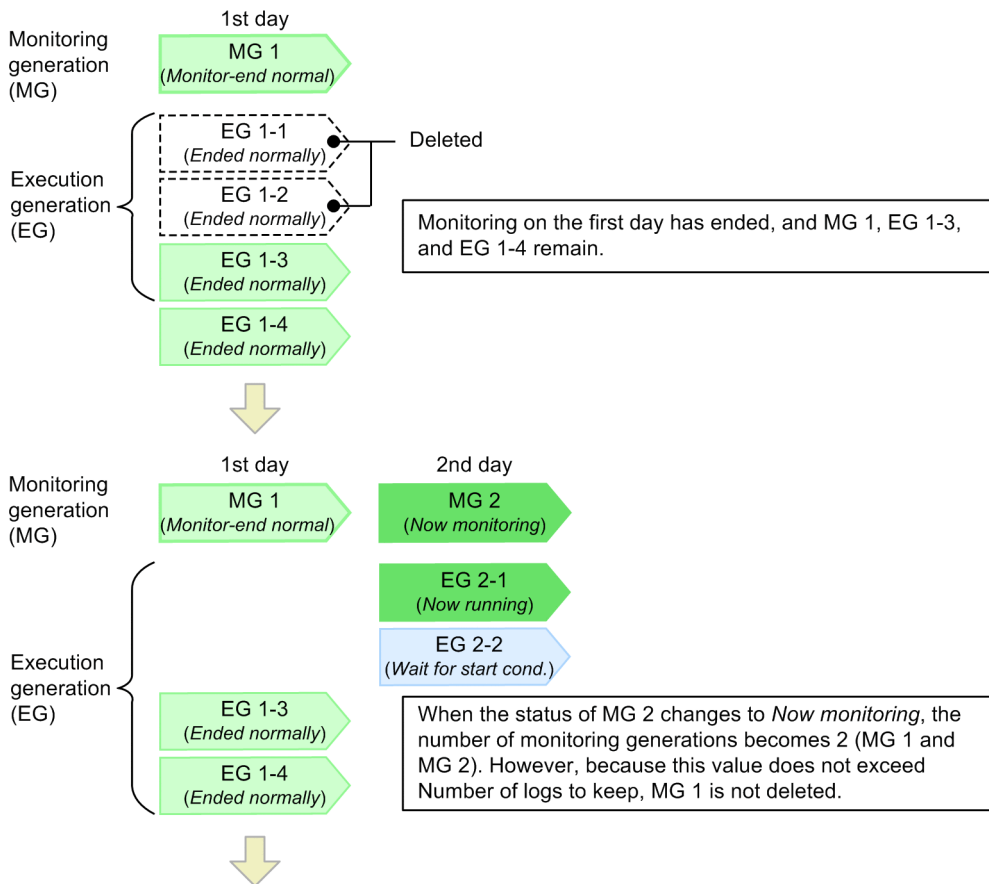
- Number of logs to keep: 2, Valid range of start condition: 4 times

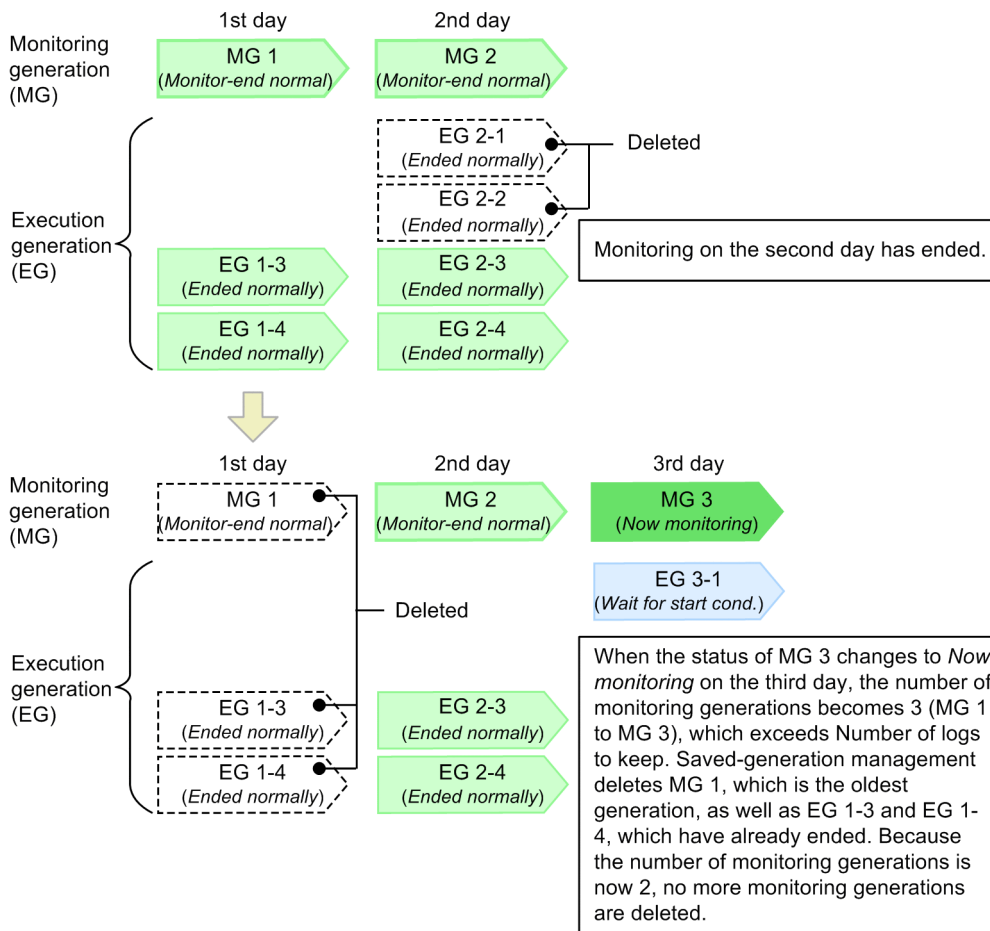


- If the number of saved generations exceeds the number of logs to keep on the second day or later  
The following figure shows an example of processing when the number of saved execution generations exceeds the value of *Number of logs to keep* on the second day or later.

Figure 4–18: Example of the individual management method when the number of logs to keep is exceeded on the second day or later

- Number of logs to keep: 2, Valid range of start condition: 4 times





#### (4) Notes applying when JP1/AJS3 has been upgraded from JP1 Version 8 or an earlier version of JP1/AJS2

If JP1/AJS3 has been upgraded from JP1 Version 8 or an earlier version of JP1/AJS2, the individual management method is selected as the method for managing saved generations. If you use a jobnet with a start condition, take into consideration that you will be switching to the total management method.

##### (a) Switching from the individual management method to the total management method

The method for managing saved generations for a jobnet with a start condition is determined by the value of the `SAVEGENTYPE` environment setting parameter. You can use the `jajs_config` command to change the value of this parameter. For details about this environment setting parameter, see *20.4 Setting up the scheduler service environment* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

If you switch from the individual management method to the total management method, generations other than those retained by the total management method are deleted when the scheduler service starts. The time (in seconds) required for this deletion processing is proportional to the sum of *number-of-logs-to-keep* x *number-of-logs-to-keep* calculated for all registered root jobnets, but can be affected by hardware and other conditions. If the time required for the deletion processing is likely to affect operation, manually delete unnecessary generations before changing the management method. This step should reduce the time required for the scheduler service to start.

Note that generations are also deleted when a generation's registration for execution is canceled. If generations are canceled by using registration cancellation or saved-generation management in synchronous mode, the time required for deletion can be reduced by switching to asynchronous mode. Asynchronous mode is the default setting when JP1/

AJS3 09-00 or a later version or JP1/AJS2 08-00 or a later version is installed as a new installation. For an upgrade installation, however, synchronous mode might be set. You can determine which deletion mode has been set by checking the value of the `BACKGROUNDLEAVE` environment setting parameter. If the value is `no` (synchronous mode) or the parameter is not specified, we recommend that you explicitly specify the value `yes` for the parameter to set asynchronous mode.

For details about changing the mode in Windows, see *6.1.7 Changing the mode in which unregistration or generation management deletes the generations of a jobnet* in the *JP1/Automatic Job Management System 3 Configuration Guide*. For details about changing the mode in UNIX, see *15.1.7 Changing the mode in which unregistration or generation management deletes the generations of a jobnet* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

## **(b) Continuing operation with the individual management method**

If you continue operation with the individual management method rather than switching to the total management system, the following notes, which apply for JP1 Version 8 or an earlier version of JP1/AJS2, apply.

For the individual management method, the setting of the number of logs to keep is applied separately to monitoring generations and to execution generations, as described in *(3)(b) Saved-generation management by using the individual management method*. The number of generations that are saved by this method is equal to  $\text{number-of-logs-to-keep} \times (\text{number-of-logs-to-keep} + 1)$ . The load on any database access operation, such as canceling execution registration, increases proportionately with the number of logs to keep that is set. Determine the value to be specified as the number of logs to keep after careful consideration of the possible impact on system performance. For the relationship between the number of logs to keep that is set and system performance, see *7.2 Relationship between number of logs to keep and performance* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

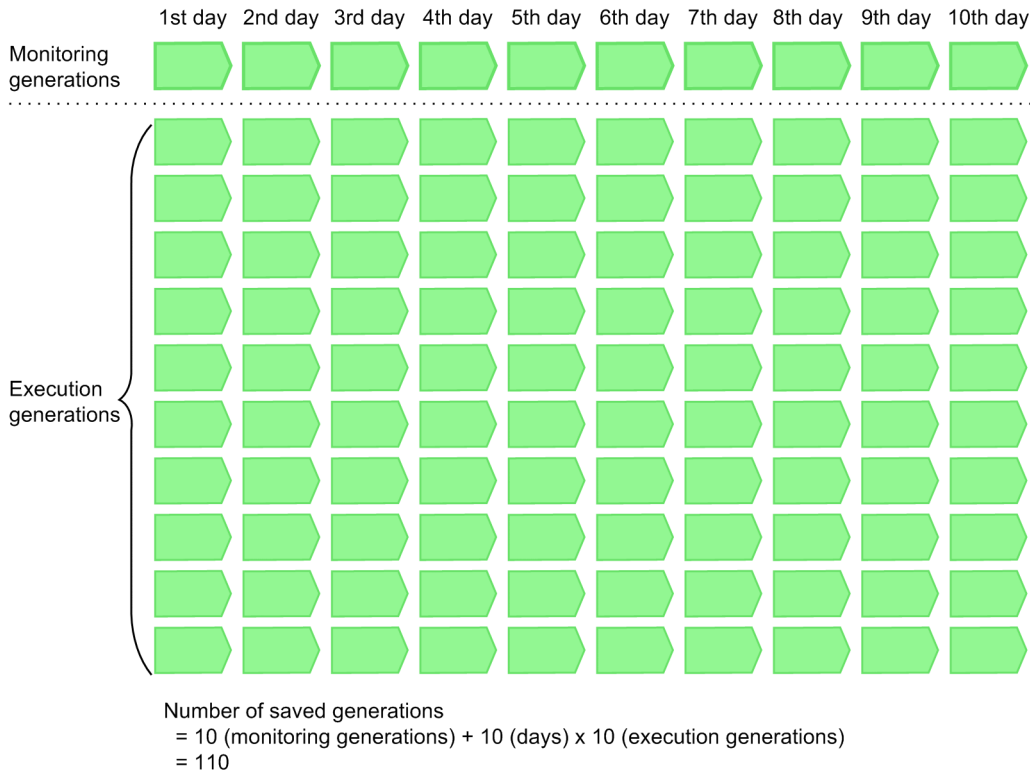
If you choose to use the individual management method for operation of a jobnet with a start condition, we recommend that you set the number of logs to keep to a value less than 10, and use the `ajsshow` command to save execution results as needed. For details about the `ajsshow` command, see `ajsshow` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

You must also be careful when decreasing the number of logs to keep for a jobnet that is already registered for execution. Decreasing the number of logs to keep will render some saved generations no longer necessary, and will cause them to be deleted all at once. Their deletion occurs when the status of the next scheduled monitoring generation changes to *Now monitoring*, creating generations that exceed the number of logs to keep that has been set. The number of generations that will be eliminated is the value of  $(\text{old-number-of-logs-to-keep} - \text{new-number-of-logs-to-keep}) \times (\text{old-number-of-logs-to-keep} + 1)$ . Accordingly, if the value of *Number of logs to keep* is large, and monitoring generations and execution generations each equal to this value have been saved, a drastic decrease in *Number of logs to keep* will eliminate many generations at one time, adversely affecting operation.

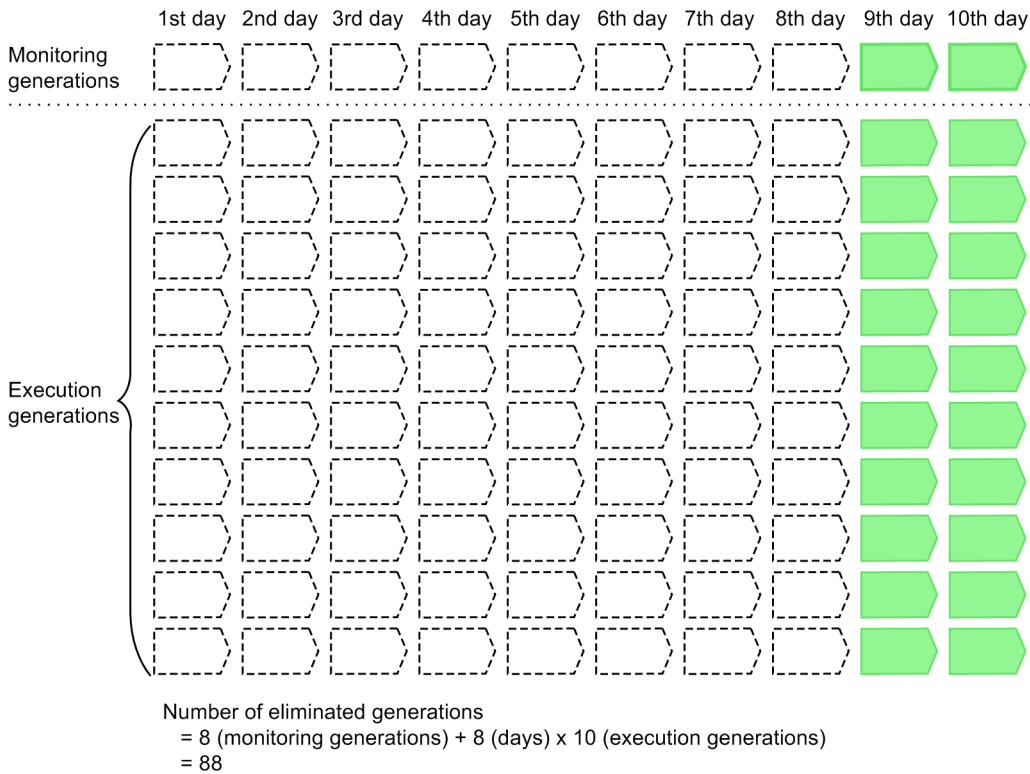
The following figure explains how generations are eliminated when the number of logs to keep is decreased.

Figure 4–19: How generations are eliminated when the number of logs to keep is decreased

- Number of logs to keep is set to 10, and 10 generations are saved



- Number of logs to keep is changed to 2, and saved generations are eliminated



As shown in the above example, when the value of *Number of logs to keep* is 10 and 110 generations are saved, if the value is changed to 2, 88 generations are deleted all at once.



If you continue to use the individual management method, keep the above notes in mind.

## 4.3 Checking execution registration information

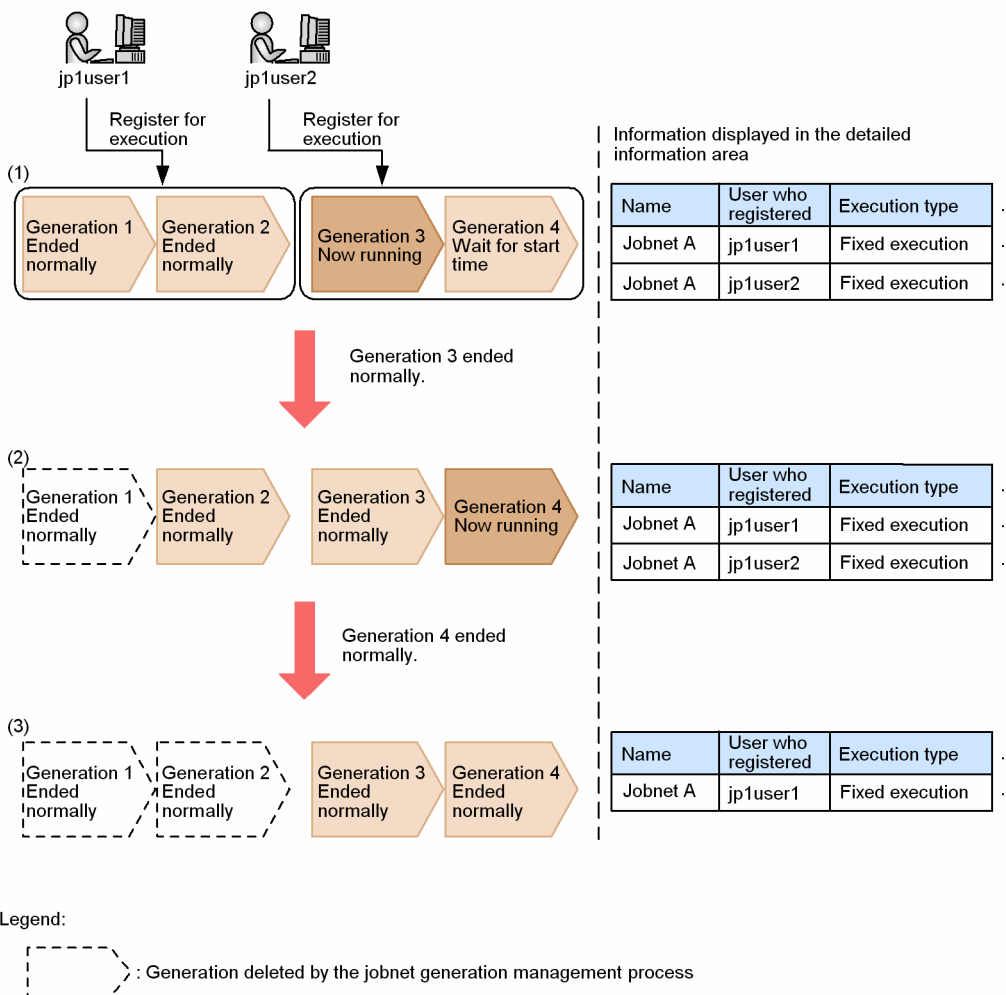
You can view the execution registration information for a jobnet in the JP1/AJS3 - View window (main window) of JP1/AJS3 - View or in the List window of the Web GUI. From the **Function** menu, choose **Register for Execution**. You can then display execution registration information about a selected 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 4–20: Relationship between jobnet generation management and the information displayed in the detailed information area

● Number of logs to keep: 2



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 *JP1/Automatic Job Management System 3 Operator's Guide*.

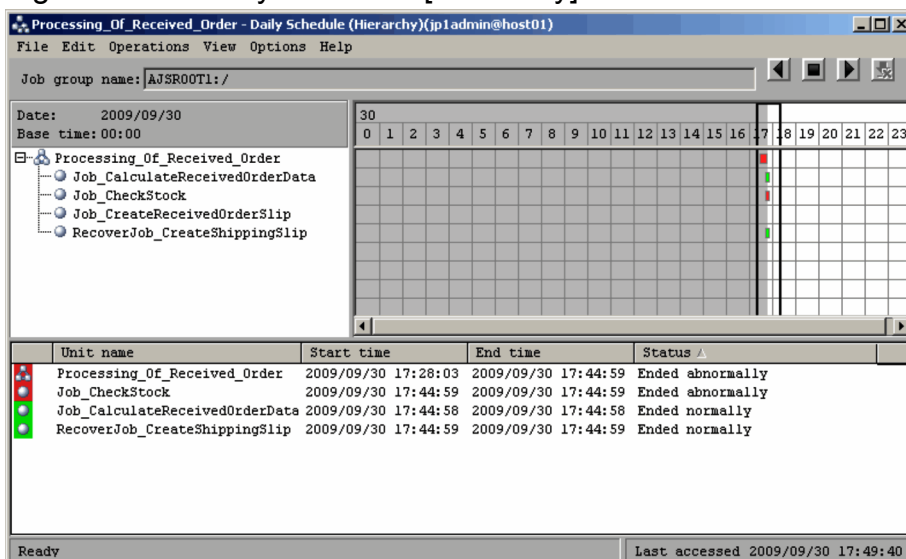
You can also check the schedule of a jobnet on the Monthly Schedule screen of the Web GUI. For details about Monthly Schedule screen, see *A.7 Monthly Schedule screen* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

#### (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 4–21: Daily Schedule [Hierarchy] window



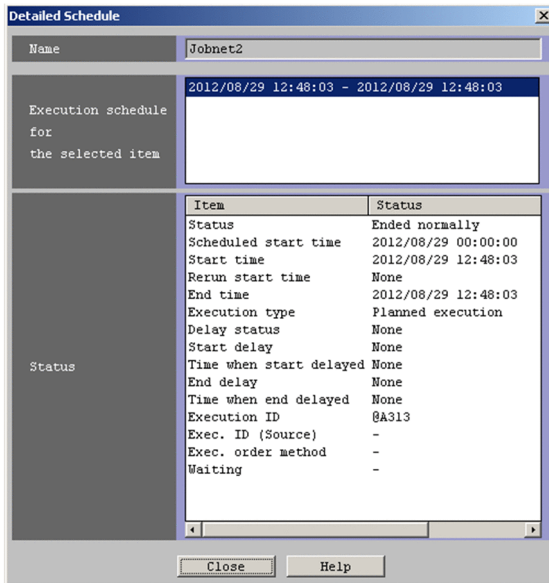
The Daily Schedule [Hierarchy] window is divided into hours of the day. Here you can check the daily schedules, their execution status and results. In the schedule area, the gray part shows past schedules and the white part shows future schedules. When you click a unit schedule in the gray area, the 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 4–22: Detailed Schedule dialog box

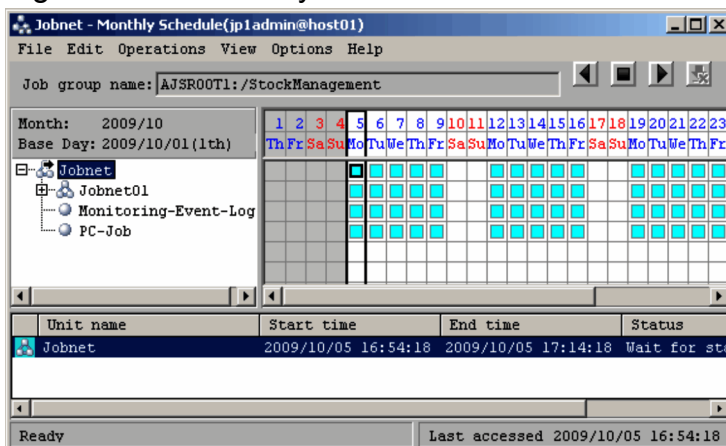


## (2) Checking schedules in the Monthly Schedule window

In the Monthly Schedule window, you can check the execution schedules, status, and results of the jobnets and jobs scheduled for a specific month. You can also display the execution schedules of jobnets or jobs that have not been registered for execution to check the definition of the schedule information.

The following figure shows the Monthly Schedule window.

Figure 4–23: 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.

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 4-22](#).

The following describes how execution schedules are displayed before and after execution registration.

### (a) How execution schedules are displayed before execution registration

A triangle appearing in the schedule area indicates the execution schedules of units that have not been registered for execution. Note that by default, the execution schedules of unregistered units are not displayed. For details about how to display these execution schedules, see *6.4 Checking a defined schedule* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

Note that when you display the execution schedules of units that have not been registered for execution, by default, the detailed information is not displayed in the execution result list. For details about how to display the detailed information, see *11.9.8 Displaying the execution schedule of unregistered units in the execution result list* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

When you display the detailed information in the execution result list, the start time for each unit is displayed. Note that because the start time for units that have not been registered for execution is calculated without taking into consideration the sequence of preceding/succeeding or upper/lower units, the schedule definition information is displayed as is.

### (b) How execution schedules are displayed after execution registration

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.

If you select the schedule of a unit in the gray portion of the schedule area, you can check the detailed information of the execution result or execution status in the execution result list. If you select the schedule of a unit in the white portion of the schedule area, you can check the detailed information of the execution schedule in the execution result list. Note, however, that generations that do not have an execution ID are not displayed.

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*.

## 4.4.2 Simulation

JP1/AJS3 uses two types of simulation:

- Schedule simulation
- Execution simulation

Schedule simulation and execution simulation are used to calculate budget and result information. The budget and result information is displayed in the execution result list in the Daily Schedule and Monthly Schedule windows of JP1/AJS3 - View and displayed in the Monthly Schedule window of the Web GUI. 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 *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

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 4–24: 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

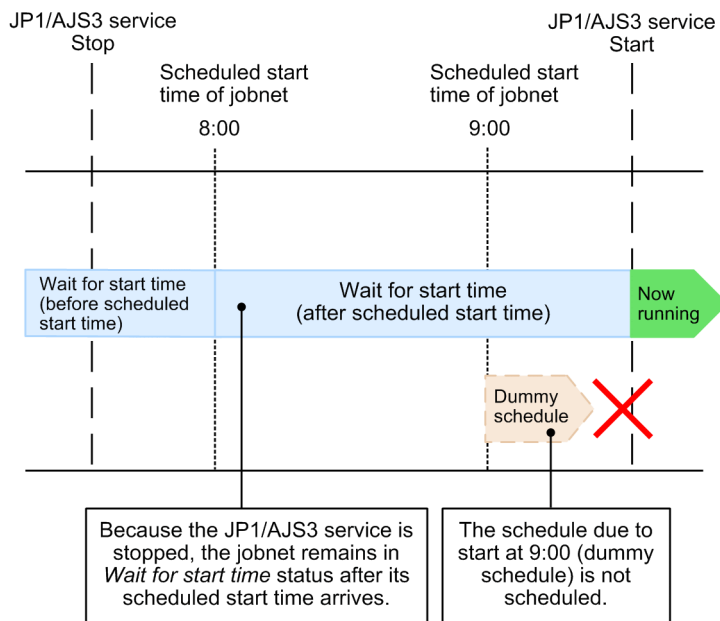
A *dummy schedule* is one that has not been finalized.

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 4–25: 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*.

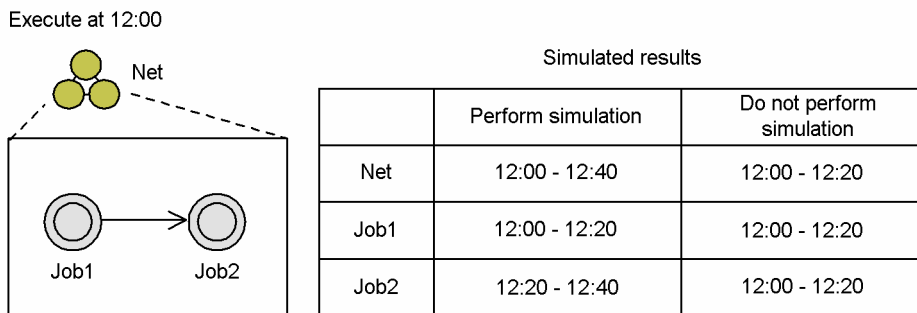
In execution simulation, JP1/AJS3 simulates the scheduled start and end times of a jobnet and jobs to more closely match actual operation. The simulated times are based on various information, including the current time, scheduled start time of the jobnet, its execution status, past execution time, types of units and unit flow. If JP1/AJS3 - View is used, JP1/AJS3 simulates the scheduled start and end times of a jobnet as follows. If the jobnet has never been executed before, the time (in seconds) set in **First execution time** in the Preferences dialog box is assumed 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 based on the assumed execution time. If the Web GUI is used, 1,200 seconds is assumed as the required execution time of the first job. If the `ajsshow` command is used, 60 seconds is assumed as the required execution time of the first job.



You can select whether to perform execution simulation in the Preferences dialog box (if JP1/AJS3 - View is used) or the Setting dialog box (if JP1/AJS3 - Web Console is used). For details on settings in the Preferences dialog box, see *12.3.42 Preferences dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details about Setting dialog box, see *A.27 Setting dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. If the `ajsshows` command is used, the settings in the Preference dialog box and Setting dialog box are ignored.

If you select not to perform execution simulation, the scheduled start time is displayed based on the jobnet's schedule rule. In that case, the scheduled end time is calculated, assuming 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 4–26: Differences between start and end times depending on whether execution simulation is performed**



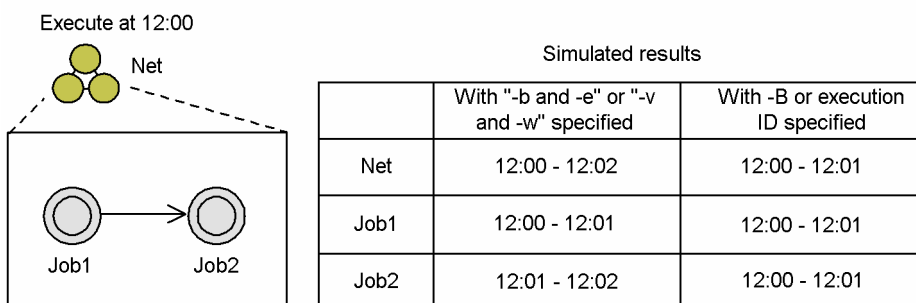
In this example, assume that 1,200 seconds (default) is set in **First execution time** in the Preferences dialog box of JP1/AJS3 - View, and that the jobnet has never been executed since it was registered.

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 `ajsshows` command always performs simulation. When the `-B` option or execution ID is specified, the `ajsshows` command does not perform simulation. The following figure shows the difference in output depending on the specified option.

**Figure 4–27: Difference in output depending on the option specified in the `ajsshows` 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
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.
- Execution simulation is not performed for unregistered units. For details, see *6.4 Checking a defined schedule* in the *JP1/Automatic Job Management System 3 Operator's Guide*.
- 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

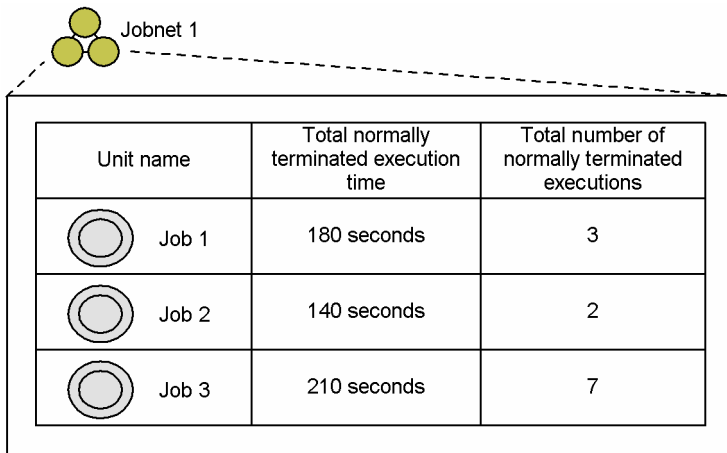
Simulation is performed based on statistical information. For details on execution simulation, see [4.4.2\(2\) Execution simulation](#).

## (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 4–28: Management of statistical information



The diagram shows a root jobnet labeled 'Jobnet 1' at the top left, represented by three yellow circles. A dashed line connects this jobnet to a table below. The table has three columns: 'Unit name', 'Total normally terminated execution time', and 'Total number of normally terminated executions'. It lists three jobs: Job 1 (180 seconds, 3 executions), Job 2 (140 seconds, 2 executions), and Job 3 (210 seconds, 7 executions). Each job name is accompanied by a small icon of a circle with a dot inside.

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 times 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

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 4–3: Processes available for jobnets and jobs that have been registered for execution

No.	Process	Content	Reference
1	Cancel jobnet registration	Cancels registration of a root jobnet for execution. When registration is canceled, all results from previous executions are also deleted.	4.5.1
2	Add scheduled execution times and dates	When the schedule of a root jobnet whose status is <b>Being held</b> or <b>Waiting for start time</b> 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.	4.5.2
3	Temporarily change execution time	Temporarily changes the scheduled execution start time of a unit. <sup>#1</sup>	4.5.3
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. <sup>#1</sup> <ul style="list-style-type: none"> <li>• With root jobnets: the scheduled execution start time is changed to the current time.</li> <li>• With nested jobnets: the scheduled execution start time is changed to the base time of the execution date.</li> </ul>	4.5.4
5	Prohibit execution	Cancels the next scheduled execution, or the execution in a fixed schedule.	4.5.5
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. <sup>#2</sup>	4.5.6
7	Temporarily change delay monitor	Temporarily changes the delay monitor time (in hours) of a jobnet that is waiting to execute.	4.5.7
8	Temporarily change jobnet priority	Temporarily changes the execution priority of a root jobnet that is waiting to execute.	4.5.8
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.	4.5.9
10	Kill	Kills the unit being executed. <sup>#3</sup> 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.	4.5.10

No.	Process	Content	Reference
11	Rerun	Reruns a jobnet or job. The rerun is not counted in the number of generations (execution count).	<a href="#">4.5.11</a>
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.	<a href="#">4.5.12</a>
13	Specify generation when executing command	Specifies a generation whose execution registration information is to be manipulated by using a command.	<a href="#">4.5.13</a>
14	Switch jobnet definition at the specified time	Automatically switches the definition of the jobnet currently being executed at the specified time by using the jobnet release function.	<a href="#">4.5.14</a>
15	Temporarily change wait condition settings	Temporarily enables or disables the wait condition settings for each unit whose end is being waited for. A unit with the wait condition that all units whose end is being waited for must be disabled, starts execution immediately.	<a href="#">4.5.15</a>
16	Display and re-apply temporary changes	Displays in JP1/AJS3 - View a list of temporary change information for a job or jobnet registered for execution as temporary changes, and re-applies the changes selected in the list to the job or jobnet.	<a href="#">4.5.16</a>
17	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.	<a href="#">4.5.17</a>

#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.

To cancel jobnet registration, you can use JP1/AJS3 - View, the Web GUI, the API function to cancel jobnet registration, or the `ajsleave` command. For details on using JP1/AJS3 - View, see *7.3 Releasing the jobnet from registration for execution* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.21 Cancel Registration dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on the registration cancellation API, see *7.1.7 Registration cancellation API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajsleave` command, see the description of `ajsleave` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## 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 register a jobnet for fixed execution by specifying a specific date and time, an execution schedule is added. For details about fixed execution by specifying a specific date and time, see *4.1.1(3) Fixed execution*.

Any schedule definitions for the added execution schedule are ignored by both root jobnets and nested jobnets. 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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to fixed execution registration, or the `ajsentry` command. For details on using JP1/AJS3 - View, see *9.2 Adding an execution schedule for a jobnet* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.22 Add dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on the fixed execution registration API,

see *7.1.5 Fixed execution registration API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajsentry` command, see the description of `ajsentry` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

Note that for a jobnet registered for fixed execution by specifying the number of future generations, when you add an execution schedule that is to take place later than the dummy schedules of the jobnet, dummy schedules earlier than the added execution schedule are not executed, the same as when you temporarily change the execution start time. If you want to execute dummy schedules earlier than the added execution schedule, add the execution schedule to the date and time of the dummy schedules. For details, see *4.5.3(4) Changing to a time that extends beyond the dummy schedule*.

## (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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to change plan (change time), or the `ajsplan` command. For details on using JP1/AJS3 - View, see *9.2 Adding an execution schedule for a jobnet* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.22 Add dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the change plan (change time) API, see *7.1.11 Change plan (change time) API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. For details on the command operations, see the `ajsplan` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## 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. Furthermore, if you push ahead the execution schedule of a root jobnet registered for fixed execution and then add a new execution schedule, you can cancel the changes and restore the execution time of the execution schedule that you pushed ahead and added to its original execution time. However, because the execution schedule for which changes were canceled does not just disappear and return to the same execution start time that existed

in the original execution schedule, two execution schedules are executed at the same execution start time. To cancel the execution schedule that you pushed ahead and added, prohibit execution for the execution schedule instead of canceling the changes.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to change plan (change time), or the `ajsp1an` command. For details on using JP1/AJS3 - View, see *9.3 Temporarily changing and restoring the execution start time of a jobnet* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.23 Change Time dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the change plan (change time) API, see *7.1.11 Change plan (change time) API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajsp1an` command, see the description of `ajsp1an` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## (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.

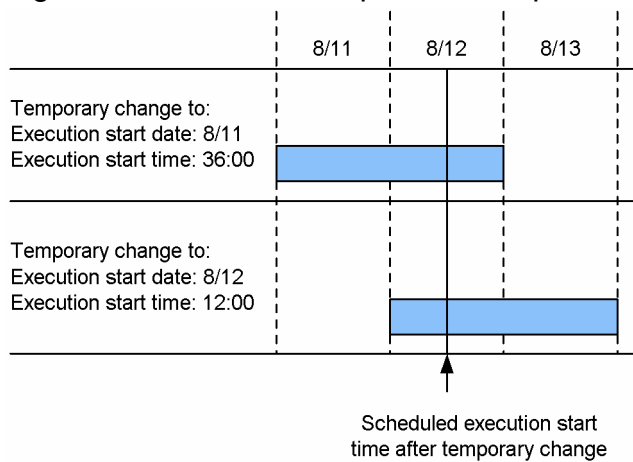
If you temporarily change the execution start time to 12:00 on August 12, the execution date becomes August 12. This means that you can define a schedule between 0:00 and 47:59 on August 12 for a nested jobnet.

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 4–29: Relationship between specified start time and execution date



Legend:

Valid range for the schedule of the nested jobnet

#### Cautionary notes

- 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

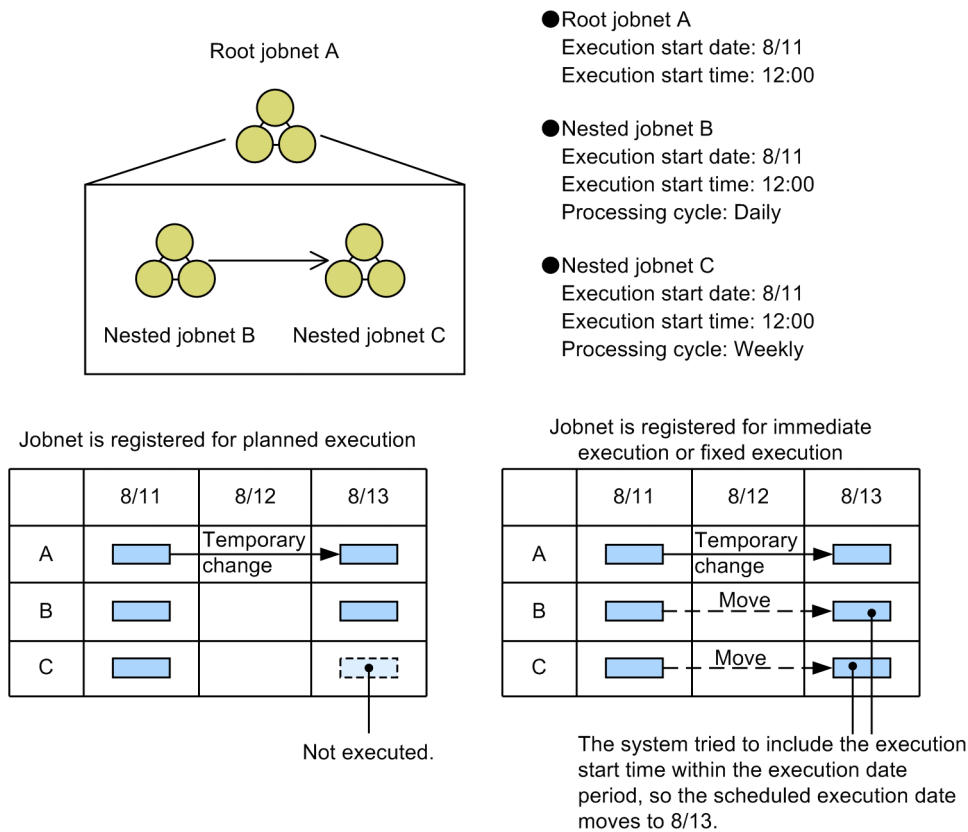
When you temporarily change the execution schedule of a jobnet under which nested jobnets are defined, you can pass the changes on to all lower jobnets. If you shift the start time of the upper-level jobnet by two hours, then provided you have set JP1/AJS3 to pass the changes on, the start times of the 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 4–30: 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 4–31: Changing the scheduled execution date to a date beyond the dummy schedule

	8/4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Old schedule	●				●			●			△			△		
Changed date								→				●				
New 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.

If you want to execute the dummy schedule for August 14, perform the following operations:

For jobnets registered for planned execution:

If you want to execute the dummy schedule for August 14, rather than temporarily changing the execution time, prohibit execution for the execution schedule for August 11, and then add a schedule rule that adds an execution schedule for August 15.

For details about prohibit the execution of execution schedules, see [4.5.5 Prohibiting execution of a job or jobnet](#).

For jobnets registered for fixed execution by specifying the number of future generations:

Change the execution schedule from August 11 to August 15, and then add an execution schedule for August 14, the day on which a dummy schedule exists.

For details about add an execution schedule, see [4.5.2 Adding an execution schedule to a jobnet](#).

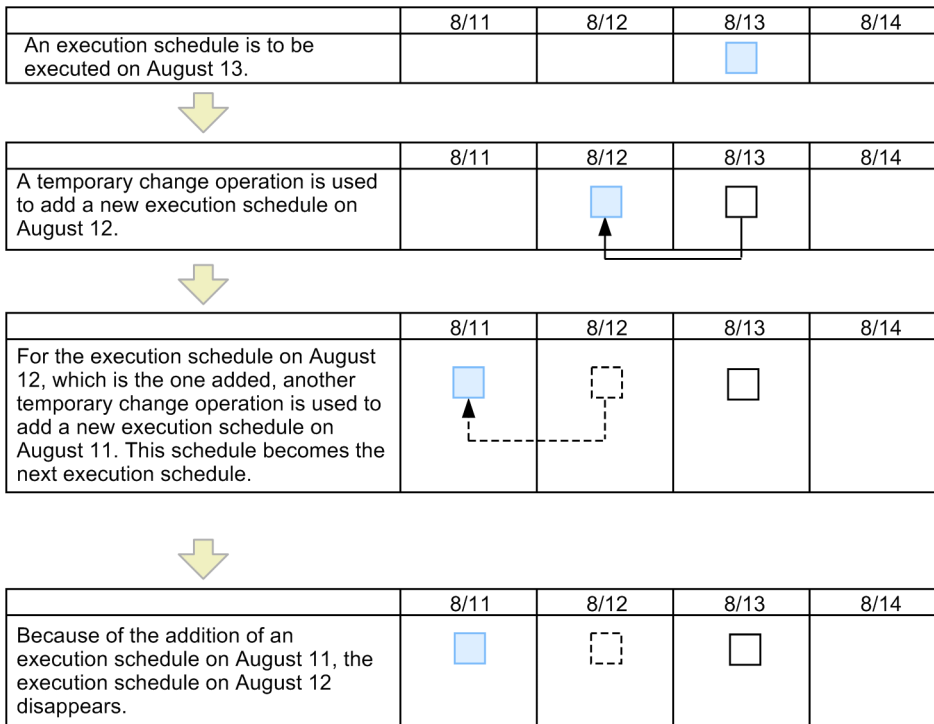
## (5) Adding execution schedules to a jobnet registered for planned execution by performing multiple temporary change operations

In planned execution registration, only the next execution schedule is determined from the last execution schedule, and the execution schedules to be executed following the next execution schedule are treated as dummy schedules. For this reason, for a jobnet registered for planned execution, only the next execution schedule whose execution time has already been determined can be temporarily changed.

If you want to temporarily set a new execution schedule before the next execution schedule, you can select either to move the next execution schedule (**Not in next schedule**) or to add a new execution schedule (**Also in next schedule**). Note that, when a new execution schedule is added, it can only be added as the next execution schedule. You cannot perform a temporary change operation that adds multiple execution schedules, including those that follow the next execution schedule. If you perform a temporary change operation that adds new execution schedules in succession, only the last specified schedule becomes the next execution schedule.

The following figure shows how the execution schedules are determined if multiple temporary change operations are performed for a jobnet registered for planned execution.

Figure 4–32: How the execution schedules are determined if multiple temporary change operations are performed for a jobnet registered for planned execution



Legend:

- : Next execution schedule
- : Dummy execution schedule
- : Execution schedule that has disappeared

#### 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.

#

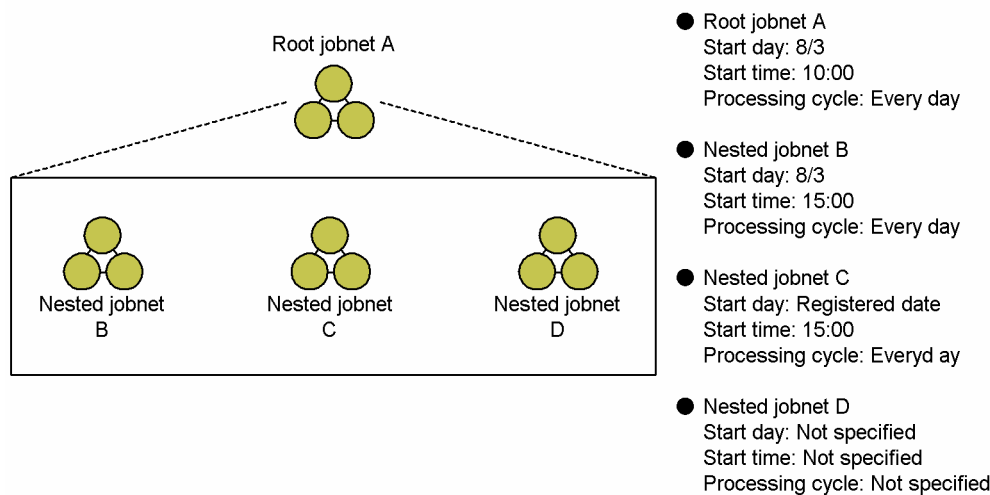
For JP1/AJS3 - View, this applies when the **Change date and time of all lower jobnets** check box is selected, and immediate execution is performed by selecting **Execute Immediately** from **Change Plan**. For a command, this applies when immediate execution is performed by executing the `ajsp lan` 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 4–33: 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)

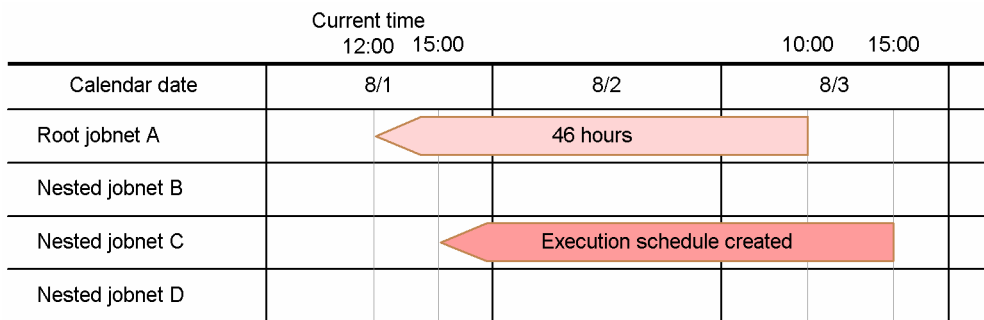
	8/1	8/2	8/3
Root jobnet A			<input type="radio"/>
Nested jobnet B			<input type="radio"/>
Nested jobnet C			<input type="radio"/>
Nested jobnet D			

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 4–34: 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.

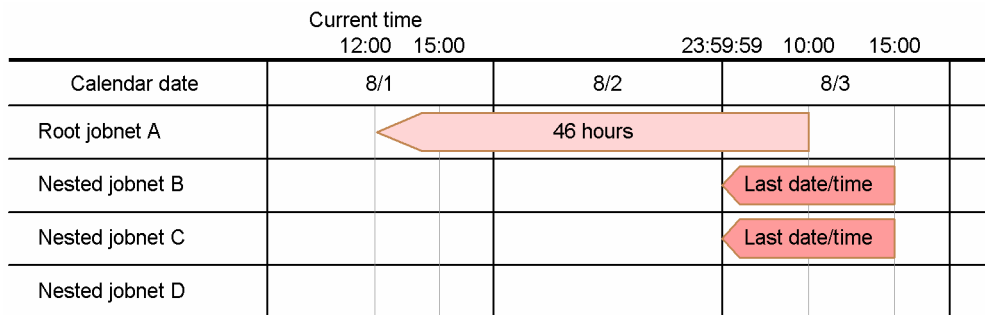
Nested jobnet D is not executed because it has not been scheduled for execution on August 1 (8/1).

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 4–35: 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)).

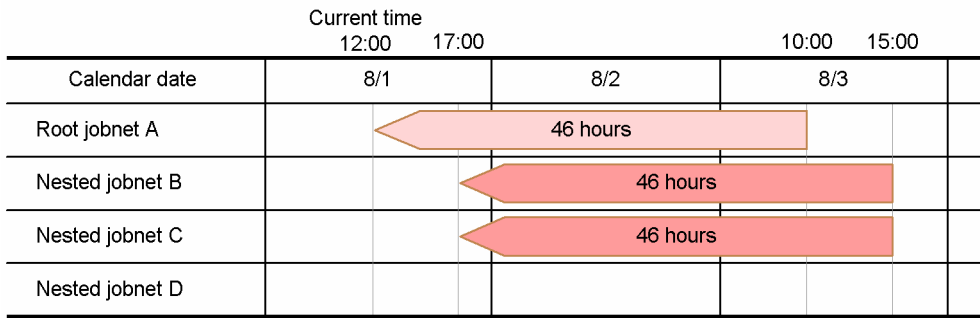
Nested jobnet D is not executed because it has not been scheduled for execution on August 1 (8/1).

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 4–36: 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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to change plan (immediate execution), or the `ajsplan` command. For details on using JP1/AJS3 - View, see *9.4 Immediately executing the execution schedule of a jobnet* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.24 Execute Immediately dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the change plan (execute immediately) API, see *7.1.12 Change plan (execute immediately) API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajsplan` command, see the description of `ajsplan` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## 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<sup>#</sup> 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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to change plan (execution prohibited), or the `ajspplan` command. For details on using JP1/AJS3 - View, see *9.5 Temporarily canceling the execution of jobnets and jobs* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.7 Monthly Schedule screen* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the change plan (execution prohibited) API, see *7.1.13 Change plan (execution prohibited) API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajspplan` command, see the description of `ajspplan` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

#

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.



To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to temporarily change the hold attribute, or the `ajspplan` command. For details, see the relevant sections in the following manuals:

- *9.6 Holding the execution of jobnets and jobs* in the *JP1/Automatic Job Management System 3 Operator's Guide A.5 Dashboard screen* in the *JP1/Automatic Job Management System 3 Operator's Guide*
- *A.7 Monthly Schedule screen* in the *JP1/Automatic Job Management System 3 Operator's Guide*
- *A.10 Monitor dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*
- *7.1.8 Change hold attribute API* in the manual *JP1/Automatic Job Management System 3 Command Reference*
- *ajspplan* in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

#### 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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to change delay monitor, or the `ajsplan` command. For details on using JP1/AJS3 - View, see *9.7 Temporarily changing the delay monitor date and time of jobnets* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.25 Change Delay Monitor Temporarily dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the change delay monitor API, see *7.1.9 Change delay monitor API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajsplan` command, see the description of `ajsplan` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## 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 can set **none** or **1** to **5**, where **1** is the lowest priority, and **5** is the highest. For Unix jobs or HTTP connection jobs that are executed on Linux hosts, you can specify a priority of **4** or **5** only if the OS user who will execute the job has superuser privileges. If the execution priority of a job and the priority of a jobnet are set to None, the execution priority is assumed to be the value specified for the `DEFAULTPRIORITY` environment setting parameter when the job or jobnet is executed.

For details about the `DEFAULTPRIORITY` environment setting parameter, see *20.4.2(111) DEFAULTPRIORITY* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

The following table shows execution priority values and their meanings.

**Table 4–4: 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 `jajs_spmc` 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:

$39 \approx 20$  (default) + 20 (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 `ajspplan` command. For details on using the GUI, see 9.8 *Temporarily changing the execution priority of jobnets* in the *JP1/Automatic Job Management System 3 Operator's Guide*. Alternatively, for details on the `ajspplan` command, see the description of `ajspplan` in 3. *Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## 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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to interrupt, or the `ajsintrpt` command. For details on using JP1/AJS3 - View, see 9.9 *Interrupting a jobnet being executed* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see A.10 *Monitor dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the interrupt API, see 7.1.15 *Interrupt API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajsintrpt` command, see the description of `ajsintrpt` in 3. *Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## 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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to kill, or the `ajskill` command. For details on using JP1/AJS3 - View, see 9.10 *Killing the jobnets and the jobs being executed* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see A.10 *Monitor dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the kill API, see 7.1.16 *Kill API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajskill` command, see the description of `ajskill` in 3. *Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## (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.

If the jobs that are running on Windows are forcibly terminated, only the processes started by JP1/AJS3 will be forcibly terminated. The processes under each of these processes will not be forcibly terminated. In addition, even if the running jobs are forcibly terminated, the resources that were being used by the processes under each of these processes will not always be made free.

However, if the processes started by JP1/AJS3 are the processes specified in the script file (.spt) created by JP1/Script, the range of the processes to be forcibly terminated depends on the processes under these started processes.

For details, see [5.4.10\(3\) Killing a job created by JP1/Script](#).

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.

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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to rerun, or the `ajsrrerun` command. For details on using JP1/AJS3 - View, see *9.11 Re-executing jobnets and jobs* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.12 Rerun dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the rerun API, see *7.1.17 Rerun API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajsrrerun` command, see the description of `ajsrrerun` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

### Supplementary notes

- 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 *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.
- For details about re-executing a jobnet when the execution sequence of the unit is controlled by using wait conditions, see the following subsection: *2.2.5 Using wait conditions to control the order of unit execution* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*

### (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 jobnet or job to be re-executed is in the *Now Running*, *Running + Abend*, *Running + Warning*, *Wait for prev. to end*, *Wait for start time*, or *Being held* status.
- The jobnet or job to be re-executed has the Recovery attribute, and the preceding jobnet or job has ended normally or with a warning.
- The jobnet or job to be re-executed is a judgment job or a dependent unit of a judgment job.  
Note that dependent units of judgment jobs are re-executable if `yes` is set for the `RERUNSUBORDINATE` environment setting parameter. For details about whether to enable re-execution of dependent units, see (4) *Re-executing the dependent unit of a judgment job*.
- The upper-level jobnet of the jobnet or job to be re-executed 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<sup>#</sup>**
- **From abnormally ended jobnet**
- **From top**
- **From any job or jobnet** (excludes dependent units)
- **Only jobs that ended with warning**

<sup>#</sup> 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 unit of a judgment job

The execution of the dependent unit of a judgment job depends on the execution result of the judgment job. If the dependent unit associated with a judgment job terminates abnormally, whether the dependent unit can be re-executed and how to re-execute it depend on the setting of the `RERUNSUBORDINATE` environment setting parameter. For details

about the parameter, see the appropriate manual section for the platform you are using. If you are using Windows, see 6.2.23 *Settings for re-executing the dependent units* in the *JP1/Automatic Job Management System 3 Configuration Guide*. If you are using UNIX, see 15.2.21 *Settings for re-executing the dependent units* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

If re-execution of dependent units is disabled (environment setting parameter `RERUNSUBORDINATE = no`):

Unlike ordinary jobs (such as PC jobs and Unix jobs), dependent units cannot be re-executed. Re-execution from the judgment job that precedes a dependent unit is also not possible.

If a dependent unit terminates abnormally, re-execute the jobnet that contains the dependent unit.

If re-execution of dependent units is enabled (environment setting parameter `RERUNSUBORDINATE = yes`):

Whether dependent units can be re-executed depends on the unit type and the status of the dependent units.

The dependent unit of a judgment job can be re-executed when all of the following conditions exist:

- The version of JP1/AJS3 - Manager is 10-00 or later.
- The judgment job evaluates to true.
- The judgment job is in *Ended normally* status.
- In the following table, Y indicates the unit type of a dependent unit that can be re-executed.

**Table 4–5: Unit types of dependent units that can be re-executed**

No.	Unit type	Unit defined as a dependent unit	Unit defined in a dependent jobnet
1	Job group	--	--
2	Root jobnet	--	--
3	Nested jobnet	Y	Y
4	Root remote jobnet	--	--
5	Nested remote jobnet	Y	Y
6	Manager job group	--	--
7	Manager jobnet	--	--
8	Planning group	--	--
9	Standard job	Y	Y
10	Jobnet connector	--	Y
11	OR job	--	Y
12	Judgment job	--	N
13	Event job	Y	Y
14	Action job	Y	Y
15	Custom job	Y	Y
16	Passing information setting job	Y	Y
17	HTTP connection job	Y	Y

Legend:

Y : The unit can be re-executed (if all necessary conditions are satisfied).

N : The unit cannot be re-executed.

-- : The unit cannot be defined as a dependent unit.

- In the following table, Y indicates a dependent unit in the indicated can be re-executed.

Table 4–6: Statuses and types of dependent units that can be re-executed

No.	Status	Type of dependent unit		
		Jobnet	Job	Jobnet connector
1	Not sched. to exe.	N	N	N
2	Wait for start time	N	--	--
3	Wait for prev. to end	N	N	N
4	Being held	N	N	--
5	Waiting to execute	--	N	--
6	Now queuing	--	N	--
7	Not executed + Ended	Y	Y	Y
8	unexec-W	Y	Y	--
9	Bypassed	Y	Y	Y
10	Now running	N	N	N
11	Running + Abend	N	--	N
12	Running + Warning	N	--	N
13	Ended normally	Y	Y	Y
14	Normal end + False	--	--	--
15	Ended with warning	Y	Y	Y
16	Ended abnormally	Y	Y	Y
17	abnormal-WR	Y	Y	--
18	Skipped so not exe.	N	--	--
19	Invalid exe. seq.	Y	--	--
20	Interrupted	Y	--	--
21	Killed	Y	Y	Y
22	kill-WR	Y	Y	--
23	Failed to start	--	Y	--
24	fail-WR	--	Y	--
25	Unknown end status	--	Y	Y
26	unknown-WR	--	Y	--
27	Shutdown	N	N	N
28	Wait for start cond.	--	--	--
29	Now monitoring	--	--	--
30	Unmonitored + Ended	--	--	--
31	Monitor terminated	--	--	--
32	Interrupted monitoring	--	--	--
33	Monitor-end normal	--	--	--



Legend:

- Y : The unit can be re-executed.
- N : The unit cannot be re-executed.
- : The unit cannot have the indicated status.

## (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 4–7: 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	V5: Ended with warning #2 V6: Interrupted #2

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
17	Interrupted	Y	Y	N	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
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

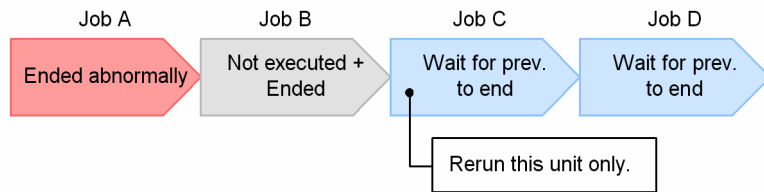
unit might not move from the *Waiting for prev. to end* status to the *Not executed + Ended* status, depending on the timing. Before you attempt to rerun the succeeding unit of the abnormally terminated unit, wait until all the succeeding units are placed in the *Not executed + Ended* status. You must be especially careful when you rerun the succeeding unit from a recovery unit.

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 4–37: When the unit in the *Wait for prev. to end* status is rerun

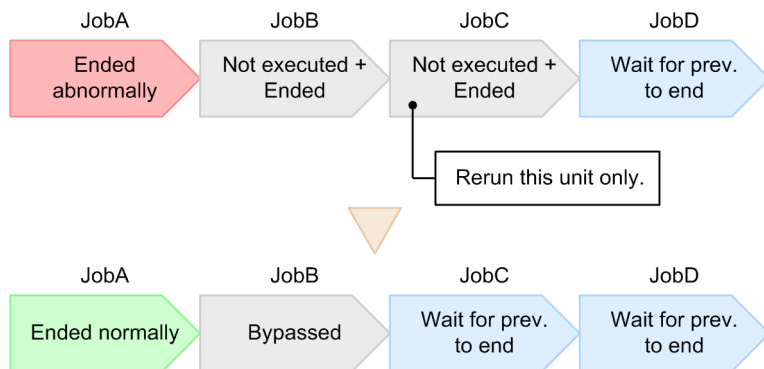


Job C is in the *Wait for prev. to end* status and therefore cannot be rerun. Wait until Job C moves to the *Not executed + Ended* status, and then rerun it.

- 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 4–38: 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

Execution schedules are not skipped when a jobnet whose schedule option is set to **Schedule skip** is rerun and the scheduled start time of the next run is reached during the re-execution. In this case, the execution schedule, which was not skipped, is executed in accordance with the settings for concurrent executions. 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 *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

## (6) Notes on resuming processing while a recovery unit is running

If a job has terminated abnormally, JP1/AJS3 starts a sequence of recovery units. However, if you resume the processing from the job after the abnormally terminated one, the rest of the recovery units 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 unit 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, are running, or have 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. In such cases, until the currently running job actually ends, the agent management control will count the job toward the maximum number of jobs that can be executed concurrently.

After changing the status of a job, you can use the `ajsagtshow` and the `jpgagtshow` commands to check the actual number of jobs in the queue and the number of jobs currently running. For details about these commands, see `ajsagtshow` in 3. *Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*, and see `jpgagtshow` in 4. *Commands Used for Special Operation* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

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.

To this operation, you can use JP1/AJS3 - View, the Web GUI, the API function to change job status, or the `ajschgstat` command. For details on using JP1/AJS3 - View, see *9.12 Changing the status of jobs* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using Web GUI, see *A.13 Change Job Status dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details on using the change job status API, see *7.1.10 Change job status API* in the manual *JP1/Automatic Job Management System 3 Command Reference*. Alternatively, for details on the `ajschgstat` command, see the description of `ajschgstat` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

### 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 4–8: Priority levels for automatic judgment to find a target generation

Operation	Priority			
	Highest	High	Low	Lowest
Temporarily change plan	Now rerunning <sup>#1</sup>	Now running <sup>#2</sup>	Scheduled next	Previous completed <sup>#4</sup>
Change job status, rerun	Now rerunning <sup>#1</sup>	Now running <sup>#2</sup>	Previous completed <sup>#3</sup>	Scheduled next <sup>#4</sup>
Interrupt, kill	Now rerunning <sup>#1</sup>	Now running <sup>#2</sup>	Previous completed <sup>#4</sup>	Scheduled next <sup>#4</sup>

#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 *3. Commands Used for Normal Operations* in the manual *JPI/Automatic Job Management System 3 Command Reference*.

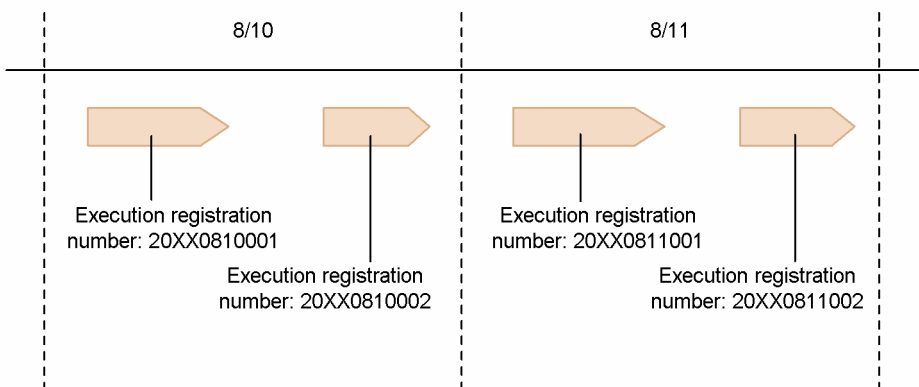
## (2) Execution registration numbers

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 4–39: Assigning execution registration numbers

●When a jobnet is executed twice a day



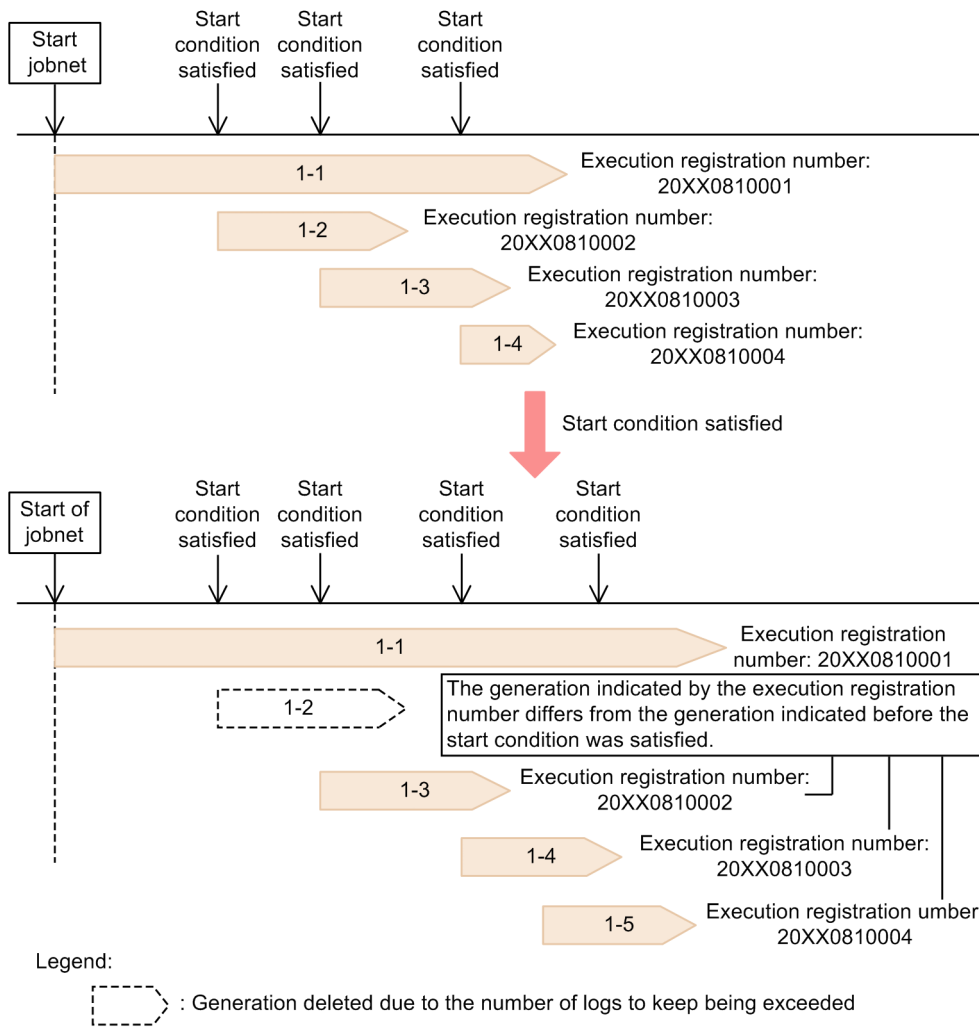
Assume that a jobnet has an execution date of August 10 20XX. Based on this information, the execution registration number of the generation that is executed first will be `20XX0810001`. The generation that is executed second will have the execution registration number `20XX0810002`.

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 4–40: 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



In the above figure, the execution registration number 20XX0810002 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 20XX0810002 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:

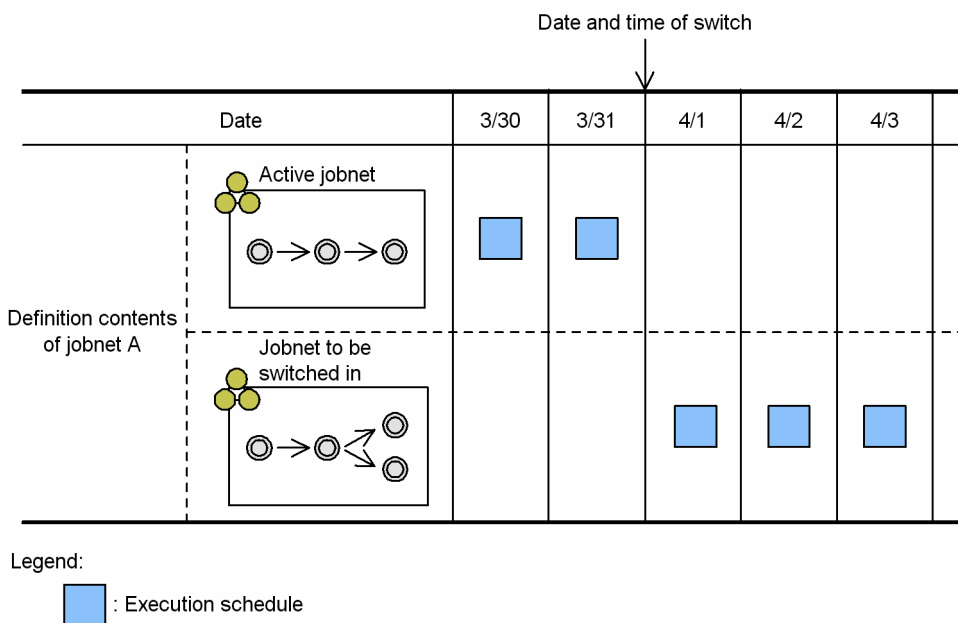
- Change a jobnet definition at a specified time without stopping the jobnet.
- Change the jobnet definition without changing the name of the jobnet.
- Set up a jobnet definition change in advance.

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 *JP1/Automatic Job Management System 3 Operator's Guide*. For the command operation, see *ajsrelease* in 3. *Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

### (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 4–41: 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

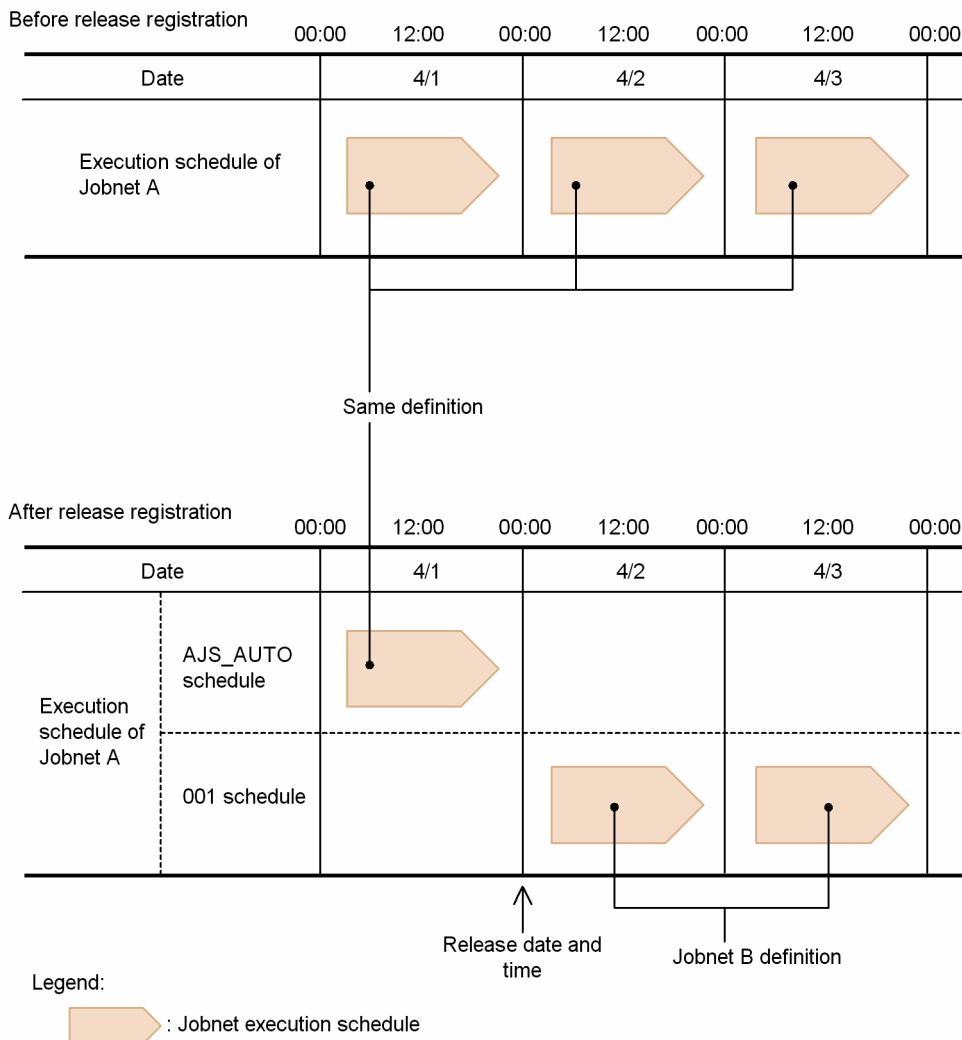
Under the jobnet release function, the jobnet definitions are distinguished by *release ID*. When you register the release source jobnet, in addition to its release ID, you specify the release date and time (for switching jobnet definitions) and the release target jobnet (the active jobnet that is to 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 7.3.8 *Using the jobnet release function* in the *JP1/Automatic Job Management System 3 Administration Guide*.

The following figure illustrates how a jobnet definition is switched using the jobnet release function.

Figure 4–42: 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 *7.3.3 Jobnet status after registration of a jobnet-definition release* in the *JPI/Automatic Job Management System 3 Administration Guide*.

For details about the jobnet release function, see *7.3 Switching a jobnet definition while the jobnet is registered for execution* in the *JPI/Automatic Job Management System 3 Administration Guide*.

## 4.5.15 Temporarily changing the wait condition settings for a jobnet or job

You can temporarily enable or disable the wait condition settings specified for a jobnet or job.

Wait conditions are used to control the execution order of jobnets and jobs that are managed by the same scheduler service. A unit with wait conditions set waits for the termination of other units called *units whose end is being waited for*, after which it can start execution. For details about wait conditions, see *2.2.5 Using wait conditions to control the order of unit execution* in the *JPI/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

When units with wait conditions set are registered for execution, the units start waiting for units whose end is being waited for to finish execution. (Multiple units whose end is being waited for can be set for one unit with wait conditions.) For a unit with wait conditions, if waiting for all units whose end is being waited for has been disabled, the unit immediately starts execution.

Also, you can re-enable waiting that you have disabled or re-enable the waiting of a unit that has completed waiting and has already ended. When a unit with wait conditions is re-executed after waiting has been enabled, the unit starts waiting again.

Whether to enable or disable wait conditions can be set individually for each of the units whose end is being waited for. If multiple units whose end is being waited for are set for one unit with wait conditions, you can enable or disable the waiting for only specific units whose end is being waited for.

For example, there might be a unit with wait conditions that you want to start when one of the units whose end is being waited for has terminated abnormally. In this case, if the unit whose end is being waited for has terminated abnormally, you can immediately start the unit with wait conditions by preventing it from waiting for the unit whose end is being waited for. Also, there might be a unit with wait conditions that abnormally terminates and you want the unit to wait for a unit whose end is being waited for again when it is rerun. To do this, you can enable waiting before the unit is rerun.

You can temporarily change wait condition settings from the GUI or by using the `ajsplan` command. For the GUI-based procedure, see *9.16 Temporarily enabling or disabling wait conditions* in the *JPI/Automatic Job Management System 3 Operator's Guide*. For the command-based procedure, see `ajsplan` in *3. Commands Used for Normal Operations* in the manual *JPI/Automatic Job Management System 3 Command Reference*.

## 4.5.16 Displaying and re-executing temporary change operations for a job or jobnet

You can display information about *Change Plan*, *Change Hold Attribute*, and other operations performed on a job or jobnet in list format for checking purposes. This information is called *temporary change information*.

Also, you can select specific operations in the Temporary Changes list that is displayed, and re-execute those operations for the job or jobnet. The re-execution of a temporary change operation is called the *re-execution of a temporary change*.

Users are able to display temporary change information and perform re-execution of temporary changes by setting `yes` for the `SAVEPLANINFO` environment setting parameter. For details about the `SAVEPLANINFO` environment setting parameter, see *20.4 Setting up the scheduler service environment* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

### (1) Displaying temporary change information

You can display temporary change information in the Temporary Changes dialog box of JP1/AJS3 - View and output the displayed information to a CSV file or the standard output by using the `ajspplanout` command. For details about how to display temporary change information, see *9.17.1 Displaying temporary changes* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details about the Temporary Changes dialog box, see *12.3.40 Temporary Changes dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*. For details about the `ajspplanout` command, see `ajspplanout` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

#### (a) Operations saved as temporary change information

The following operations can be saved as temporary change information:

- *Change Plan* operations (*Change Time*, *Execute Immediately*, *Execution Prohibited*, and *Release Change*)<sup>#</sup>
- *Change Hold Attribute* operations (*Hold* and *Hold-Release*)<sup>#</sup>
- *Change Delay Monitor* operations (*Monitor start delay*, *Monitor end delay*, and *Monitor jobnet*)<sup>#</sup>
- *Change Priority* operations<sup>#</sup>
- *Change Execution Order Method* operations (*Synchro* and *Asynchro*)<sup>#</sup>
- Disabling a start condition, changing the time the system will wait for the start condition to be satisfied, and changing the number of times the system will wait for a start condition to be satisfied<sup>#</sup>
- Temporary change operations for wait conditions (disabling and enabling a wait)<sup>#</sup>
- Adding an execution schedule
- Canceling registration by specifying the generation<sup>#</sup>

#

The execution registration number is saved as temporary change information even when the generation subject to an operation is specified by the execution ID or by the function for automatically determining the generation.

When a user performs a temporary change operation, the relevant temporary change information is saved in JP1/AJS3 - Manager. Note that the information is saved only when the temporary-change operation and management functionality is enabled by setting `yes` for the `SAVEPLANINFO` environment setting parameter. Information about temporary change operations performed before `yes` is set for `SAVEPLANINFO` is not displayed in the list.

Supplementary notes:

- If a temporary change operation is performed for multiple units by a command in which multiple unit names are explicitly specified or with wildcard characters, temporary change information is saved separately for each unit.
- If a temporary change operation is performed by the `ajspplan` command with a planning group specified, the temporary change information for the planning group is not saved. However, the temporary change information for the root jobnet automatically selected by the `-x` option is saved.
- When a user requests the displaying of the temporary change information list for a unit, JP1/AJS3 checks whether the user has view permission for the root jobnet of the unit. The list is displayed only when the user has this permission. The permissions for the unit and its subordinate units are not checked.
- When a user requests the displaying of the temporary change information list for a root jobnet registered for release, JP1/AJS3 checks whether the user has view permission for the root jobnet currently in use.

## (b) Storage period for temporary change information

The following table describes how long the temporary change operations performed for a job or jobnet are saved as temporary change information.

Table 4–9: Storage period for temporary change operations

No.	Temporary change category	Storage period
1	<i>Change Plan</i> operations	<ul style="list-style-type: none"> <li>• <i>Change Time</i> This item is saved until the scheduled start time existing before the change or the scheduled start time existing after the change, whichever is later.</li> <li>• <i>Execute Immediately</i> This item is saved until the scheduled start time of an execution schedule that is executed immediately.</li> <li>• <i>Execution Prohibited</i> For a root jobnet registered for planned execution, this item is saved until the scheduled start time of the next execution schedule, which is determined after execution is prohibited, arrives.<sup>#</sup> For other root jobnets, this item is saved until the scheduled start time of an execution schedule whose execution is prohibited arrives.</li> <li>• <i>Release Change</i> This item is saved until the storage period for the released temporary change operation expires.</li> </ul>
2	<i>Change Hold Attribute</i> operations	This item is saved until the scheduled start time of a temporarily changed execution schedule arrives.
3	<i>Change Delay Monitor</i> operations	
4	<i>Change Priority</i> operations	
5	<i>Change Execution Order Method</i> operations	
6	Disabling a start condition, changing the time the system will wait for a start condition to be satisfied, and changing the number of times the system will wait for a start condition to be satisfied	
7	Temporary change operations for wait conditions	
8	Adding an execution schedule	This item is saved until the scheduled start time of the added execution schedule arrives.

No.	Temporary change category	Storage period
9	Canceling registration by specifying the generation	This item is saved until the scheduled start time of an execution schedule whose execution registration is canceled arrives.

#

For a root jobnet registered for release, if execution of the execution schedule immediately before a release is prohibited, temporary change information is saved until the scheduled start time of the execution schedule whose execution is prohibited arrives.

However, the storage period for temporary change information changes depending on whether the schedule of the root jobnet is based on 24-hour system or 48-hour system.

- For a schedule based on 24-hour system  
Temporary change information is saved for 24 hours beginning with the base time for the storage period for temporary change information (if the base time is 0:00:00, the information is saved until 23:59:59).
- For a schedule based on 48-hour system  
Temporary change information is saved for 48 hours beginning with the base time for the storage period for temporary change information (if the base time is 0:00:00, the information is saved until 47:59:59).

If multiple temporary change operations have been performed for a temporarily changed execution schedule, temporary change information is saved until all the storage periods for these operations have expired.

### (c) Notes

- Be careful if the local time for the scheduler service has been changed by using the `ajslocaldate` command. If the time has been changed, the temporary change information output by the `ajsplanout` command and the information displayed by choosing **Temporary Changes** in the JP1/AJS3 - View window or another window might be different.

The current time used as the base for the start date when the `ajsplanout` command is executed without the `-b` option specified differs from the current time used as the base when **Temporary Changes** is chosen. The following describes the differences:

- When the `ajsplanout` command is executed without the `-b` option specified  
The current time is the local time of the scheduler service that manages the specified jobnet.
- When **Temporary Changes** is chosen  
The current time is the time of the system in which JP1/AJS3 - View is running.

The Temporary Changes dialog box can simultaneously display the temporary change information for multiple scheduler services. The system time is used so that the base time does not depend on the scheduler service.

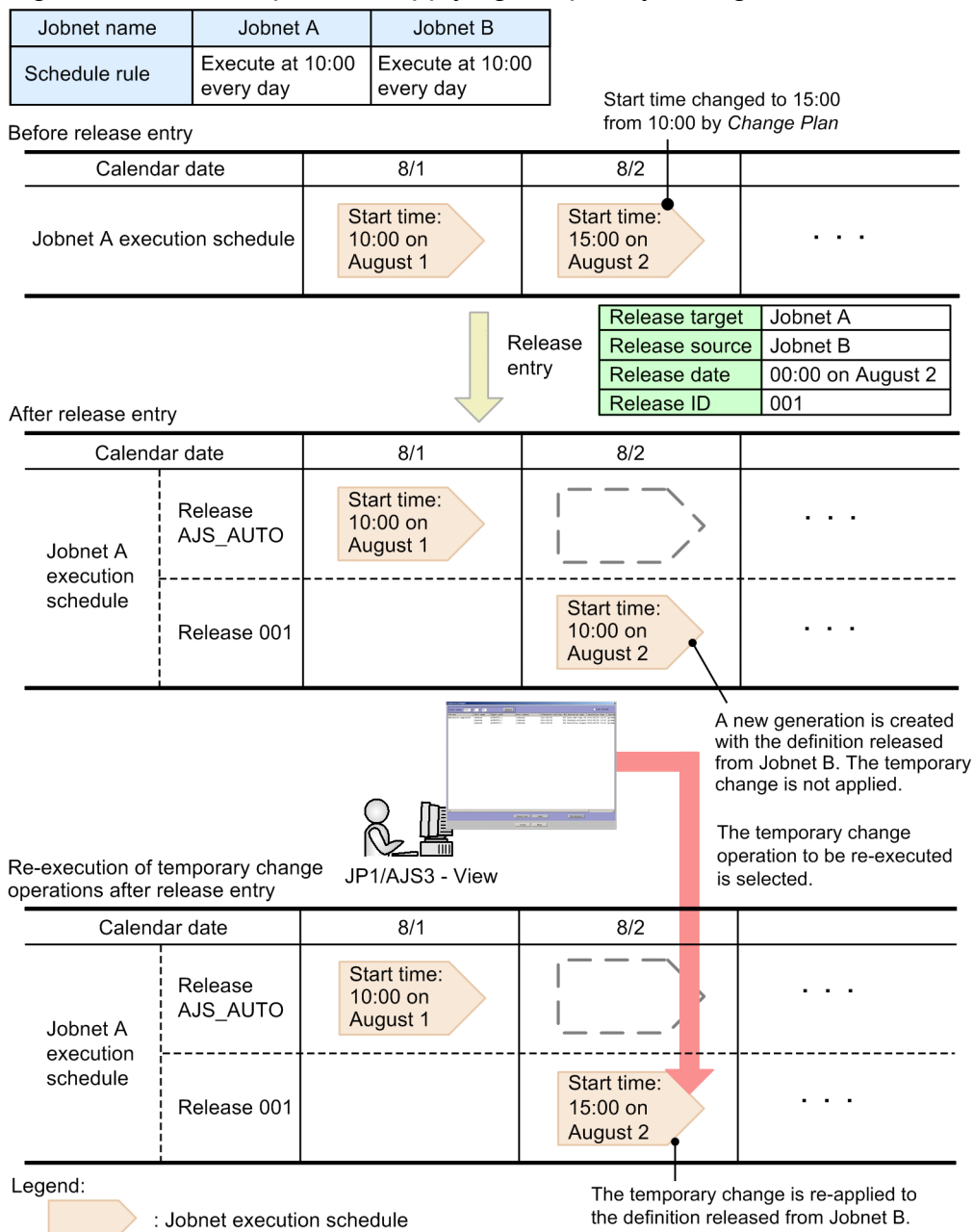
- Temporary change information is data that is stored in a directory for job error information. The required space increases every time a temporary change operation is performed. Note that when a root jobnet and its lower units have expired temporary change information saved, and on that jobnet, you perform an operation that was saved as temporary change information, the expired temporary change information is automatically deleted. For details about how to estimate required disk space, see *3.2.4 Estimating disk capacity* in the *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

## (2) Re-execution of a temporary change

The temporary change operations selected in the Temporary Changes dialog box can be re-executed by clicking the **Re-execute** button. Re-execution of a temporary change is useful when a jobnet definition that will be switched by the jobnet release function has been temporarily changed before release of the jobnet definition is registered (release entry). In such cases, the temporary changes can be re-applied to the jobnet after the release entry has been performed.

The following figure shows an example in which temporary changes made before release entry are re-applied after release entry.

Figure 4–43: Example of re-applying temporary changes after release entry



For details about the procedure for re-executing temporary change operations, see 9.17.2 *Re-executing temporary changes* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

**(a) Execution schedule for which temporary change operations are re-executed**

In JP1/AJS3, when changing a jobnet definition temporarily, the target execution schedule can be specified by any of the following:

- Execution ID
- Automatic determination
- Execution registration number

However, when re-applying the temporary changes, regardless of how the generation to which the changes were originally made was specified, the execution registration number is always used to specify the re-application target generation. Even if the generation to which temporary changes were originally made is specified by the execution ID or is determined automatically, the generation's execution registration number is obtained by translation. This number becomes the basis for specifying the generation to which temporary changes are re-applied.

The reason for this is that the following problems would arise if the target generation were specified by the execution ID or were determined automatically.

**If the target generation were specified by the execution ID**

When the jobnet definition of an execution schedule is switched by the jobnet release function, a new execution schedule is created with the new definition and is assigned a new execution ID.

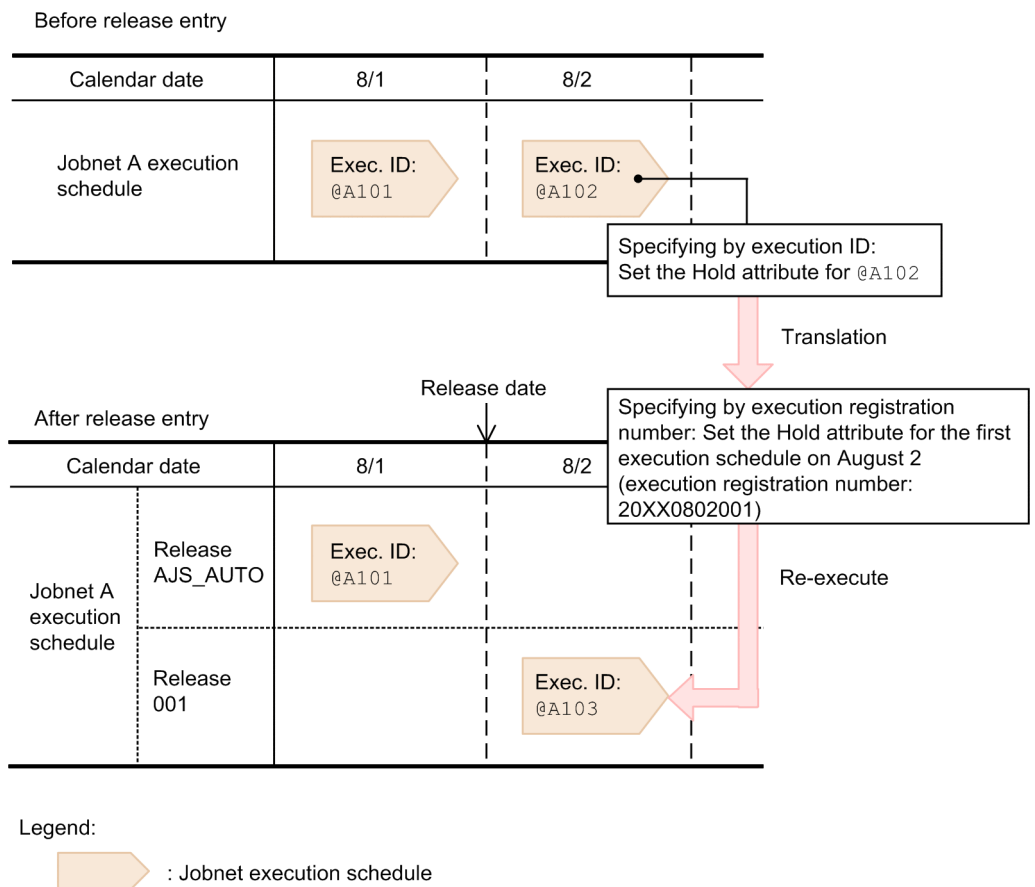
If a new execution ID is assigned, the new execution schedule no longer has the execution ID that was specified when a temporary change operation was performed. Consequently, the execution schedule to which temporary changes are to be re-applied can no longer be determined.

**If the target generation were determined automatically**

The automatic generation determination function uses the current status of the execution schedule as the basis for determinations. If the jobnet definition of an execution schedule is switched by the jobnet release function, the status of the execution schedule changes. Accordingly, an unintended execution schedule might be automatically determined as the target for re-applying temporary changes after the jobnet definition is switched (release entry).

The following figure describes how the execution schedule to which temporary changes will be re-applied is determined.

**Figure 4-44: How the execution schedule to which temporary changes will be re-applied is determined**





In this example, the Jobnet A execution schedule to be executed on August 2 has execution ID @A102, and the Hold attribute is set for the generation as a temporary change by specifying the execution ID. When Jobnet A is registered for release and jobnet release is performed for the execution schedule on and after August 2, the execution ID is translated to the execution registration number. For this reason, if the execution ID for the execution schedule to be executed on August 2 changes from @A102 to @A103, the execution schedule is identified from the execution registration number as the target for re-applying temporary changes.

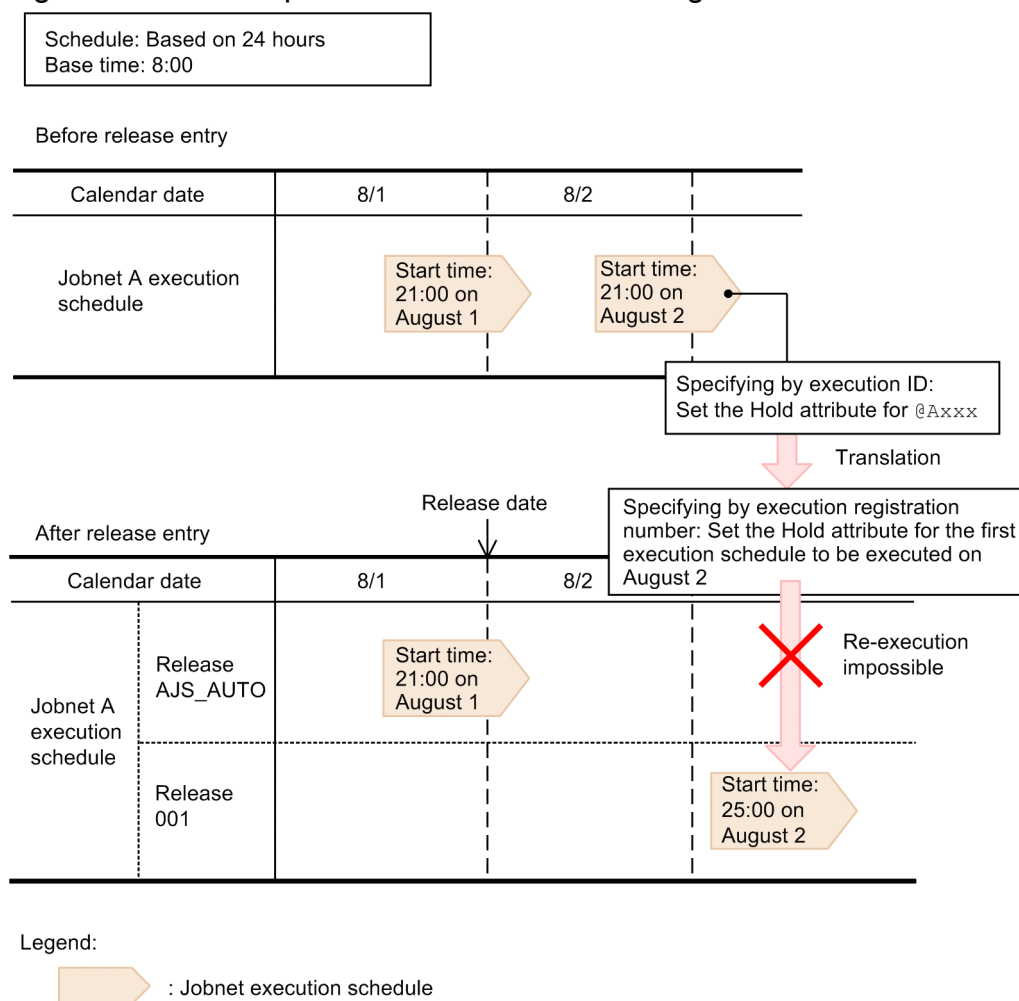
Note, however, that the intended execution schedule sometimes does not become the target for re-applying temporary changes even if the translated execution registration number is used. The following explains cases in which temporary changes are not re-applied to the intended execution schedule.

When the setting for treating the execution registration number as a calendar date is used

When the setting for treating the execution registration number as a calendar date is used, temporary changes might not be re-applied to the target execution schedule after release entry.

The following figure shows an example in which the execution registration number is treated as a calendar date and the temporary changes cannot be re-applied.

Figure 4–45: Example in which the execution registration number is treated as a calendar date



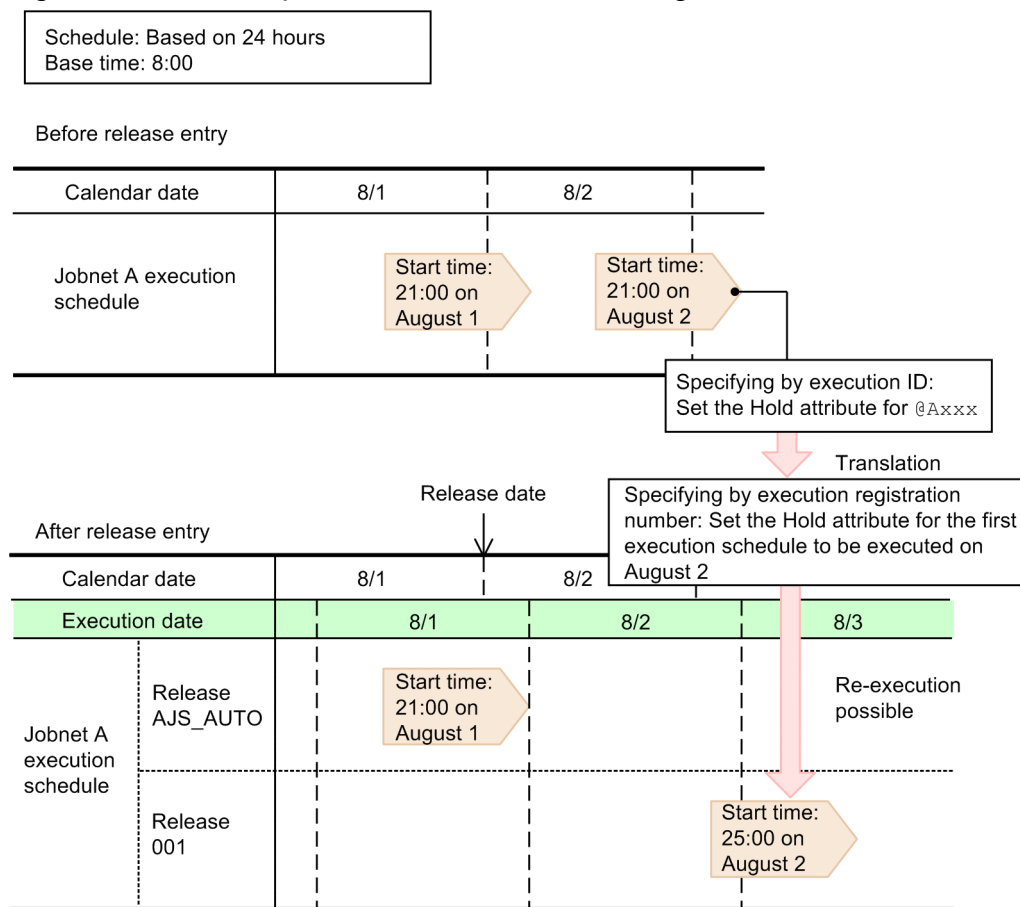
In this example, the Jobnet A schedule is based on 24 hours and has a base time of 8:00. To allow the Jobnet A execution schedule to be executed on August 2, the Hold attribute is set as a temporary change by specifying the execution ID. When Jobnet A is registered for release and jobnet release is performed for the execution schedule on and after August 2, the execution ID is translated to the execution registration number. In this example, the translated execution registration number means the first execution schedule to be executed on August 2. However,

because the new start time of the execution schedule on August 2 is 25:00, which is actually 1:00 on August 3, the temporary changes cannot be re-applied with the execution registration number treated as a calendar date.

In cases such as this, temporary changes can be re-applied by changing the setting so that the execution registration number is treated as an execution date.

The following figure shows an example in which the execution registration number is treated as an execution date.

**Figure 4–46: Example in which the execution registration number is treated as an execution date**



Legend:

: Jobnet execution schedule

If the execution registration number is treated as an execution date, 25:00 on August 2 is treated as a time on August 2, not a time on August 3. Accordingly, because an execution schedule to be executed at 25:00 on August 2 is treated as an execution schedule of August 2, temporary changes can be re-applied to the execution schedule after the jobnet definition is released.

Note that whether to treat the execution registration number as a calendar date or an execution date is set by using the EXECREGISTRATIONNUMBER environment setting parameter.

Notes on using the EXECREGISTRATIONNUMBER environment setting parameter

- In JP1/AJS3 upgraded from version 09-10 or earlier, EXECREGISTRATIONNUMBER is set to calendar (calendar date) by default. We therefore recommend that you change the setting to execution (execution date).
- If you change the setting of the EXECREGISTRATIONNUMBER environment setting parameter during JP1/AJS3 operation, cold-start the scheduler service. If you do not do this, there might be a mismatch between the execution registration numbers in the temporary change information displayed in JP1/AJS3 - View or

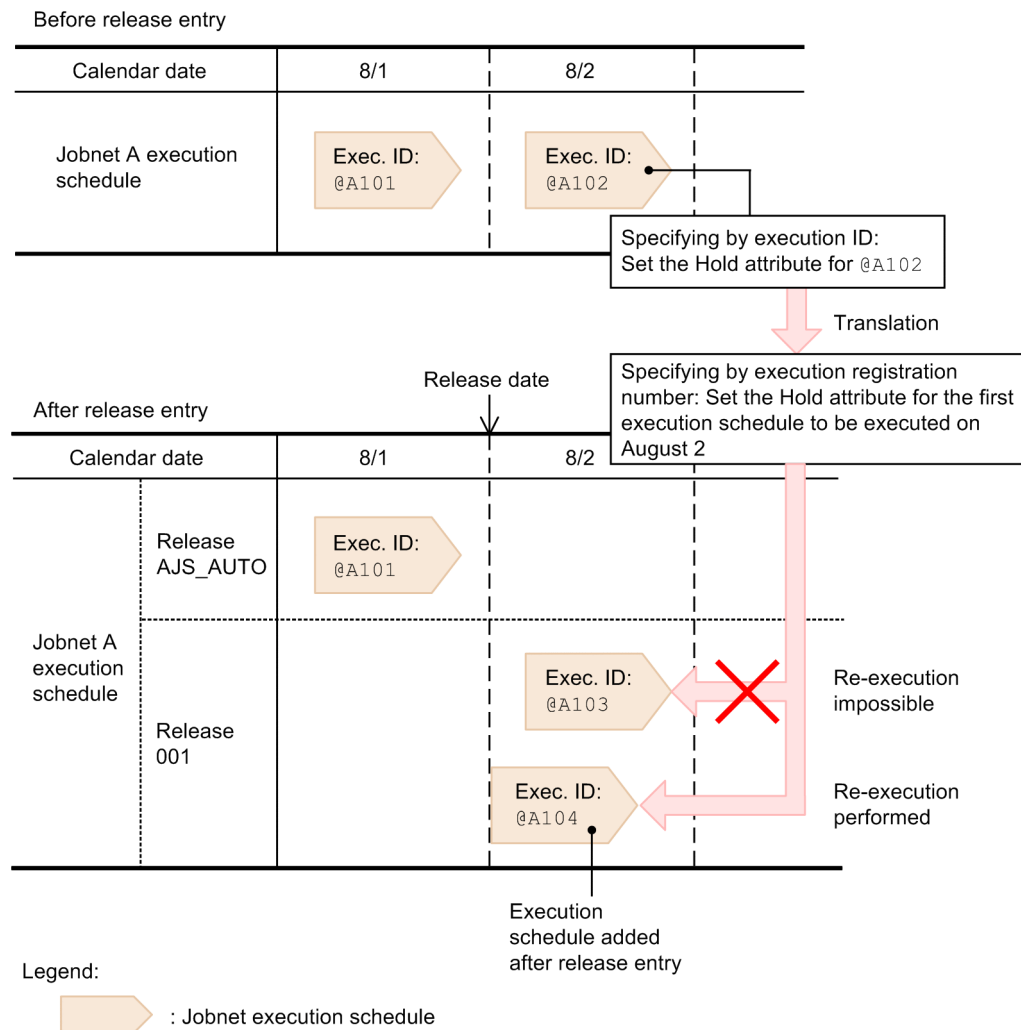
by a command and the execution registration numbers that JP1/AJS3 holds. Any such mismatch prevents temporary change operations from being re-executed correctly.

When the order of execution schedules changes because of an addition or deletion

If the order of execution schedules is changed by adding or deleting execution schedules, you might not be able to re-apply temporary changes to the intended execution schedule.

The following figure shows an example in which addition or deletion changes the order of execution schedules changes.

Figure 4–47: Example in which the order of execution schedules changes because of an addition or deletion



In this example, the Jobnet A execution schedule to be executed on August 2 has execution ID @A102, and the Hold attribute is set for the schedule as a temporary change. The execution ID is used to identify the execution schedule. When Jobnet A is registered for release and a jobnet release is performed for the execution schedule on and after August 2, the execution ID is translated to the execution registration number. According to the execution registration number, the first execution schedule to be executed on August 2 becomes the target for re-applying temporary changes. However, if execution schedule @A104 is added, because its start time is earlier than the start time of execution schedule @A103, execution schedule @A104 becomes the first execution schedule to be executed on August 2. As a result, temporary changes are re-applied to execution schedule @A104 rather than @A103.

If you add or delete schedule rules after release entry, you must verify that temporary changes will be re-applied as you intend, and that no problems exist as a result of re-applying temporary changes.

## (b) Notes

The notes below apply to re-execution of temporary changes. Before you re-execute temporary change operations from the Temporary Changes dialog box, read the notes and confirm that no problems will occur as a result of re-execution.

Note that if an error occurs during re-execution of temporary changes, the Temporary Change Re-execution Processing Results dialog box displays information about the error. For details about the Temporary Change Re-execution Processing Results dialog box, see *12.3.41 Temporary Change Re-execution Processing Results dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

### ■ Notes on changing a jobnet definition

- Do not change the name of a unit for which re-execution will be performed. If you do so, re-execution results in an error.
- Do not delete a unit for which re-execution will be performed. If you do so, re-execution results in an error.

### ■ Notes on changing a schedule definition

- Do not add or delete schedule rules for a jobnet for which re-execution will be performed. If you do so, re-execution of temporary changes will not be performed correctly or will result in an error.
- Do not change the execution date in the schedule definition of a jobnet for which re-execution will be performed. If you do so, re-execution will not be performed correctly for the target execution schedule or will result in an error.

### ■ Notes on re-execution of a temporary change

- If you make temporary changes to the jobnet definition of an execution schedule that has already been executed, do not use the temporary-change re-execution function. If you re-execute a temporary change for an execution generation that has been executed in the past, an error might occur.
- If the immediate execution of a temporary change is re-executed, temporary changes are immediately re-applied.
- If you make a time change to a different day that also spans the current time, do so manually without using re-execution of a temporary change. Re-execution of a temporary time change that spans the current time results in an error.
- Operations for execution generations of a jobnet with a start condition are not saved.
- Operations for generations whose status is *Not sched. to exe* are not saved.
- For the execution schedules listed below, do not re-execute a temporary change. If you want to temporarily change these execution schedules, do so manually.
  - Execution schedules to be executed on the day specified as the release date for a jobnet registered for release by specifying a time other than the base time as the release time
  - (When the release entry of a jobnet registered for release by specifying a time other than the base time as the release time is canceled) Execution schedules to be executed on the day specified as the release date before the release entry is canceled

For example, if the base time is 0:00 and a jobnet is registered for release by specifying 9:00 on July 10 as the release date, the execution schedule to be executed on July 10 applies. In this case, you cannot use re-execution of a temporary change for this execution schedule. To temporarily change this execution schedule, do so manually.

- Do not perform re-execution of a temporary change for the same unit from multiple instances of JP1/AJS3 - View at the same time. If you do so, the same temporary changes will be re-applied more than once or temporary changes that were made into an unintended execution schedule will be re-applied.
- When you use the `ajsleave` command to cancel registration, you can specify the target generation by using either the execution ID or the execution registration number. If you choose to use the execution registration number for specifying the target generation, you can use one of the following specification formats:

- all
- schedule
- result
- YYYYMMDD#
- YYYYMMDDNNN#

#

YYYY: year, MM: month, DD: day, NNN: generation sequence number on the execution date

If you specify all, schedule, or result, temporary change information will not be saved.

If you specify a string in YYYYMMDD format, the temporary change information for all generations that are executed on the specified day is saved.

Note that when you cancel registration from JP1/AJS3 - View, you can only perform the cancellation by specifying a period or by canceling all generations. Accordingly, temporary change information is not saved.

- Temporary change information for the 1st to 999th generations of a jobnet can be saved for one day. If an attempt is made to save temporary change information for 1,000th and following generations, the KAVS4671-E message is output to the integrated trace log, and the temporary change information is not saved. Ensure that no more than 999 generations of a jobnet will be executed per day.

#### ■ Other notes

- Do not change the value of the TZ environment variable during JP1/AJS3 operation. If the time zone setting used when temporary change was performed and the time zone setting used when the temporary change is re-executed do not match, re-execution of a temporary change might be performed incorrectly or might result in an error.
- Do not create execution schedules whose TZ environment variable values in the same root jobnet are different. If the execution dates of these execution schedules are the same, it might be difficult to guess their execution registration numbers because the time at which the day starts is not the same.
- Do not change the base time during JP1/AJS3 operation. If the base time used when temporary change was performed and the base time used when the temporary change is re-executed do not match, re-execution of a temporary change might be performed incorrectly or might result in an error.

If you need to change the base time during operation, make sure you cold-start the scheduler service.

- Re-execution of a temporary change is performed as the JP1 user who performs the re-execution, not as the JP1 user who performed the temporary change. Therefore, before you re-execute a temporary change for a unit, make sure you are a JP1 user that has the appropriate permission for the unit. If you do not have required permission, re-execution will result in an error.
- If the temporary change operations you are re-executing include the addition of a root jobnet execution schedule (the `ajsentry` command with the `-d` or `-t` option specified), pay careful attention to the OS user mapped to the JP1 user.

If the execution-user type of a job that will be executed by an added execution schedule is *User who registered*, re-execution is performed as the OS user mapped to the JP1 user. Therefore, if the relevant OS user does not have required permission for OS resources such as executable files, an error occurs during execution of the job.

## 4.5.17 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 *JP1/Automatic Job Management System 3 Operator's Guide*. For the command operation, see *ajssuspend* in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## **(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
- Changing a wait condition
- Deleting an existing unit
- Changing the map size
- Adding a start condition
- Deleting 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 4–10: Status of a unit added during suspension

Status of unit one level above added unit	Status of added unit
<b>Waiting</b>	<b>Not scheduled</b>
<b>Now running</b>	<b>Not scheduled</b>
<b>Ended</b>	<b>Bypassed</b>

## 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.

## Changing a wait condition

You can change a wait condition. However, note the following:

- If you suspend a unit with wait condition and change the wait condition when the status of the unit is *Now running*, the new change is not applied. When the suspension is released, the unit continues execution with the old wait condition.
- While execution of a unit with wait condition is suspended, if the unit whose end is being waited for terminates, the result is a wait condition that is not satisfied immediately. If the unit whose end is being waited for has terminated when suspension is released, the result is a wait condition that has been satisfied.
- While execution of a unit whose end is being waited for is suspended, the associated unit with wait condition does not terminate abnormally even if the definition of the unit whose end is being waited for is not correct. If the definition of the unit whose end is being waited for is incorrect when the suspension is released, the associated unit with wait condition terminates abnormally.

## 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 `ajsshows` command, before deleting the unit. For the `ajsshows` command syntax, see `ajsshows` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.
- 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, 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. Note that the schedule recalculation method differs according to the execution registration method. Therefore, the time when a start condition added to a jobnet is enabled also depends on the execution registration method.

The time when an added start condition is enabled is as follows:

- When the jobnet is registered for immediate execution  
The added start condition is enabled for generations created after suspension is released. The added start condition is not enabled for any generations created before the suspension.  
To enable added start conditions for generations created before the jobnet execution is suspended, use either of the following methods:
  - Cancel execution registration of the jobnet, add start conditions, and then re-register the jobnet for execution.
  - When there are generations in *Being held* status, suspend execution of the jobnet, add start conditions, release suspension, and then execute the `ajspplan` command for the generations created before suspension to enable the start conditions.
- 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  
The added start condition is enabled for generations created after suspension is released. The added start condition is not enabled for any generations created before the suspension.  
To enable added start conditions for generations created before the execution of the jobnet is suspended, use either of the following methods:
  - Cancel execution registration of the jobnet, add start conditions, and then re-register the jobnet for execution.
  - Add start conditions to cancel suspension, and then execute the `ajspplan` command for the generations created before suspension.
- 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. The added start condition is not enabled for any generations created before the suspension.  
To enable added start conditions for generations created before the jobnet execution is suspended, use either of the following methods:
  - Cancel execution registration of the jobnet, add start conditions, and then re-register the jobnet for execution.
  - Add start conditions to cancel suspension, and then execute the `ajspplan` command for the generations created before suspension.

Note that, if you add start conditions to a root jobnet, an error occurs when suspension is canceled.

For details on the `ajspplan` command, see *ajspplan* in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## Deleting start conditions

You can delete start conditions that have been defined (`.CONDITION`). Note, however, that you might be unable to delete start conditions depending on the execution registration method. This is because the schedule re-calculation method differs according to the execution registration method. If you cannot delete start conditions, you can disable them.

The time when a start condition is deleted or disabled is as follows:

- When the jobnet is registered for immediate execution  
When start conditions are deleted from a jobnet, they are not added to any jobnet generations created after suspension is released. In the case of monitoring generations created before suspension, start conditions are not deleted, and the status of the monitoring generations changes to *Monitor terminated*.



Although you cannot delete start conditions from generations that were created before suspension, you can disable the start conditions. To disable the start conditions, suspend the jobnet when there are generations in *Being held* status, delete the start conditions, release the suspension, and then use the `ajspplan` command to disable the start conditions.

- When the jobnet is registered for planned execution  
The deleted start conditions are disabled, starting from the next generation that has been scheduled for execution.
- When the jobnet is registered for fixed execution in a specified period  
When start conditions are deleted from a jobnet, they are not added to any jobnet generations created after suspension is released. In the case of monitoring generations created before suspension, start conditions are not deleted, and the status of the monitoring generations changes to *Monitor terminated*.  
To delete or disable the start conditions from generations created before the jobnet execution is suspended, use either of the following methods:
  - Cancel the execution registration of the jobnet, delete the start conditions, and then re-register the jobnet for execution.
  - Delete the start conditions, release suspension, and then use the `ajspplan` command to disable start conditions for the generations that were created before suspension.
- When the jobnet is registered for fixed execution for a specific number of future generations  
When start conditions are deleted from a jobnet, they are not added to any jobnet generations created after suspension is released. In the case of monitoring generations created before suspension, start conditions are not deleted, and the status of the monitoring generations changes to *Monitor terminated*.  
To delete or disable the start conditions from generations created before the jobnet execution is suspended, use either of the following methods:
  - Cancel the execution registration of the jobnet, delete the start conditions, and then re-register the jobnet for execution.
  - Delete the start conditions, release suspension, and then use the `ajspplan` command to disable start conditions for the generations that were created before suspension.

For details on the `ajspplan` command, see *ajspplan* in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

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 4–11: 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

Edit operation		Unit		
		Job	Jobnet	
Editing a relation line	Adding a relation line		Y	Y
	Deleting a relation line		Y	Y
Editing a wait condition	Changing units whose end is being waited for	Adding	Y#1	Y#1
		Deleting	Y#1	Y#1
	Changing the wait method		Y#1	Y#1
	Changing the behavior if there is no generation to wait for		Y#1	Y#1
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#1,#2	Y#1,#2
Other attribute		Y	Y	
Other	Changing the map size		--	Y

Legend:

Y : Available

N : Not available

-- : Not applicable

#1

This edit operation can be performed during suspension.

#2

This edit operation can only be performed from JP1/AJS3 - View.

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 4–12: Editing options available for units according to status

Edit operation		Unit status		
		Wait for prev. to end/Wait for start time	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

Edit operation			Unit status		
			Wait for prev. to end/Wait for start time	Running	Ended
Editing a wait condition	Changing units whose end is being waited for	Adding	Y#1	Y#1	Y#1
		Deleting	Y#1	Y#1	Y#1
	Changing the wait method		Y#1	Y#1	Y#1
	Changing the behavior if there is no generation to wait for		Y#1	Y#1	Y#1
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#1,#2	N	Y#1,#2
Other attribute		Y	Y	Y	
Other	Changing the map size (jobnet)		Y	Y	Y

Legend:

Y : Available

N : Not available

-- : Not applicable

#1

This edit operation can be performed during suspension.

#2

This edit operation can only be performed from JP1/AJS3 - View.

Table 4–13: 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
		Existing unit	--	N	--	N
	Changing a comment		Y	Y	--	Y
	Changing an execution host		Y	Y	--	Y

Edit operation		Status of start condition			
		No start condition	Waiting for prev. to end	Monitoring	Ended
Editing unit information	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. 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 *JP1/Automatic Job Management System 3 Command Reference*.

To activate the suspend function, execute the command as follows:

```
ajssetup -F scheduler-service-name -m
```

You can enable the suspend function without stopping JP1/AJS3 services or scheduler services. The setting takes effect immediately. However, after you enable the suspend function, you will need to log in to JP1/AJS3 - View before you can perform operations in JP1/AJS3 - View.

Cautionary note

Once you activate the suspend function using the `ajsetup -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 3. *Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

#### (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 4–14: 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.

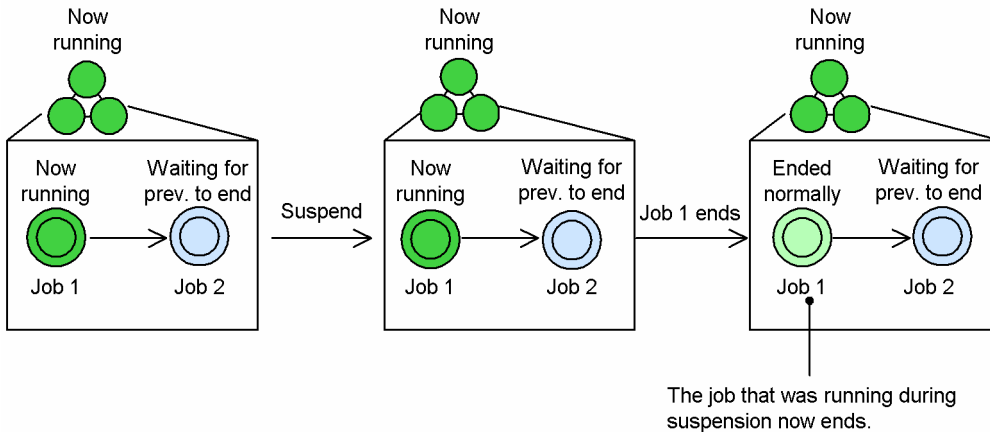
## (4) Status change of suspended jobs and jobnets

### (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.

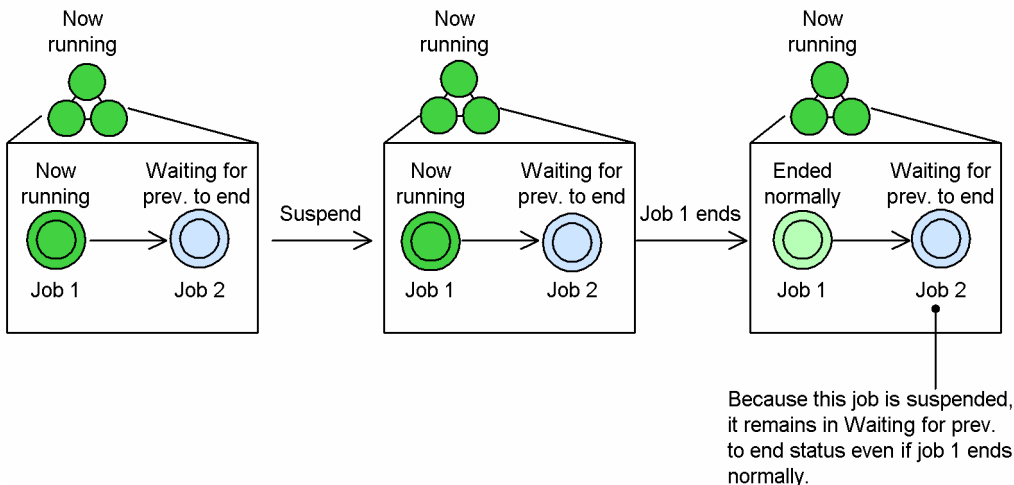
Figure 4–48: Status change of suspended job 1 (in "Now running" status)



- 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.

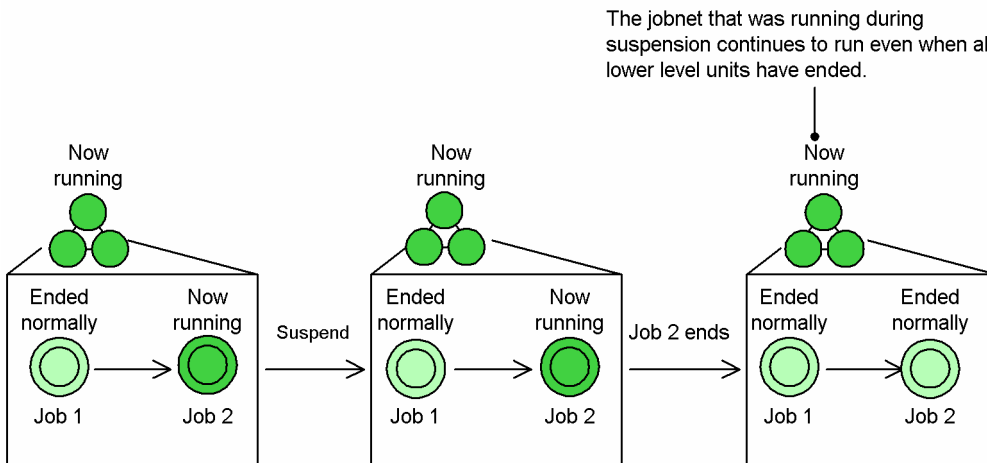
Figure 4–49: Status change of suspended job 2 (in "Waiting for previous to end" status)



### (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 4–50: 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.

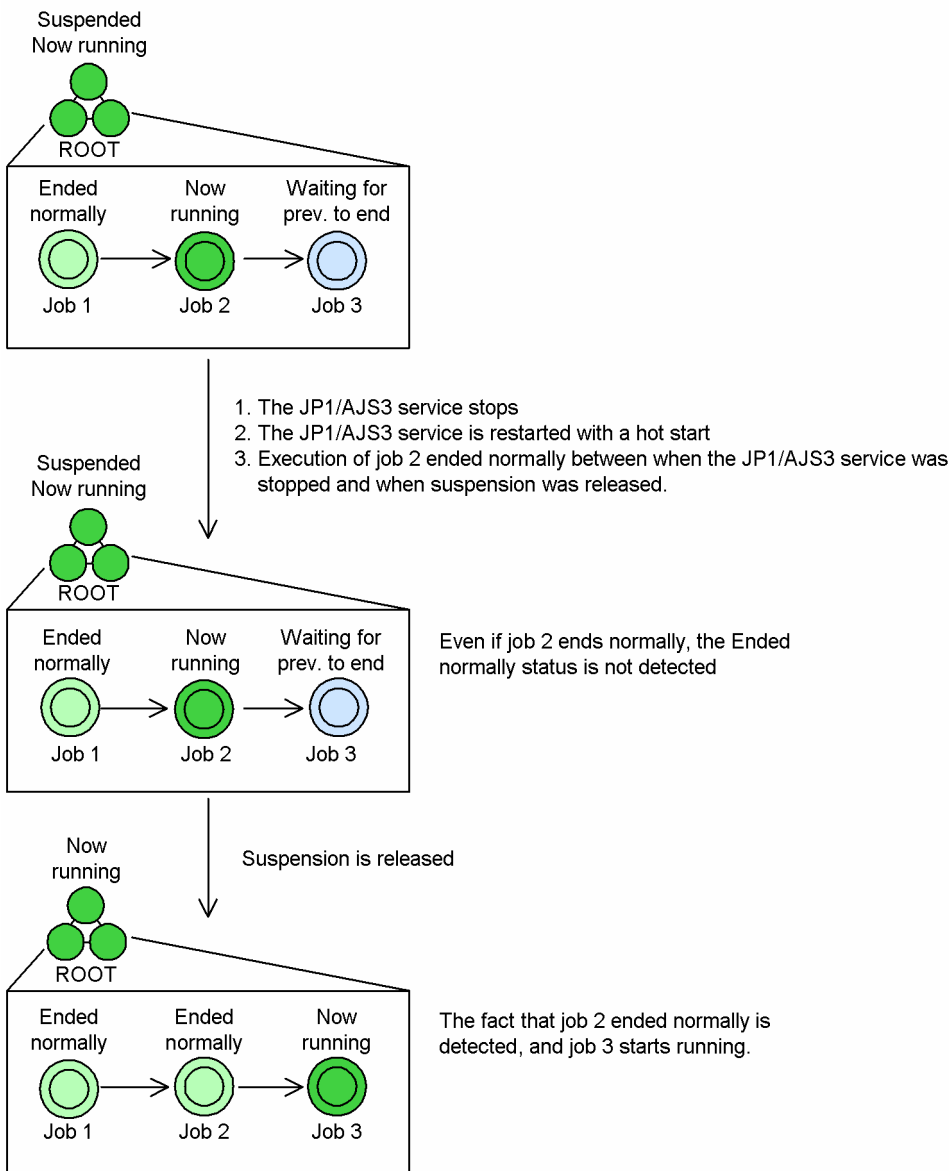
### (d) Status change when the JP1/AJS3 service is hot-started during suspension

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.



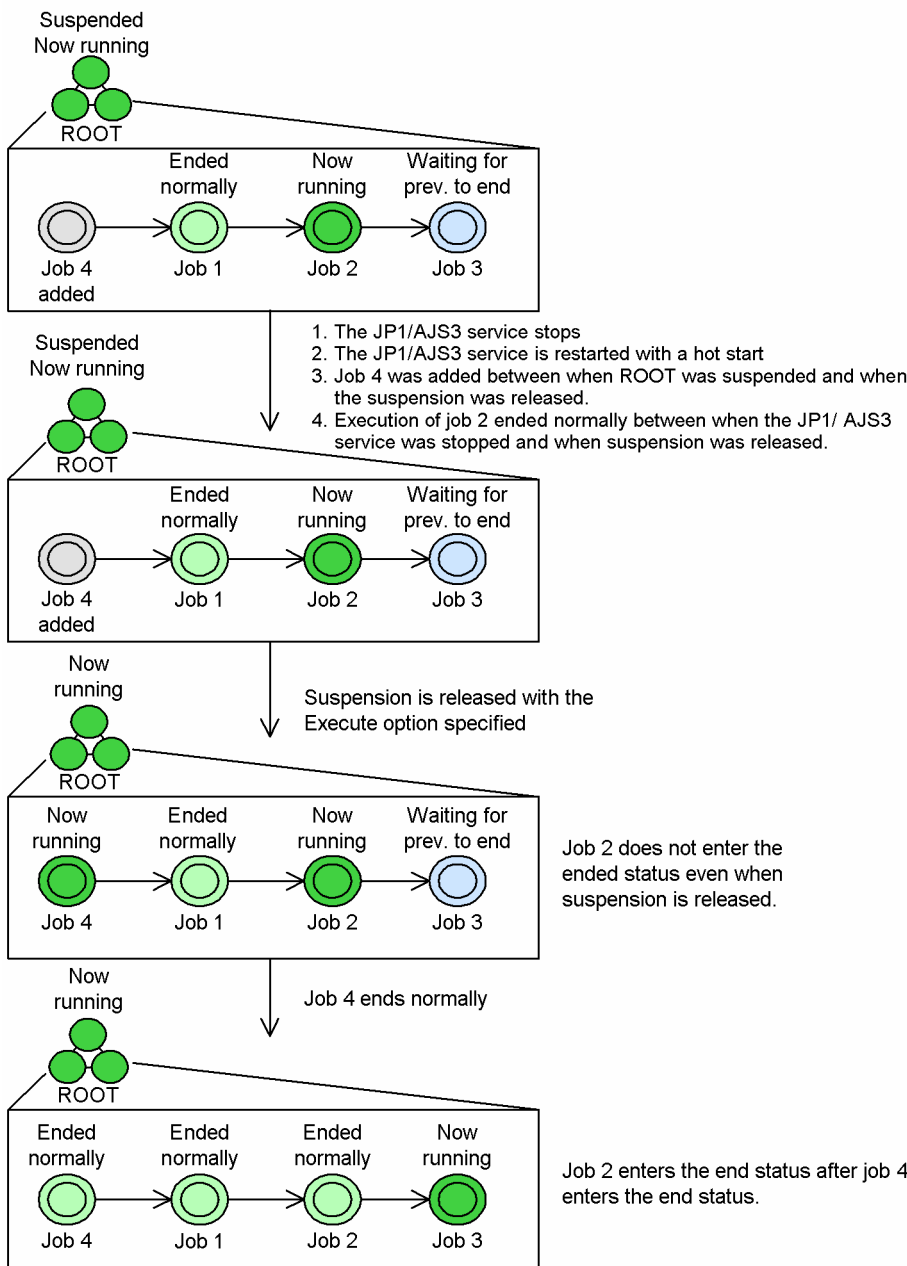
Figure 4–51: Status change when the JP1/AJS3 service is hot-started during suspension



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 4–52: Status change when the JP1/AJS3 service is hot-started during suspension (Execute option specified)



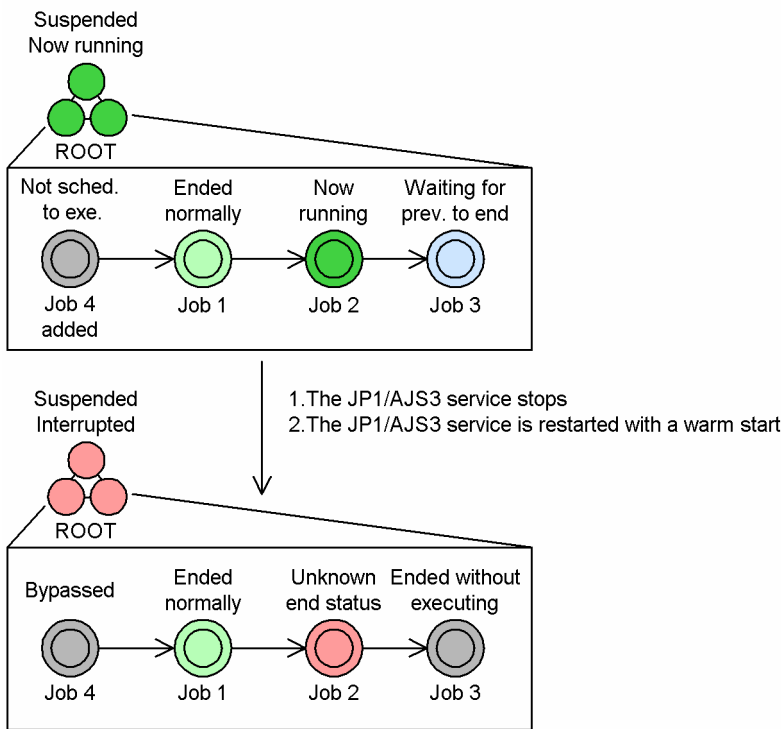
### (e) Status change when the JP1/AJS3 service is warm-started during suspension

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 4–53: 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.

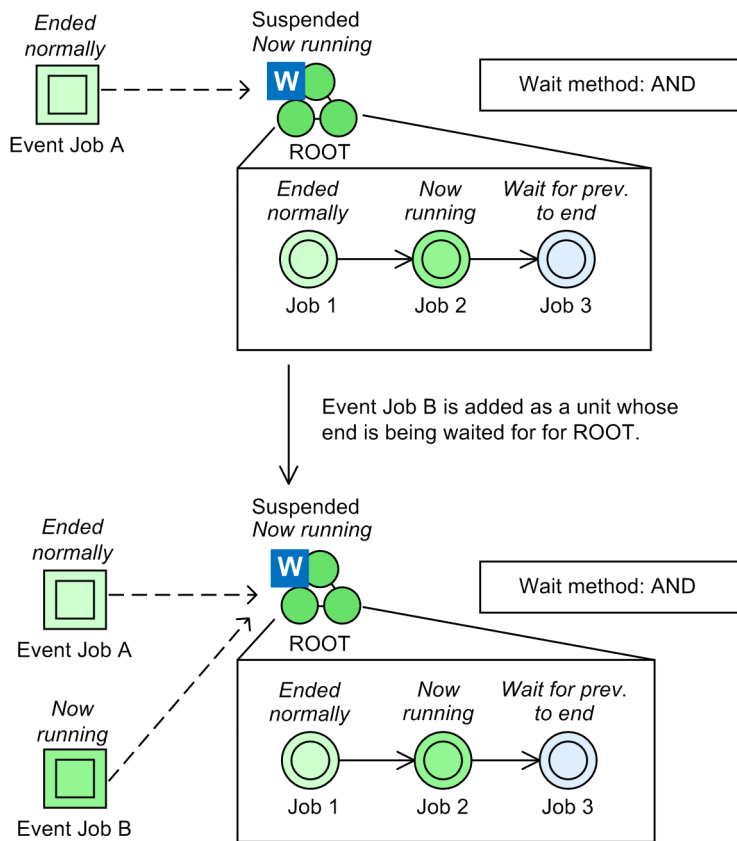
```
ajssuspend -U -R -T /
```

## (5) Changing the wait condition of a unit while execution of the unit is suspended

If the wait condition of a unit is changed while execution of the unit is suspended, the behavior of the unit is determined by the status of the unit.

- If the unit with wait condition has already started or ended execution
  - The changes made to the wait condition settings while execution of the unit is suspended are applied the next time the unit is executed (or re-executed).
  - The following figure shows an example of changing the wait condition settings of a unit while execution of the unit is suspended.

Figure 4–54: Example of changing the wait condition of a unit while execution of the unit is suspended



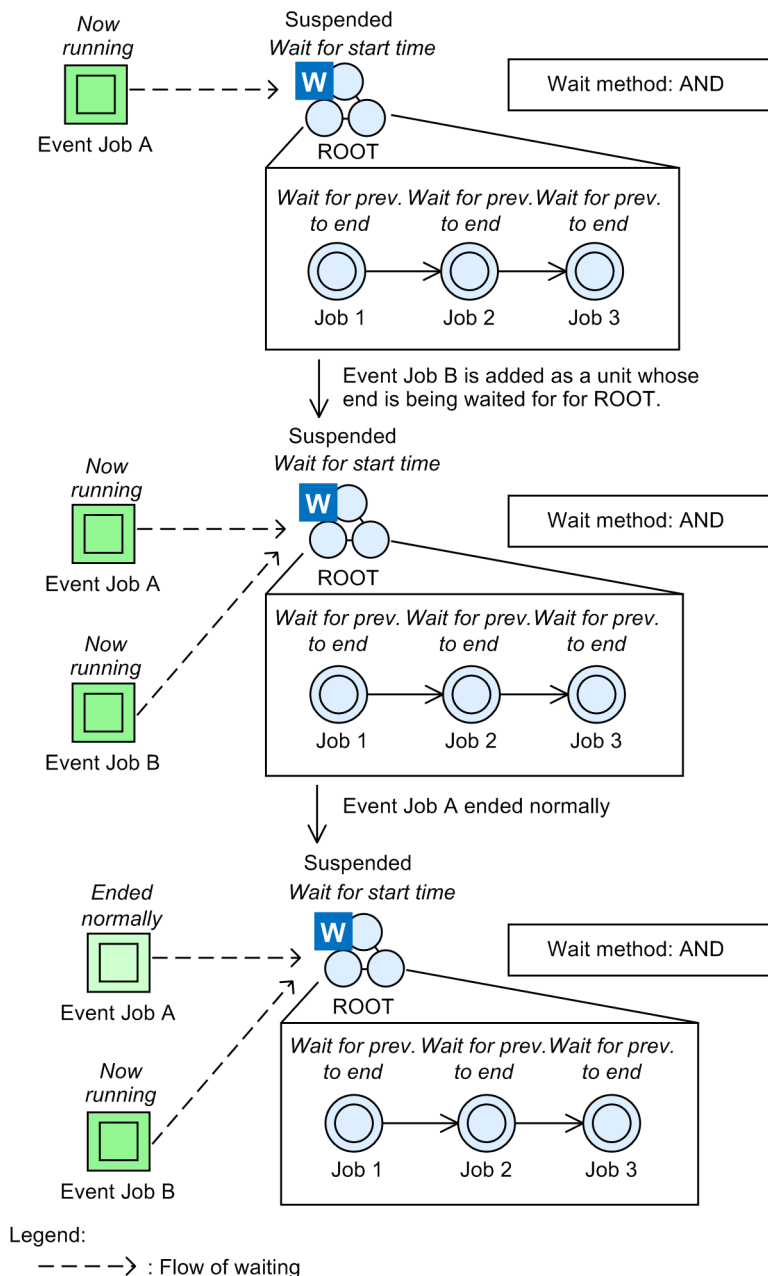
Legend:

-----> : Flow of waiting

In this example, the ROOT unit whose wait condition has already been satisfied is suspended, and Event Job B is added as a unit whose end is being waited for. Because ROOT has already started execution, it continues execution without waiting for Event Job B to end.

- If the status of the unit with wait condition is *Wait for prev. to end* or *Wait for start time*  
The unit continues to wait based on the wait condition settings as changed while the unit is suspended.  
The following figure shows an example of changing the wait condition settings of a unit while the status of the unit is *Wait for start time*.

Figure 4–55: Example of changing the wait condition settings of a unit while the status of the unit is "Wait for start time"



In this example, the ROOT unit with wait condition that has not been satisfied yet is suspended, and Event Job B is added as a unit whose end is being waited for. Because the wait condition has not been satisfied yet, when Event Job B is added, ROOT also waits for Event Job B to end. AND is set as the wait method so the wait condition is not satisfied when only Event Job A ends. ROOT continues to wait until Event Job B ends.

## (6) Scheduling after suspension is released

### (a) Allocating a schedule when the suspension is released

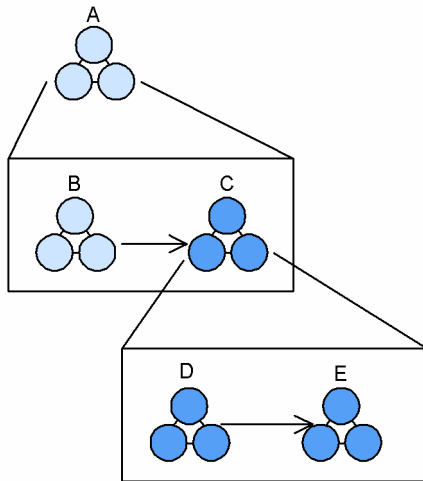
The allocation of a schedule to a unit after the suspension is released is described below, according to the category of execution registration that applies to the unit.

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 4–56: 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 4–15: 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:

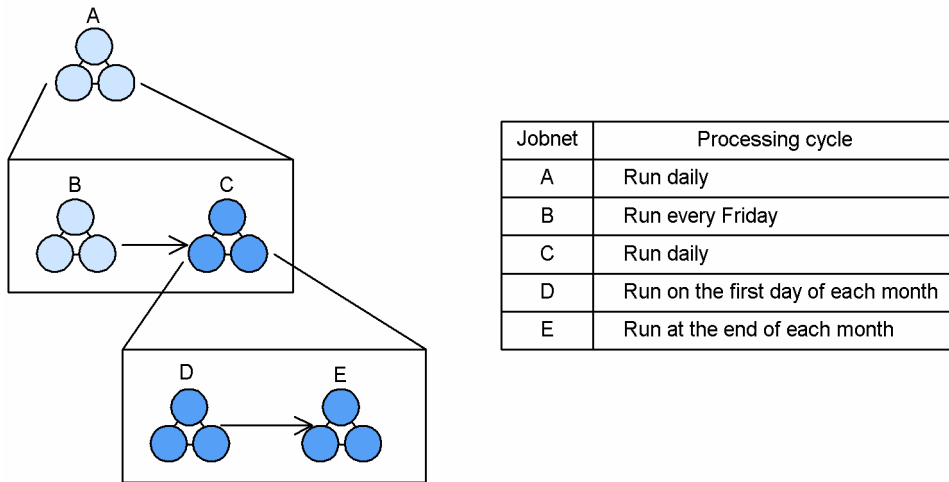
-- : Does not apply

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. Schedules are recalculated only for those jobnets added to jobnets in generations whose root jobnet is running or being rerun. The exclusive schedule of the uppermost added jobnet becomes invalid.

The following figure shows an example of adding units after suspending the root jobnet that is registered for planned execution.

Figure 4–57: 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 4–16: 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

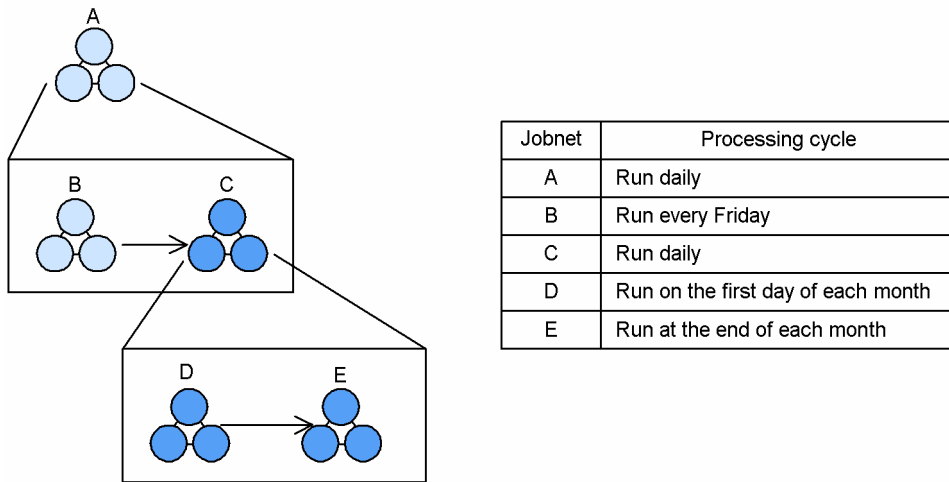
In this example, the schedule for November 30 is recalculated for jobnet C, and for the jobnets below jobnet C (D and E). Also, the exclusive schedule of jobnet B and jobnet C becomes invalid. On December 12, the exclusive schedule is valid, since the schedule is recalculated from the uppermost jobnet.

When the root jobnet is registered for fixed execution

Schedules are recalculated only for those jobnets added to jobnets whose root jobnet is waiting or running. Also, the exclusive schedule of the added uppermost jobnet becomes invalid.

The following figure shows an example of adding units after suspending a root jobnet that is registered for fixed execution.

Figure 4–58: 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 4–17: 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. In the above example, schedules are recalculated for jobnet C and the jobnets (D and E) below it, and the exclusive schedules defined for jobnet B and jobnet C become invalid. When a jobnet is registered for fixed execution with the number of future generations, schedules are recalculated from the uppermost jobnet of the jobnets in the generations created after the suspension is released, and exclusive schedules become valid.



## (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 4–18: 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.

## (7) Inheriting macro variables when adding an event job

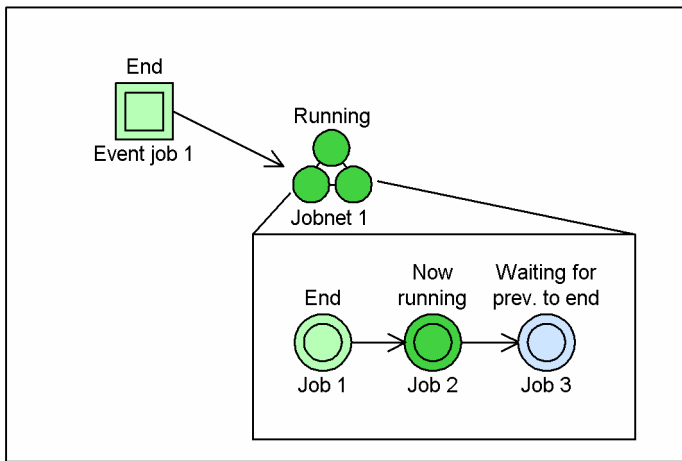
Macro variables are passed on to a succeeding unit when the status of the succeeding unit changes from Waiting to execute to Now running.

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.

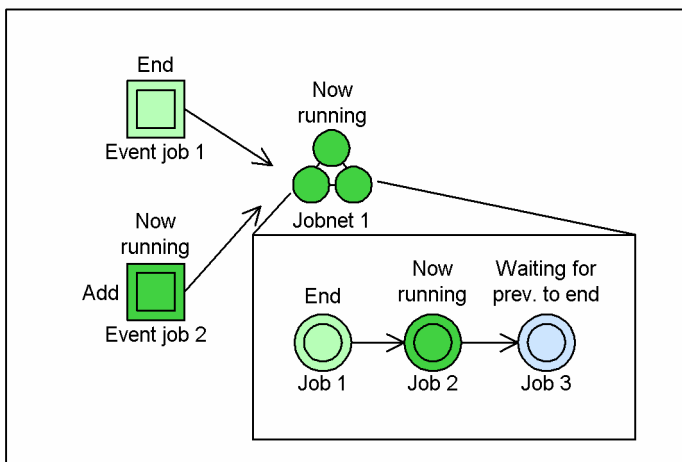
The way in which macro variables are inherited depends on the status of the unit that succeeds the added event job.

- When the succeeding unit is running  
The following figure shows macro variable inheritance according to the status of the succeeding unit.

Figure 4–59: Adding an event job that precedes a unit that is running



Suspend the root jobnet, add event job 2, and then release the suspension.

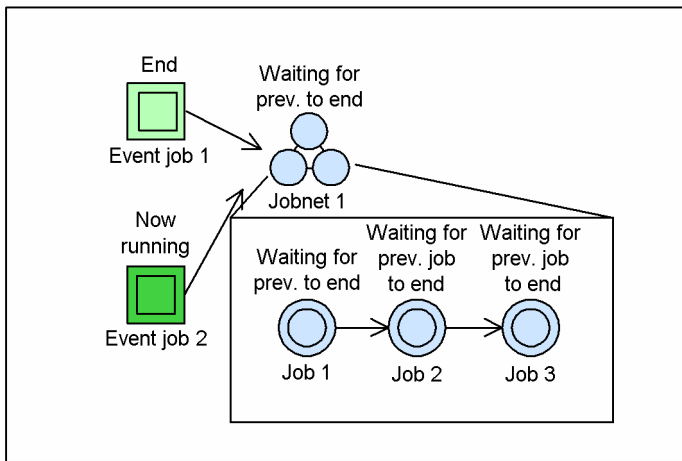


When job 2 ends, job 3 waits for event job 2 to end. The value of the macro variable that jobnet 1 inherits is created when jobnet 1 changes to Now running status. In *Figure 4-59*, the value of the macro variable that is created before suspending the root jobnet continues to be used under jobnet 1.

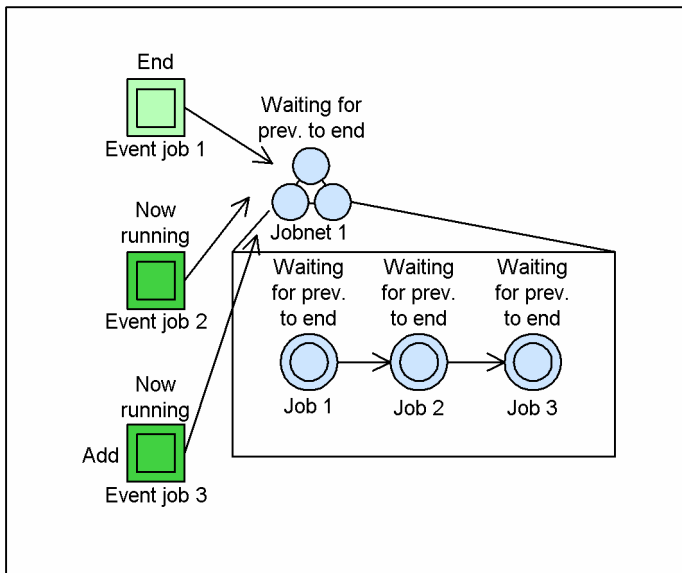
- When the succeeding unit is waiting for the previous unit to end

The following figure shows macro variable inheritance according to the status of the succeeding unit.

Figure 4–60: 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.



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 4-60*, 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.

# 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.

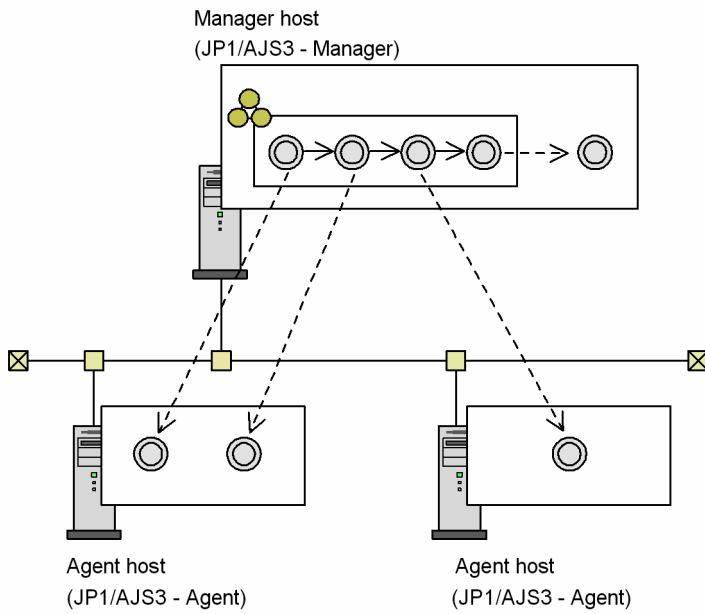
## 5.1 How jobs are executed

This section describes how jobs are executed in JP1/AJS3.

The environment for executing Unix jobs, PC jobs, flexible jobs, HTTP connection 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.

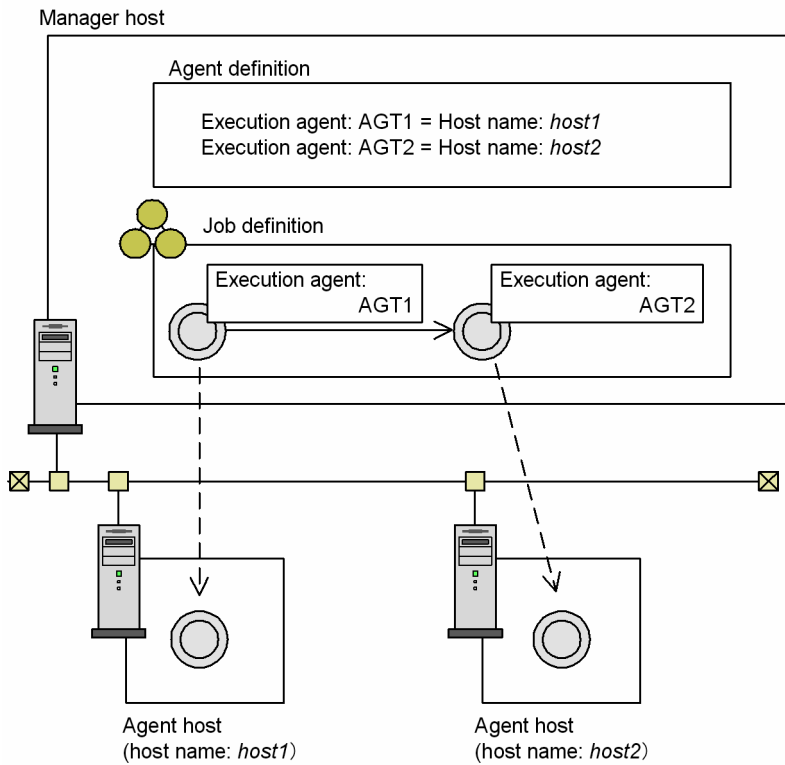
Figure 5–1: Flow of JP1/AJS3 job execution



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

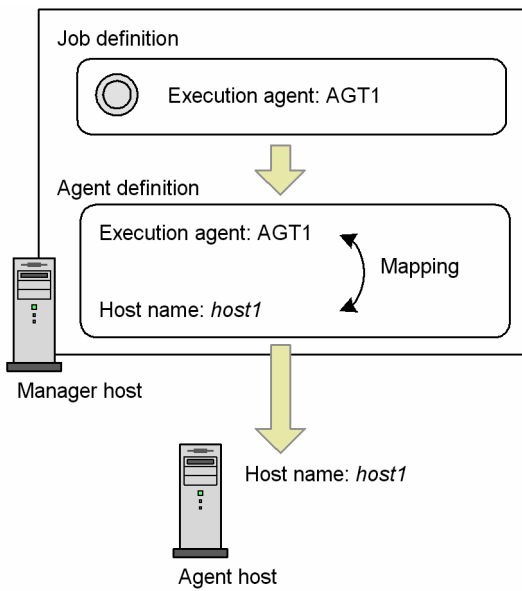
An *execution agent* is a logical name defined in JP1/AJS3 as a job execution host. JP1/AJS3 determines where each job is to be executed by mapping the execution agent name with the physical host name of an agent host that executes jobs. The job is then sent to that agent host.

#### Supplementary note

Queueless jobs do not use execution agents. To define a queueless job, specify the execution host in **Exec-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
- Flexible job<sup>#1</sup>
- HTTP connection job<sup>#2</sup>

#1

For flexible jobs, replace *execution agent* with *relay agent*.

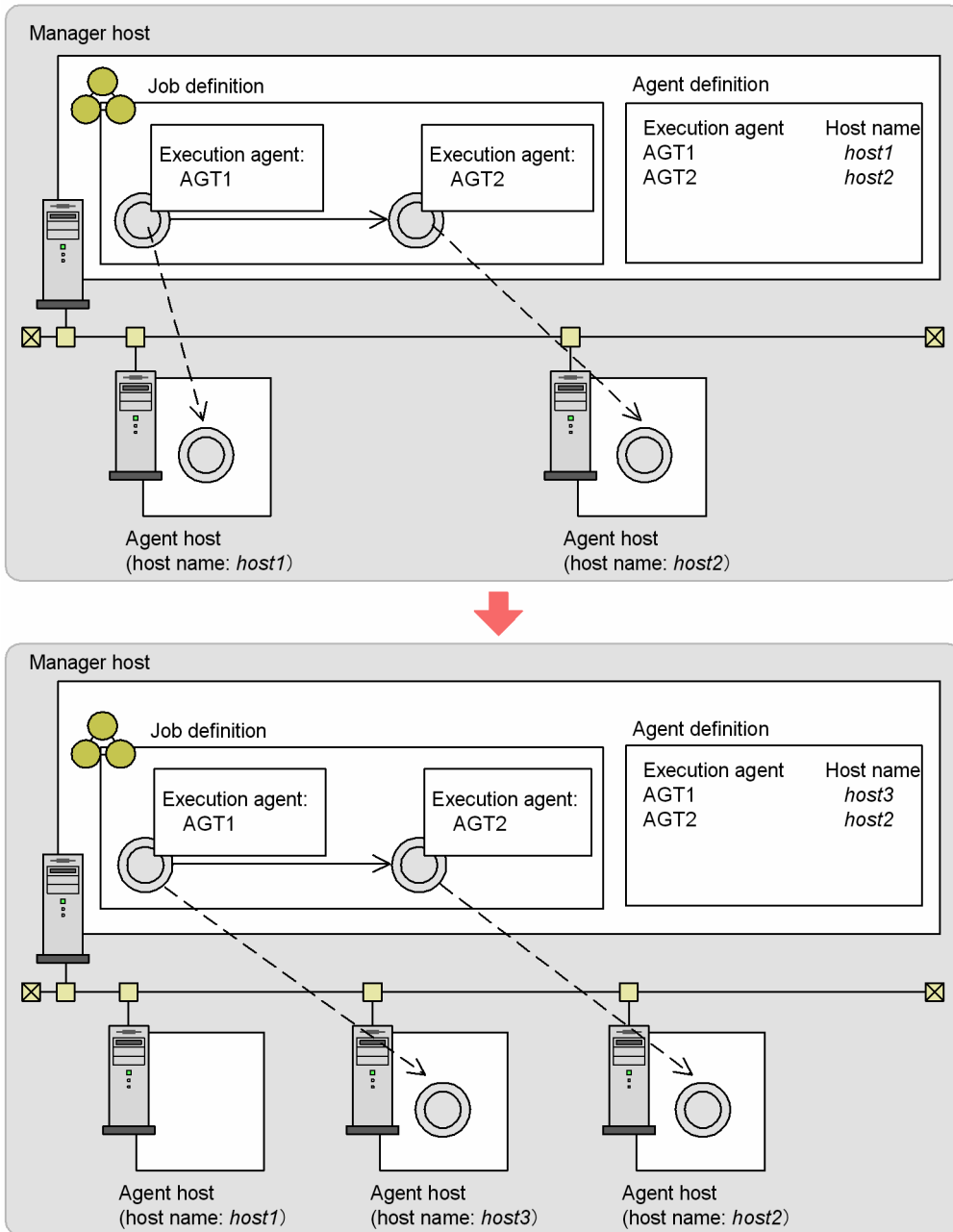
#2

The service for executing HTTP connection jobs is fixed to **Standard**.

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.

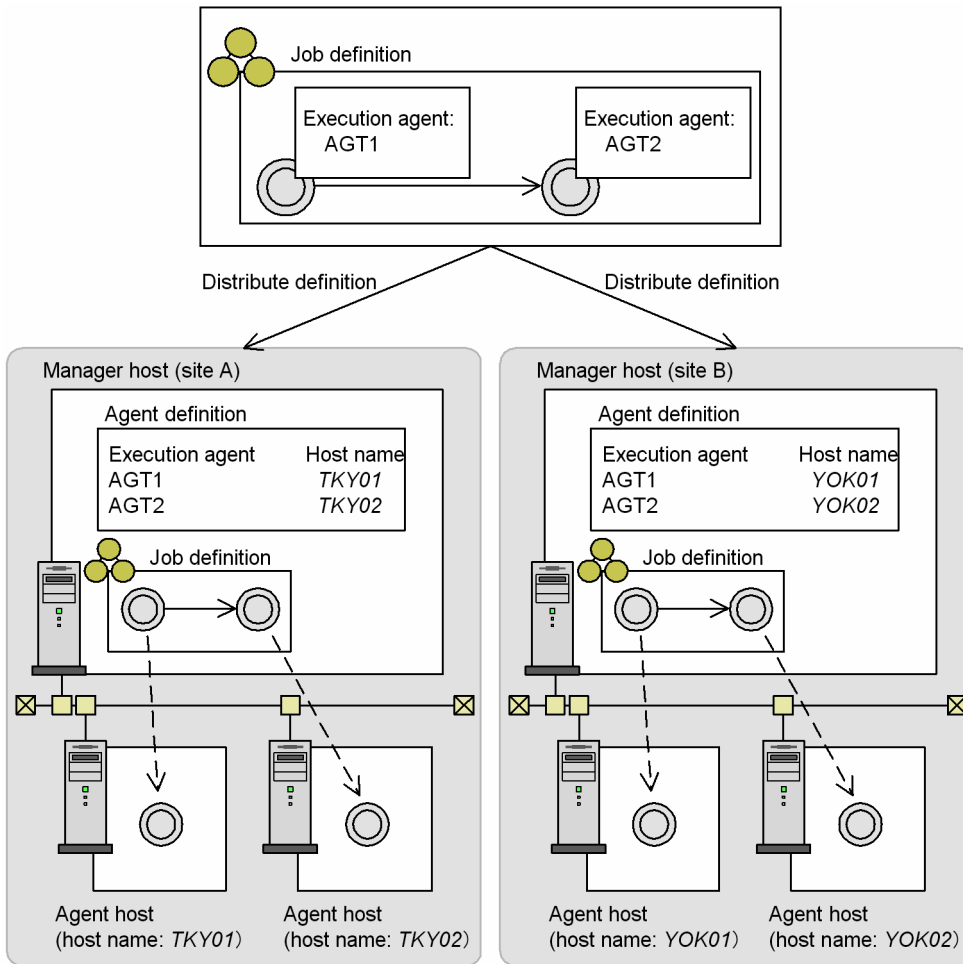
Figure 5–4: Using execution agents: Example 1



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, flexible jobs, HTTP connection jobs, action jobs, custom jobs, and passing information setting jobs, other than queueless 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 *JPI/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 *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## (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 *5.4.9 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.

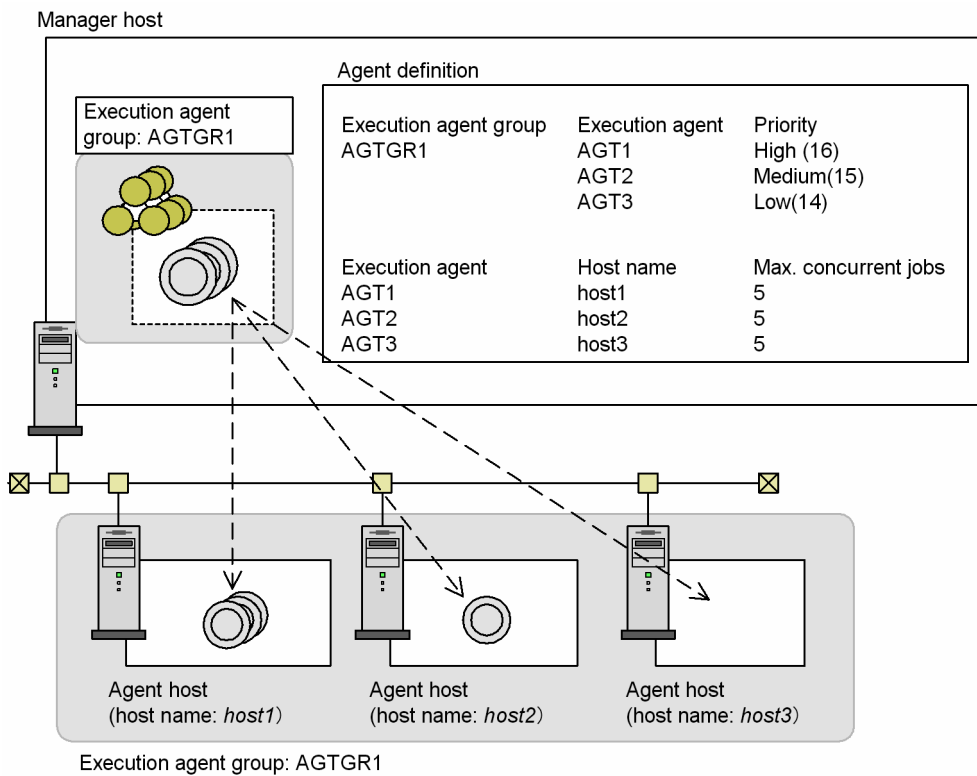
When you specify the maximum number of concurrently executable jobs, specify a value appropriate for the execution host.

### 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<sup>#1</sup>
- Nested jobnet<sup>#1</sup>
- PC job
- Unix job
- Flexible job<sup>#2</sup>
- HTTP connection job<sup>#3</sup>
- Action job
- Custom job

Note that this applies only when **Standard** is selected for **Exec. Service** in the detailed definition of a job.

#1

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.

#2

For flexible jobs, replace execution agent with relay agent.

The service for executing HTTP connection jobs is fixed to **Standard**.

## 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 *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

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 *JP1/Automatic Job Management System 3 Command Reference*. 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
<code>ajsagtadd</code>	Define the default execution agent as a connected execution agent when adding an execution agent group.	Y

Command	Purpose	Applicability
ajsagtadd	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
	Change the priority of the default execution agent when connected to an execution agent group.	Y
ajsagtdel	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.


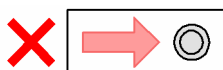
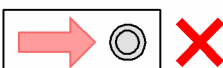

## 5.2 Restricting job transfer

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 ajsagtshow command	Meaning
Effective 	Ef	<ul style="list-style-type: none"> <li>• Jobs can be received.</li> <li>• Jobs can be sent to the agent host.</li> </ul>
Ineffective 	In	<ul style="list-style-type: none"> <li>• Jobs cannot be received (they are placed in <i>Failed to start</i> status when executed).</li> <li>• Jobs can be sent to the agent host.</li> </ul>
Hold 	Hd	<ul style="list-style-type: none"> <li>• Jobs can be received.</li> <li>• Jobs cannot be sent to the agent host (they remain on JP1/AJS3 - Manager).</li> </ul>
Blockade 	Bl	<ul style="list-style-type: none"> <li>• Jobs cannot be received (they are placed in <i>Failed to start</i> status when executed).</li> <li>• Jobs cannot be sent to the agent host (they remain on JP1/AJS3 - Manager).</li> </ul>

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> → Ended	<i>Now queuing</i> → <i>Now running</i> → Ended (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> .)	

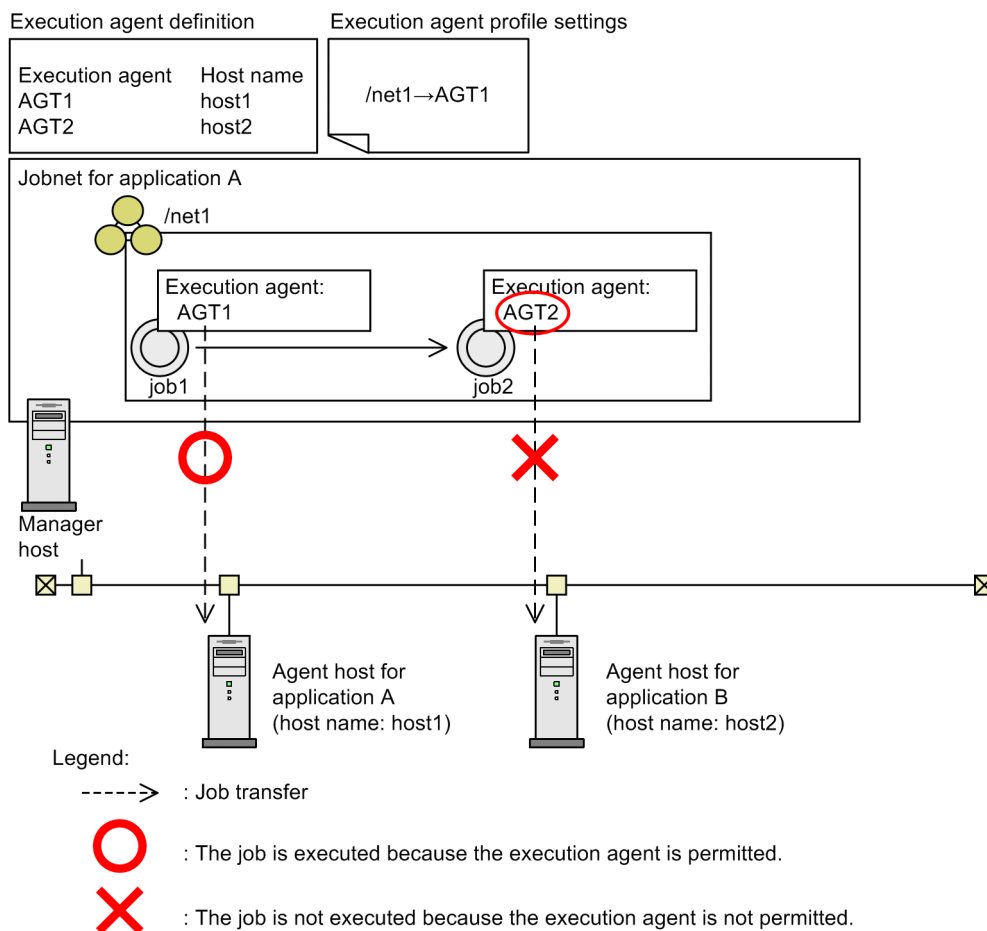
## 5.3 Execution agent restriction function

The execution agents permitted to execute jobs can be set for each unit by specifying them in operation profiles called *execution agent profiles*. The use of execution agent profiles helps to ensure that jobs are not executed by an inappropriate execution agent.

An execution agent profile is created for each scheduler service. In an execution agent profile, the execution agents or execution agent groups that are permitted to execute jobs are set for each unit. The hosts that are permitted to execute queueless jobs can also be set in an execution agent profile, in which target host names specified in queueless jobs are set in the execution agent profile.

The following figure shows an example of execution agent restriction.

Figure 5–7: Example of execution agent restriction



In this example, AGT1 is set in the execution agent profile as an execution agent permitted to execute jobs in jobnet `net1`. Job `job1` in jobnet `net1` is executed on agent AGT1 because AGT1 is specified as an execution agent. However, execution of job `job2` in jobnet `net1` is suppressed by the manager host because the execution agent specified in this job is AGT2, which is not permitted to execute jobs in jobnet `net1`.

For details about execution agent restriction, see 2.5.1(6) *Execution agent restriction* in the *JPI/Automatic Job Management System 3 System Design (Configuration) Guide*.

For details about how to specify execution agent profile settings in Windows, see 21.1.1 *Setting execution agent restrictions* in the *JPI/Automatic Job Management System 3 Configuration Guide*.

## 5.4 Job execution environments and system operation

---

### 5.4.1 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 (with the exception of queueless jobs), 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 *JP1/Automatic Job Management System 3 Configuration Guide*.

### 5.4.2 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 *JP1/Automatic Job Management System 3 Configuration Guide*.

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.

### 5.4.3 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, also see *5.4.4 Checking file access permission at job execution (UNIX only)*.

### 5.4.4 Checking file access permission at job execution (UNIX only)

If the execution host (agent) is a UNIX host, 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 also check the access control list (ACL) and secondary group settings.

The file access permission check can be performed for the following jobs in UNIX:

- Unix job
- HTTP connection job executed on a Linux host

An access permission check is performed on the following files:

- Script file<sup>#</sup>
- Environment variable file<sup>#</sup>
- Standard input file<sup>#</sup>
- Standard output file
- Standard error output file
- Transfer destination file<sup>#</sup>

#

You cannot specify this file for HTTP connection jobs.

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 *15.2.18 Enabling the file access permission check for the ACL and secondary group settings during job execution* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

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 <sup>#1</sup>	User files <sup>#2</sup>	
			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.

## 5.4.5 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 *15.2.18 Enabling the file access permission check for the ACL and secondary group settings during job execution* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

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 *2.6.2(3) Executing a standard job, action job, or custom job results in an abnormal end* in the manual *JP1/Automatic Job Management System 3 Troubleshooting*.

## 5.4.6 Login shell at job startup (Unix jobs, flexible jobs, HTTP connection jobs, and action jobs)

If the execution host (agent) is a UNIX host, when a job is started, the execution OS user's login shell (contents of the `/etc/passwd` file) on the execution agent host is executed. 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 and flexible 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 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. In Linux, use the `csh` shell instead.

Signal processing can be initialized to the default status by changing a setting as described in *15.2.17 Applying SIG\_DFL (setting for receiving the SIGCHLD signal) to a child process started from JP1/AJS3* in the *JP1/Automatic Job Management System 3 Configuration Guide*. 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.

## 5.4.7 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/jplajs2/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.

## 5.4.8 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 `jpqjjobs` command, its status changes to the status specified in the `-rs` option. For details, see `jpqjjobs` in *4. Commands Used for Special Operation* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

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), HTTP connection jobs, 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 agent host does not recover for 10 or more minutes after execution registration of a job on that agent host fails, the job is placed in *Failed to start* status. Because the manager host checks the job status for each agent host, 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), HTTP connection jobs, 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/NQSEXEC 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 `jpqj obsub` command for execution on another system. To check the job status, use the `jpqj obget` command.

#### Note on linking with JP1/NQSEXEC

Notification of job termination is not performed when a job execution request is sent from JP1/AJS3 to JP1/NQSEXEC version 05-20 or earlier. In this case, JP1/AJS3 polls the job executed by the linked JP1/NQSEXEC 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/NQSEXEC version 06-00 and later. When execution by JP1/NQSEXEC finishes, the job's status is immediately reported to JP1/AJS3.

If you are using JP1/NQSEXEC 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, HTTP connection 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 *JP1/Automatic Job Management System 3 Configuration Guide (Windows)* or *15.2.12 Changing the wait time for recovery when an agent has failed* in the *JP1/Automatic Job Management System 3 Configuration Guide (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)<sup>#1</sup>  
+ 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.<sup>#1</sup>

Note that because the manager host checks the job status for each agent host, the number of times the manager host communicates with agent hosts increases in proportion to the number of agent hosts that are managed.

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*.<sup>#2</sup> 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.<sup>#3</sup>

#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:

- Connection timeout  
Default 90 seconds
- Connection retry interval  
Default 20 seconds
- Number of connection retries  
Default 2 retries

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 connection timeout, connection retry interval and number of connection retries, see 6.2.8 *Changing the timeout period, interval of retries, and number of retries for TCP/IP connections* in the *JP1/Automatic Job Management System 3 Configuration Guide* (Windows) or 15.2.8 *Changing the timeout period, interval of*

retries, and number of retries for TCP/IP connections in the *JPI/Automatic Job Management System 3 Configuration Guide (UNIX)*.

#2

If the job was executed by the `jpqjobs` command, its status changes to the status specified in the `-rs` option. For details, see `jpqjobs` in *4. Commands Used for Special Operation* in the manual *JPI/Automatic Job Management System 3 Command Reference*.

#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*

The communication time increases in proportion to the number of agent hosts that are managed as failed. For example, if there are 10 failed agent hosts, communication might take a maximum of 3,100 seconds (about 50 minutes) at the default settings.

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 *JPI/Automatic Job Management System 3 Configuration Guide (Windows)* or *15.2.19 Placing all running jobs in an end status when a communication error occurs* in the *JPI/Automatic Job Management System 3 Configuration Guide (UNIX)*.

## 5.4.9 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, flexible jobs<sup>#1</sup>, HTTP connection 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<sup>#2</sup> 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.

#1

For flexible jobs, replace execution agent with relay agent.

#2

*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:

$$\text{Agent usage rate} = \text{number of active jobs} / \text{maximum number of concurrently executable jobs}$$

#### Cautionary note

When jobs are to be distributed, there might be multiple execution agents that have the same priority and agent usage rate. In this case, to balance the load equally, the best way is to distribute the jobs by selecting the execution agent whose idle time is longest.

For details, see 6.2.9 *Setting the method for determining the agent host to which a job will be delivered* in the *JP1/Automatic Job Management System 3 Configuration Guide* (Windows) or see 15.2.9 *Setting the method for determining the agent host to which a job will be delivered* in the *JP1/Automatic Job Management System 3 Configuration Guide* (UNIX).

## (2) Job distribution to execution hosts (agent hosts)

The manager host can distribute a job to a maximum of three agent hosts simultaneously. Note, however, that if you distribute a job to connected agent hosts when three or more of the agent hosts are unable to communicate, job distribution to normal agent hosts is delayed.

You can prevent a delay by, for example, using the `ajsagtalt` command to change the status of the job transfer restriction for the unavailable agent hosts to *Shutdown* before distributing a job. This suppresses job distribution to unavailable agent hosts.

Also, you can reduce delays in job distribution by using the job distribution delay reduction function, which suppresses job distribution to unavailable agent hosts. For details about the job distribution delay reduction function, see 2.5.5 *Considering reduction of job distribution delay* in the *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

### 5.4.10 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 `jpl1exec` 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 *JP1/Script Description and Reference (For Windows Systems)*. 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"> <li>1. JP1/AJS3 process</li> <li>2. JP1/Script process</li> <li>3. Process other than the JP1/Script process</li> </ol>	All processes up to no. 3
Pattern 2	<ol style="list-style-type: none"> <li>1. JP1/AJS3 process</li> <li>2. JP1/Script process</li> <li>3. JP1/Script process</li> <li>4. Process other than the JP1/Script process</li> </ol>	All processes up to no. 4
Pattern 3	<ol style="list-style-type: none"> <li>1. JP1/AJS3 process</li> <li>2. JP1/Script process</li> <li>3. Process other than the JP1/Script process</li> <li>4. Process other than the JP1/Script process</li> </ol>	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.

#### 5.4.11 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 *6.1.2 Stopping JP1/AJS3 services manually* in the *JP1/Automatic Job Management System 3 Administration Guide*. For details on forced termination by job execution control, see *5.4.10 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 *6.2.2 Temporarily changing the end mode of JP1/AJS3* in the *JP1/Automatic Job Management System 3 Administration Guide*.

## 5.4.12 Group ID for job execution (UNIX only)

If the JP1/AJS3 service is started automatically, the group ID of the superuser under which the service is started will be used for any jobs that are executed as the superuser. Note that this group ID is not the group ID of a user who logs in as the superuser.

## 5.4.13 Programs that JP1/AJS3 can execute (Windows only)

Jobs started by JP1/AJS3 are executed in the service space. Some programs executed by JP1/AJS3 might produce different results than when they are executed manually.

To check the behavior of programs in the service space, execute them by using the Windows Task Scheduler. If programs executed by using the Task Scheduler do not produce the same results as when they are executed manually, correct them so that they produce the same results in both cases.

When you execute a program as a task, confirm the following before starting execution:

- In the properties of the task in which the program is registered, **Run whether user is logged on or not** is selected.
- In the properties of the task in which the program is registered, the user specified for **When running the task, use the following user account** matches the user name of the OS user that executes the job in JP1/AJS3.

Note that, in the service space, you might not be able to execute interactive programs as JP1/AJS3 jobs. In such a case, you can execute the program as a JP1/AJS3 job by defining a JP1/Script script file as a JP1/AJS3 job and then using the `NetExec` command to specify the logon space. For details, see the manual *JP1/Script Description and Reference (For Windows Systems)*.

# 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 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 <sup>#1</sup> (time-wait)	Y	Y	N	N	Waiting for the execution start time to arrive. <sup>#2</sup> 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"> <li>The upper-level jobnet is in wait state.</li> <li>The scheduler service is suspended.</li> <li>The execution process for the unit has not started even though the start condition of the upper-level jobnet has been satisfied.</li> </ul>
Being held <sup>#1</sup> (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) <sup>#3</sup> (unexec)	N	Y <sup>#4, #5</sup>	Y <sup>#4, #5</sup>	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 <sup>#1</sup> (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 <sup>#1</sup> (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 <sup>#1</sup> (queuing)	N	N	Y	N	The start condition has been satisfied and the job is waiting to be executed.
Running + Abend <sup>#1</sup> (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 <sup>#1</sup>	Y	Y	N	Y	Jobs or nested jobnets running concurrently in the same jobnet ended with a warning.

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connector	
	Root	Nested			
(WA-cont)	Y	Y	N	Y	The connection-destination jobnet of the jobnet connector ended with a warning.
Ended normally <sup>#1</sup> (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 <sup>#1</sup> (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 <sup>#1</sup> (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. <sup>#3</sup> (exec-defer)	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. <sup>#4,#6</sup> A jobnet enters <i>Skipped so not exe.</i> status in the following cases: <ul style="list-style-type: none"> <li>• The jobnet's timeout period elapses while the root jobnet is in <i>Wait for start time</i> or <i>Being held</i> status.<sup>#7</sup></li> <li>• <b>Schedule skip</b> is selected in <b>Schedule option</b>, and the start time of the next generation arrives before termination of the preceding generation.</li> <li>• <b>Schedule skip</b> is selected in <b>Schedule option</b>, and a jobnet that is running in immediate execution mode is again registered for immediate execution.</li> <li>• <b>Multi-schedule</b> is selected in <b>Schedule option</b>, 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.</li> <li>• There are tasks scheduled to be executed on a day when the <code>OVERSCHEDULE</code> environment setting parameter of the scheduler service is set to <code>plan</code> and the scheduler service starts by warm start.</li> <li>• There are tasks scheduled to be executed before the scheduler service starts by warm start when the <code>OVERSCHEDULE</code> environment setting parameter of the scheduler service is set to <code>skip</code> or when registering the jobnet for</li> </ul>

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connector	
	Root	Nested			
Skipped so not exe. <sup>#3</sup> (exec-defer)	Y	Y	N	N	<p>execution in the Register for Execution dialog box, <b>Execute from next time</b> is selected for <b>Planned time passed when daemon starts</b>.</p> <ul style="list-style-type: none"> <li>When <b>Do not start monitoring</b> is set for <b>Concurrent exec. of monitoring gen</b>, it is time to start the next monitoring generation, but there is still a generation in the <i>Now monitoring</i> status.</li> <li>When <b>Does not stay with skip</b> is set for <b>Pause exec. gen. for start cond.</b> and concurrent execution is disabled, a generation is already running and the condition for starting the next generation is met.</li> </ul>
-- (invalid-seq)	Y	Y	N	N	The execution sequence is in a loop and there is job or jobnet that cannot be executed.
Interrupted <sup>#1</sup> (interrupt)	Y	Y	N	N	<p>Execution of the jobnet has been interrupted. There are no abnormally terminated<sup>#8</sup> 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:<sup>#7</sup></p> <ul style="list-style-type: none"> <li>The jobnet is interrupted from JP1/AJS3 - View or by using the Web GUI, the API, or a command.</li> <li>A queued job is killed and not executed.</li> <li>A queueless job is not executed due to a communication error or other problem.</li> <li>An abnormally terminated<sup>#7</sup> job in a jobnet is no longer in abnormally terminated status because it has been re-executed or a status change has been performed.</li> <li>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 3. <i>Commands Used for Normal Operations</i> in the manual <i>JP1/Automatic Job Management System 3 Command Reference</i>.</li> </ul> <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>
Killed <sup>#1</sup> (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"> <li>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).</li> <li>The JP1/AJS3 - Manager service stopped while a job was being executed.</li> <li>The system stopped while a job was being executed.</li> </ul>

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connector	
	Root	Nested			
Killed <sup>#1</sup> (kill)	Y	Y	Y	Y	<ul style="list-style-type: none"> <li>The active job was killed.</li> <li>Processing was cancelled because the active job exceeded its specified job execution time.</li> </ul> <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 <sup>#1</sup> (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 <sup>#1</sup> (unknown)	N	N	Y	Y	<p>The end status is unknown for either of the following reasons:</p> <ul style="list-style-type: none"> <li>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.</li> <li>The JP1/AJS3 - Manager service stopped and the system shut down while the job was being executed.</li> </ul> <p>In the case of a jobnet connector, a warm start was performed during execution.</p>
-- (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 <sup>#9</sup>	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 <sup>#9</sup>	N	N	The job or jobnet is monitoring for the events defined in the start condition.
Unmonitored + Ended (unexec-monitor)	Y	Y <sup>#9</sup>	N	N	The operation time expired before the start condition was satisfied.
Monitor terminated <sup>#10</sup> (monitor-close)	Y	Y <sup>#9</sup>	N	N	The jobnet that was monitoring for a start condition to be satisfied was interrupted or killed.
Interrupted monitoring <sup>#10</sup> (monitor-intrpt)	Y	Y <sup>#9</sup>	Y <sup>#9</sup>	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).

Status	Applicability				Explanation
	Jobnet		Job	Jobnet connector	
	Root	Nested			
Interrupted monitoring <sup>#10</sup> (monitor-intrpt)	Y	Y <sup>#9</sup>	Y <sup>#9</sup>	N	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 <sup>#10</sup> (monitor-normal)	Y	Y <sup>#9</sup>	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.
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 `ajsshow` 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 `ajsshow` command syntax, see `ajsshow` in *3. Commands Used for Normal Operations* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

#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.

In JP1/AJS3 - View, you can set different colors for the *scheduled* and *not scheduled* statuses using the **General** page in the Preferences dialog box. For details, see 12.3.42 *Preferences dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide* and 12.3.44 *Select Color dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*.

By using the Web GUI, you can set different colors for the *scheduled* and *not scheduled* statuses by using the display color settings file (`ajs3web_color.conf`). For details, see the section corresponding to the OS. For Windows, see 3.4.8 *Details on the settings in the display color settings file (ajs3web\_color.conf)* in the *JP1/Automatic Job Management System 3 Configuration Guide*. For Linux, see 13.3.8 *Details on the settings in the display color settings file (ajs3web\_color.conf)* in the *JP1/Automatic Job Management System 3 Configuration 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 timeout period elapses before the jobnet is executed.

#8

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.

#9

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.

#10

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*. In JP1/AJS3 - View, 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 *12.3.42 Preferences dialog box* in the *JP1/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.

- If you use the function for displaying a hold plan in the Web GUI, specify the necessary settings in the display color settings file (`ajs3web_color.conf`). For details, see the section corresponding to the OS. For Windows, see *3.4.8 Details on the settings in the display color settings file (ajs3web\_color.conf)* in the *JP1/Automatic Job Management System 3 Configuration Guide*. For Linux, see *13.3.8 Details on the settings in the display color settings file (ajs3web\_color.conf)* in the *JP1/Automatic Job Management System 3 Configuration Guide*.
- 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 *12.7.11 Filter Settings dialog box* in the *JP1/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.

## 6.2 Using JP1/AJS3 - View to monitor jobs and jobnets

---

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 *JP1/Automatic Job Management System 3 Command Reference*.

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 *JP1/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.

### 6.2.1 Monitoring via the JP1/AJS3 - View windows

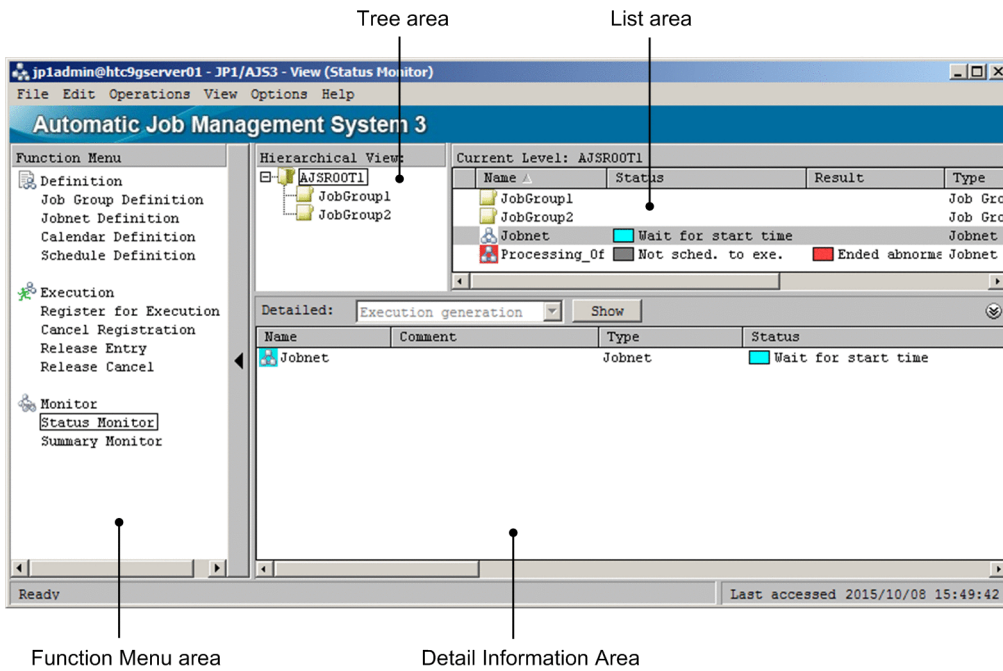
From the JP1/AJS3 - View window (Main window), you can monitor the execution status of root jobnets and job groups and check their execution results. From the JP1/AJS3 - View window (Summary Monitor window), you can monitor the progress of a root jobnet and check the execution status of the root jobnet and lower-level jobs. These two windows are described next.

#### (1) 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.

### (a) 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<sup>#</sup> whose start time or rerun time is closest to the current time
  2. A scheduled generation<sup>#</sup> that has been re-executed and is in wait state
  3. A scheduled generation<sup>#</sup> due to execute in the next run, or a generation in *Shutdown* status
- `VIEWSTATUSRANGE=today`
  1. The execution generation<sup>#</sup> whose start time or rerun time is closest to the current time
  2. A scheduled generation<sup>#</sup> that has been re-executed and is in wait state
  3. A scheduled generation<sup>#</sup> set for today and due to execute in the next run, or a generation in *Shutdown* status
  4. Of the result generations<sup>#</sup> 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"><li>• <b>Now running</b></li><li>• <b>Running + Warning</b></li><li>• <b>Running + Abend</b></li><li>• <b>Now monitoring</b></li><li>• <b>Wait for start cond.</b></li></ul>
2	Scheduled generation	<ul style="list-style-type: none"><li>• <b>Wait for start time</b></li><li>• <b>Being held</b></li></ul>
3	Result generation	<ul style="list-style-type: none"><li>• <b>Ended normally</b></li><li>• <b>Ended with warning</b></li><li>• <b>Ended abnormally</b></li><li>• <b>Interrupted</b></li><li>• <b>Killed</b></li><li>• <b>Invalid exe. seq.</b></li><li>• <b>Skipped so not exe.</b></li><li>• <b>Monitor-end normal</b></li><li>• <b>Unmonitored + Ended</b></li><li>• <b>Monitor terminated</b></li><li>• <b>Interrupted monitoring</b></li></ul>

#### 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 12.3.42 *Preferences dialog box* in the *JPI/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 12.3.42 *Preferences dialog box* in the *JPI/Automatic Job Management System 3 Operator's Guide*.

### (b) 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<sup>#1</sup>
  - Execution generation<sup>#2</sup>
  - Scheduled generation whose start time falls within today<sup>#2</sup>
  - Result generation whose end time falls within today<sup>#2</sup>
- 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.

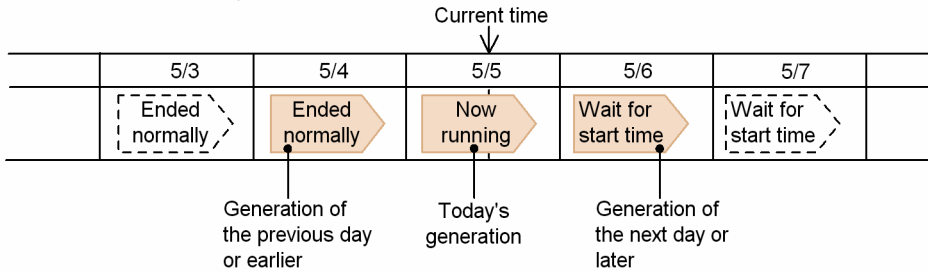
#2

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](#).

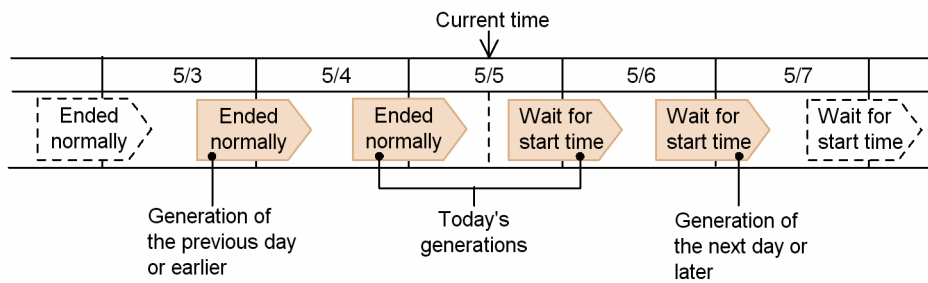
The following figure shows the generations displayed in the detailed information area.

Figure 6–2: Jobnet generations displayed in the detailed information area

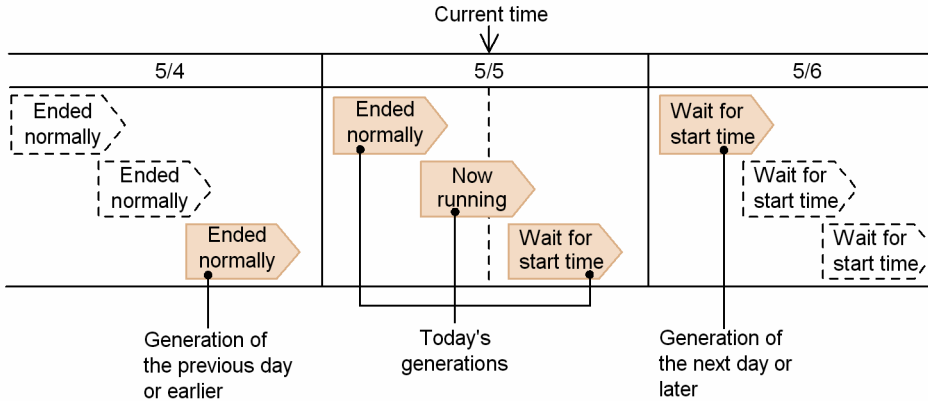
● Jobnet executed once/day



● Jobnet executed once/day that extends into the next day



● Jobnet executed 3 times/day with concurrent execution enabled



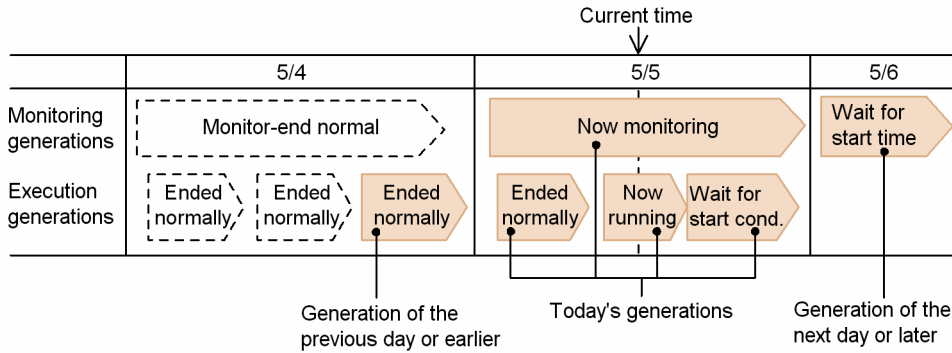
Legend:

- : Generation is displayed.
- : Generation is not displayed.

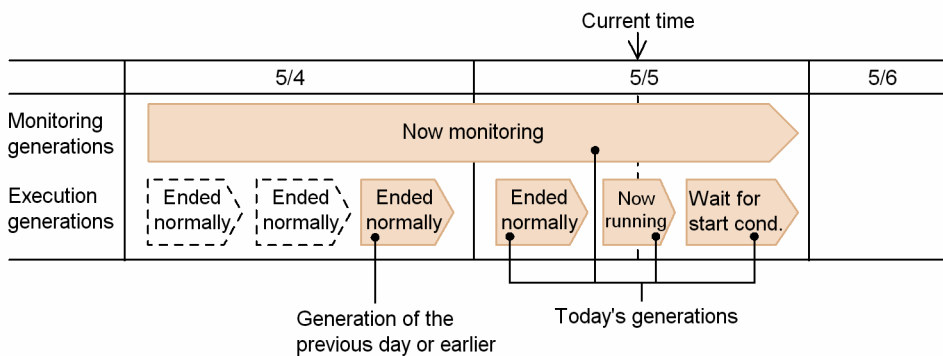
The following figure shows the generations displayed for a jobnet that has a start condition.

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 notes

- 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.

### (c) 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.

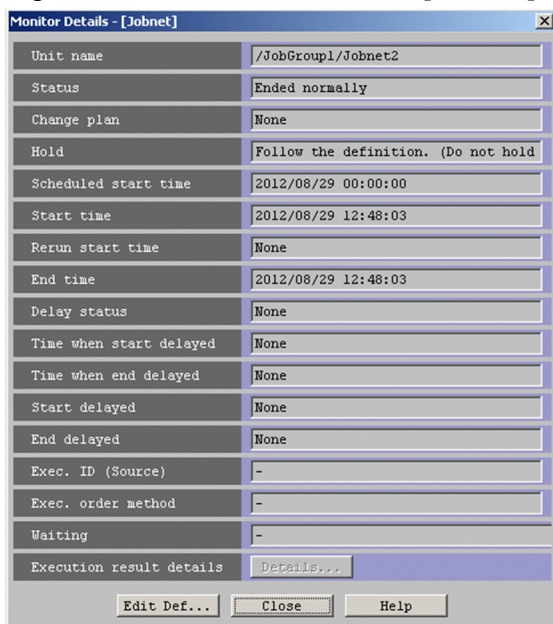
To check a root jobnet in the Monitor Details - [Jobnet] dialog box, select the root jobnet in the list area and choose **View, Detailed Information**, and then choose **Status, Result, or Next Schedule**. The generations displayed in the Monitor Details - [Jobnet] dialog box differ depending on the 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



#### (d) 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	<b>Root jobnet end delay</b>	(delayed)
2	<b>Nested jobnet or job end delay</b>	(nested jobnet delayed)
3	<b>Root jobnet start delay</b>	(delayed)
4	<b>Nested jobnet start delay</b>	(nested jobnet delayed)
5	<b>Not delayed</b>	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 *12.3.42 Preferences dialog box* in the *JP1/Automatic Job Management System 3 Operator's Guide*.



## (e) 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 *12.3.42 Preferences dialog box* in the *JP1/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	<b>Shutdown</b>	<b>Shutdown</b>
2	<b>Running + Abend</b>	<b>Running + Abend</b>
3	<b>Ended abnormally</b>	<b>Ended abnormally</b> <b>Interrupted</b> <b>Killed</b> <b>Invalid exe. seq.</b>
4	<b>Running + Warning</b>	<b>Running + Warning</b>
5	<b>Ended with warning</b>	<b>Ended with warning</b>
6	<b>Now running</b>	<b>Wait for start time</b> <b>Being held</b> <b>Wait for start cond.</b> <b>Now running</b> <b>Now monitoring</b>
7	<b>Ended normally</b>	<b>Ended normally</b> <b>Skipped so not exe.</b> <b>Unmonitored + Ended</b> <b>Monitor-end normal</b> <b>Monitor terminated</b> <b>Interrupted monitoring</b>
8	<b>Not sched. to exe.</b>	<b>Not sched. to exe.</b>
9	<b>None</b>	<b>Not registered</b>

## (2) 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.

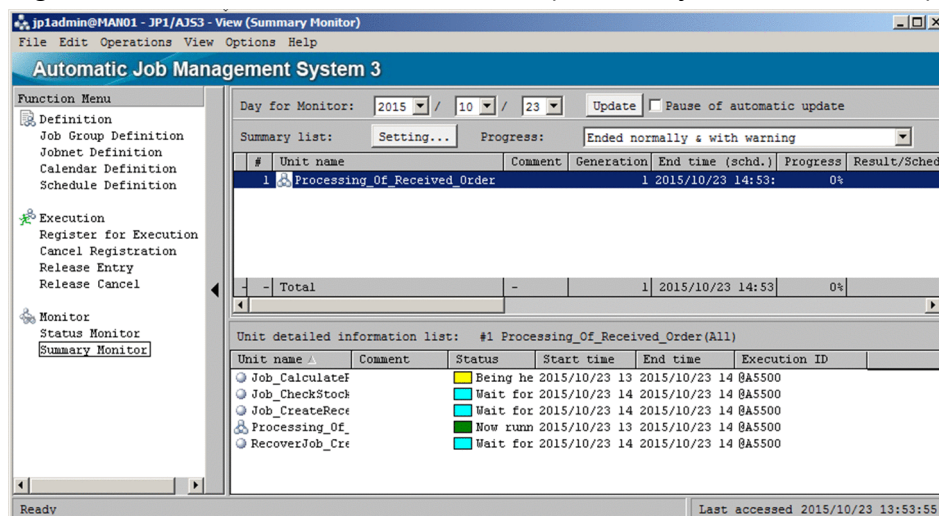
### Progress field

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.

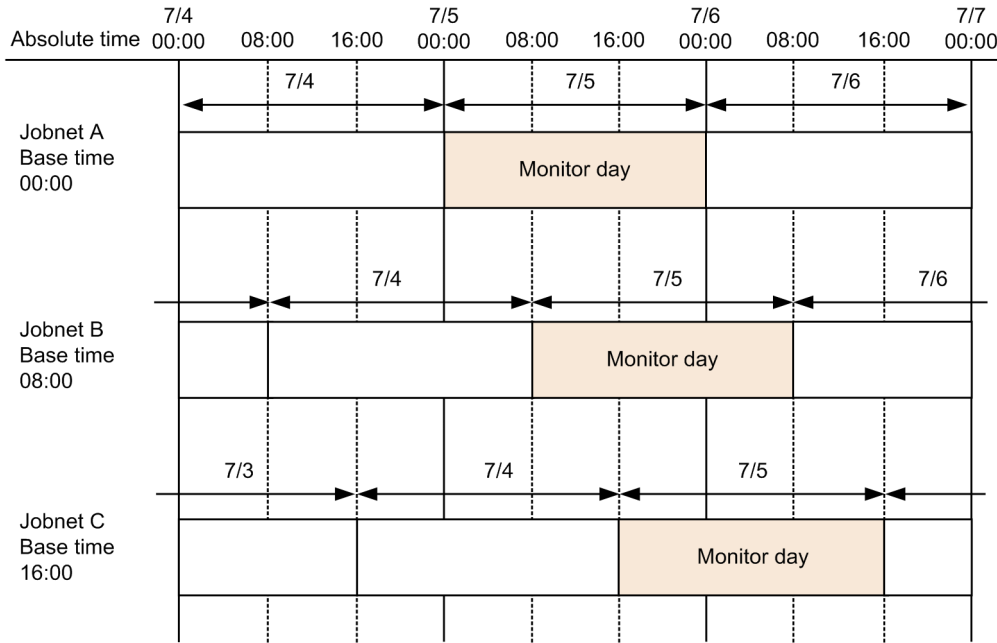
To monitor units in the Unit detailed information list, select a unit in the Summary list, and then choose **View, Unit Detailed Information List**, and *execution status*. Of the selected jobnet and its lower-level units, those units with the status you select in *execution status* are listed in the window.

### (a) 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

● Monitor day is 7/5.



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:

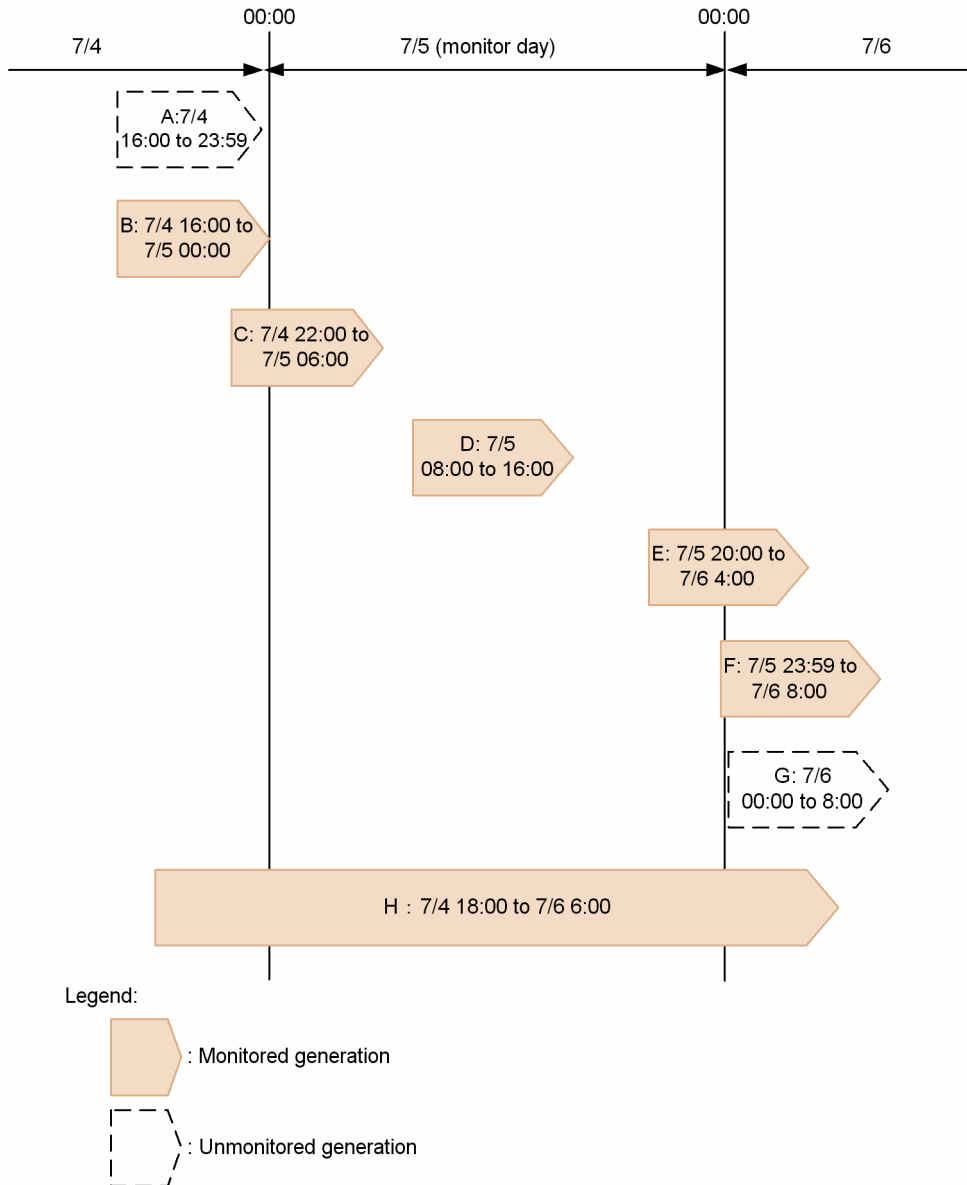
- The start time or scheduled start time falls within the specified day.
- The end time or scheduled end time falls within the specified day.
- The generation is in *Now running* status throughout the specified day.

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.

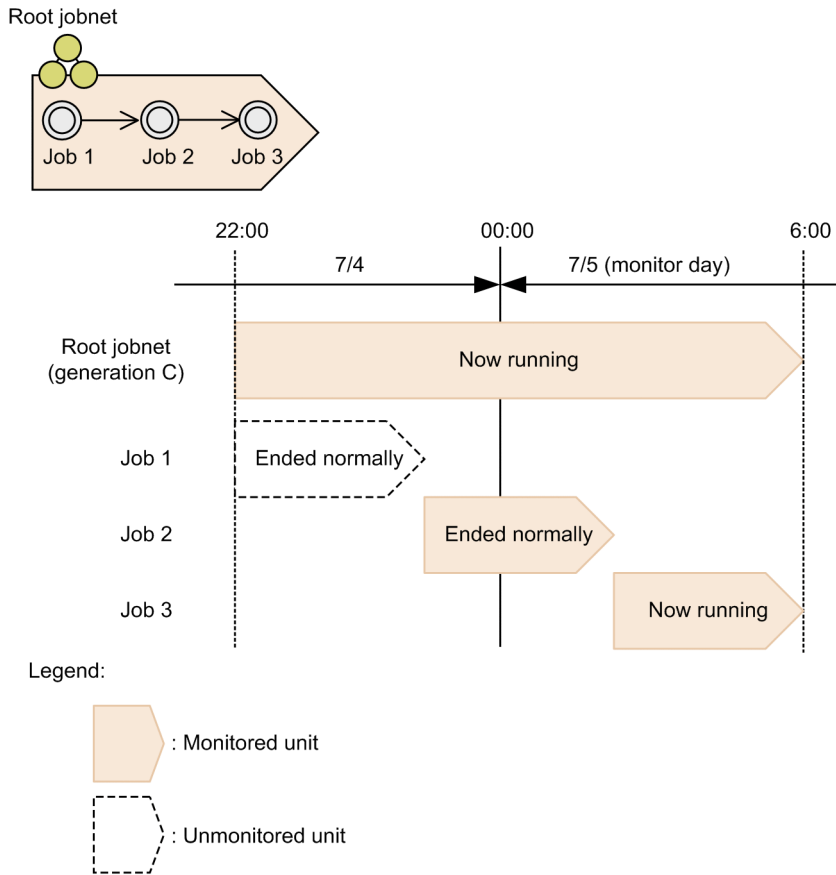
Generations E and F are monitored because their start times fall within the specified day.

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.

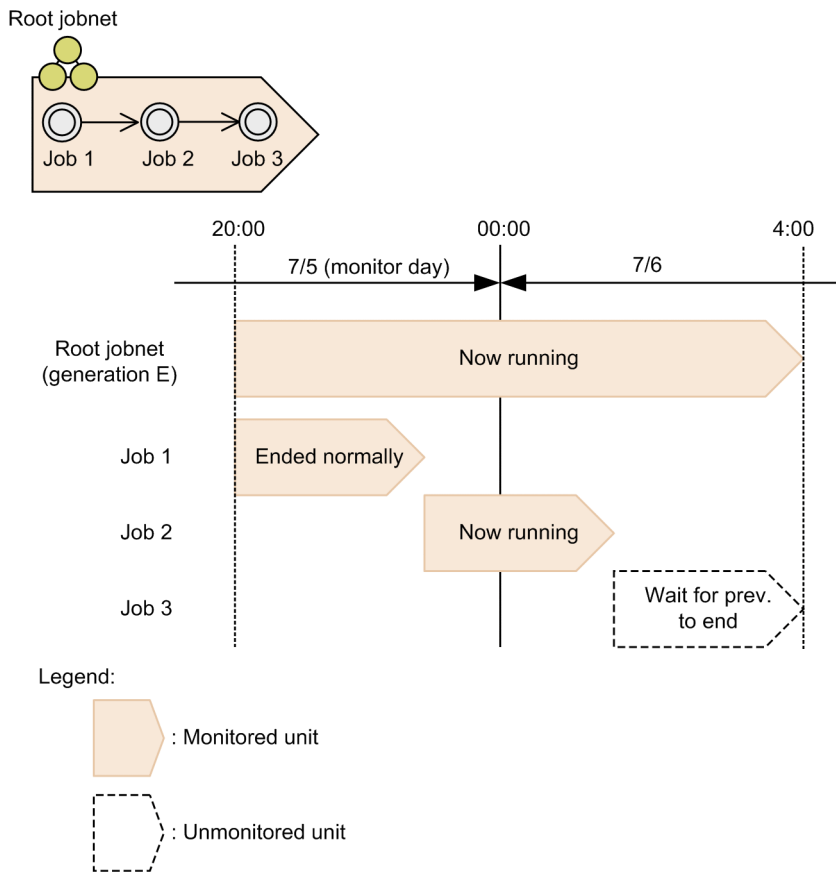
Job 2 is monitored because its end time falls within the monitor day.

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 notes

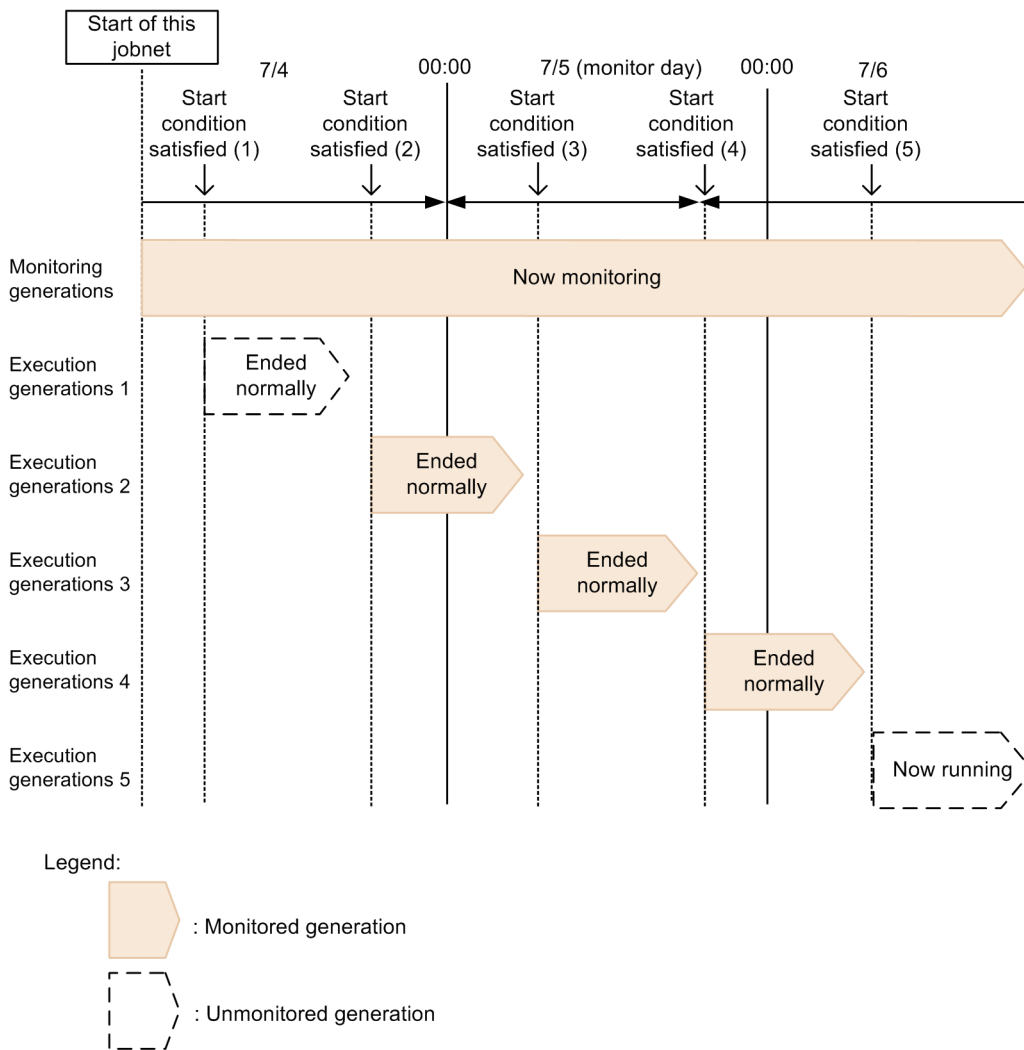
- Whether a rerun jobnet is monitored is determined from its original start time, not from its rerun start time.
- The span of the monitor day is 24 hours from the base time, regardless of whether the 24-hour schedule or 48-hour schedule is set in JP1/AJS3 - Manager.

#### 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.



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	<b>Wait for start time</b> <b>Being held</b>	The root jobnet and all lower-level units are monitored.
2	<b>Now monitoring</b> <b>Monitor-end normal</b> <b>Monitor terminated</b>	Only the root jobnet is monitored.

No.	Monitoring generation status	Monitored units
2	<b>Unmonitored + Ended Interrupted monitoring Skipped so not exe.</b>	Only the root jobnet is monitored.
3	<b>Not sched. to exe.</b>	No units are monitored.

For a dummy schedule, the root jobnet and all lower-level units are monitored.

## (b) Calculation of progress percentages

The progress percentage (%) is calculated using the following formula:

$$\text{Progress \%} = (\text{number of ended units}^{\#1}) / (\text{number of units in the execution schedules falling on the monitor day}^{\#1, \#2}) \times 100$$

#1

The total includes the root jobnet as well as the jobs in the jobnet.

#2

This number indicates the number of monitoring-target units. When monitoring a jobnet that is executed over two days, this number is the number of units whose execution start time (scheduled execution start time) or execution end time (scheduled end time) is included in the monitor day. Therefore, the number of monitoring-target units might be less than the total number of units under the root jobnet.

Cautionary note

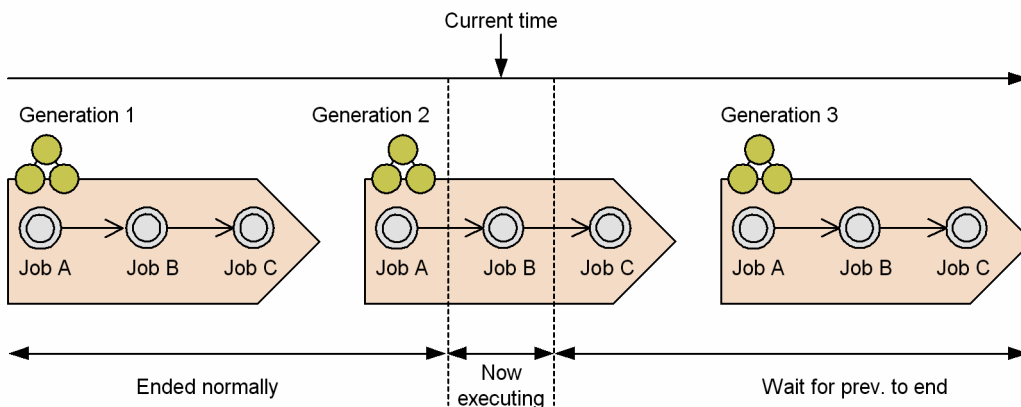
If an unlimited number of executions and an unlimited period are set as the valid range of the start condition, the monitoring generation is always in *Now monitoring* status and its progress percentage never becomes 100%.

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



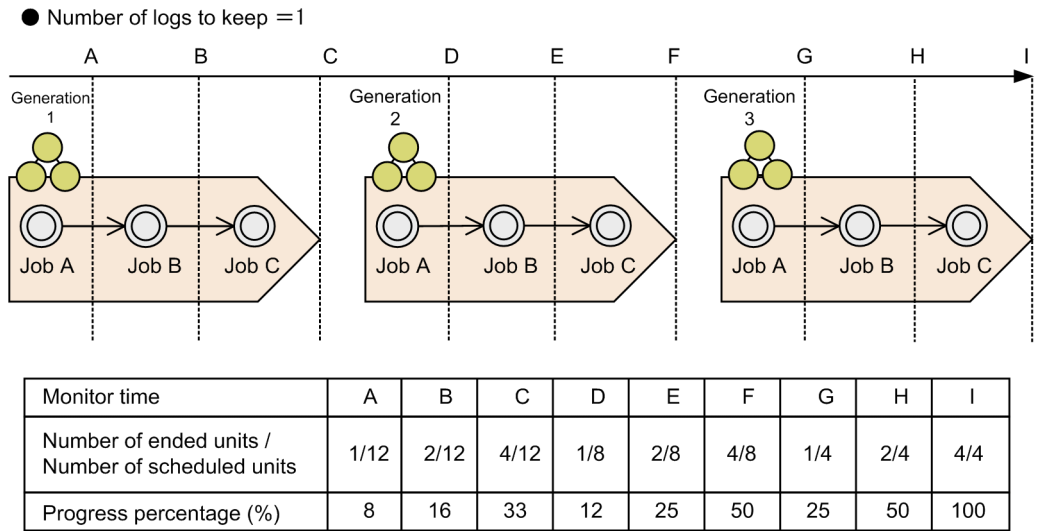
In this example, nine jobs and three jobnets (generations 1 to 3), or a total of 12 units, are scheduled to execute on the monitor day. Currently, jobs up to job A in generation 2 have ended normally, and the succeeding job B is now running. Therefore, four jobs plus one jobnet (generation 1), or a total 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**

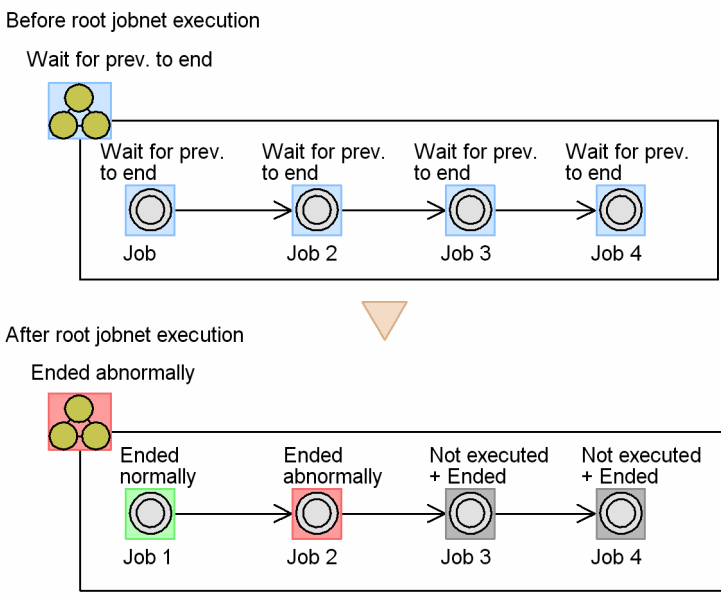


In this example, a jobnet is executed three times on the monitor day and the number of logs to keep is set to 1. In the current time frame A-C, the monitored units include the past job in generation 1 and the scheduled generations 2 and 3, making a total of 12 scheduled units to monitor. However, when 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%.

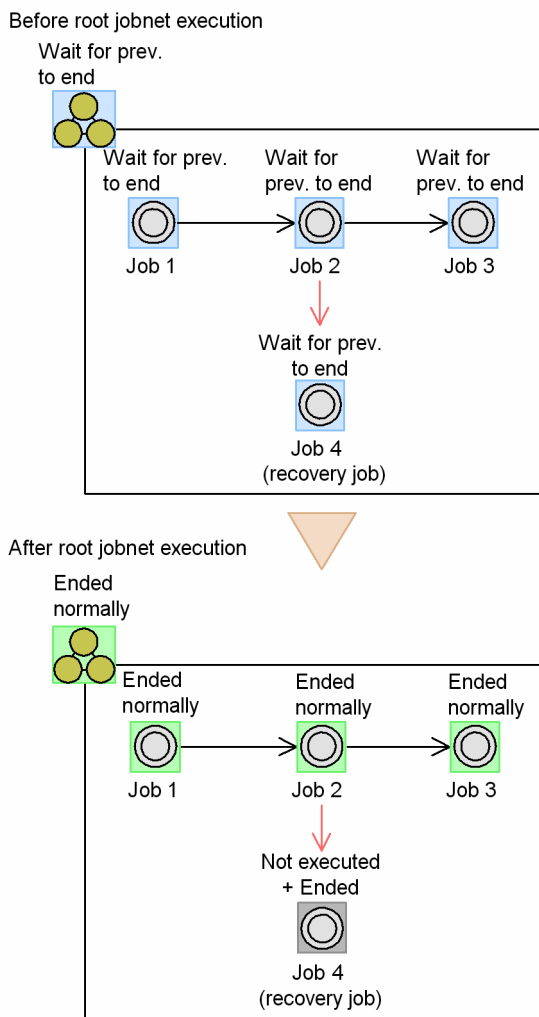
In the Summary list, 2 appears in the **Ended Abnormally** field, indicating how many jobs failed.

**Example 4: When the jobnet contains a recovery unit**

When the jobnet being monitored contains a recovery unit, the progress percentage depends on the execution result of the recovery unit.

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 units, 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.

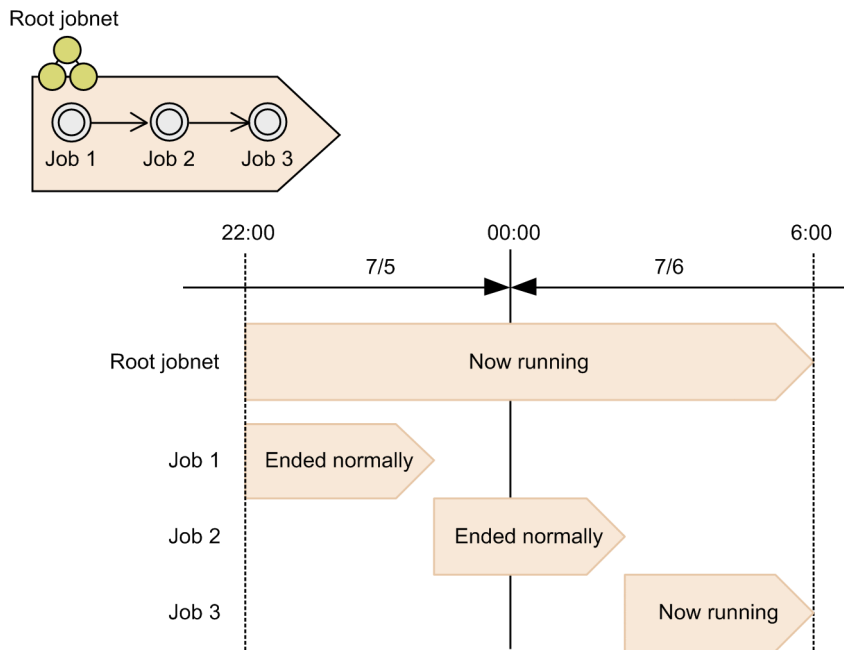
**Example 5: When monitoring a jobnet that is executed over two days**

When monitoring a jobnet that is executed over two days, the progress depends on the number of units whose execution start time (scheduled execution start time) or execution end time (scheduled end time) is included in the

monitor day. Therefore, the number of monitoring-target units might be less than the total number of units under the root jobnet.

Suppose, for example, we have a jobnet configuration like the following one.

Figure 6–15: Example of a jobnet that is executed over two days



If 7/5 is specified as the monitor day, the generations of job 3 running on 7/6 are excluded from monitoring targets. Therefore, the number of units in the *normal end* status is 2 (job 1 and job 2), and the number of units scheduled to be executed on the monitor day is 3 (root jobnet, job 1, and job 2). If these values are substituted into the formula to calculate the progress percentage, the progress percentage is calculated to be 66%.

Also, if 7/6 is specified as the monitor day, the generations of job 1 ending normally are excluded from monitoring targets. Therefore, the number of units in the *normal end* status is 1 (job 2), and the number of units scheduled to be executed on the monitor day is 3 (root jobnet, job 2, and job 3). If these values are substituted into the formula to calculate the progress percentage, the progress percentage is calculated to be 33%.

### (c) 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	<b>Ended Abnormally</b>	• Ended abnormally

No.	Class	Statuses
1	<b>Ended Abnormally</b>	<ul style="list-style-type: none"> <li>• Invalid exe. seq.</li> <li>• Interrupted</li> <li>• Killed</li> <li>• Failed to start</li> <li>• Unknown end status</li> <li>• Skipped so not exe.<sup>#1</sup></li> <li>• Monitor terminated</li> </ul>
2	<b>Ended with Warning</b>	Ended with warning
3	<b>Delay</b>	<ul style="list-style-type: none"> <li>• Start delay<sup>#2</sup></li> <li>• End delay<sup>#2</sup></li> </ul>
4	<b>Being held</b>	Being held
5	<b>Waiting to start</b>	<ul style="list-style-type: none"> <li>• Wait for start time</li> <li>• Wait for prev. to end</li> <li>• Waiting to execute</li> <li>• Now queuing</li> <li>• Wait for start cond.</li> </ul>
6	<b>Now running</b>	<ul style="list-style-type: none"> <li>• Now running</li> <li>• Running + Warning</li> <li>• Running + Abend</li> <li>• Now monitoring</li> </ul>
7	<b>Ended Normally</b>	<ul style="list-style-type: none"> <li>• Ended normally</li> <li>• Normal end + False</li> <li>• Unmonitored + Ended</li> <li>• Interrupted monitoring</li> <li>• Monitor-end normal</li> </ul>

#### 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:

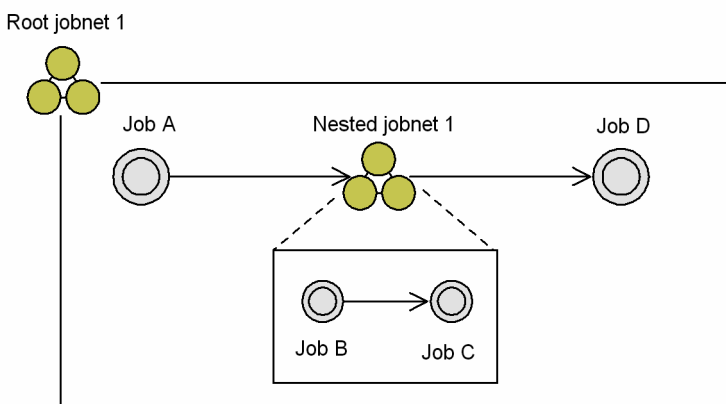
Y: Counted.

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)

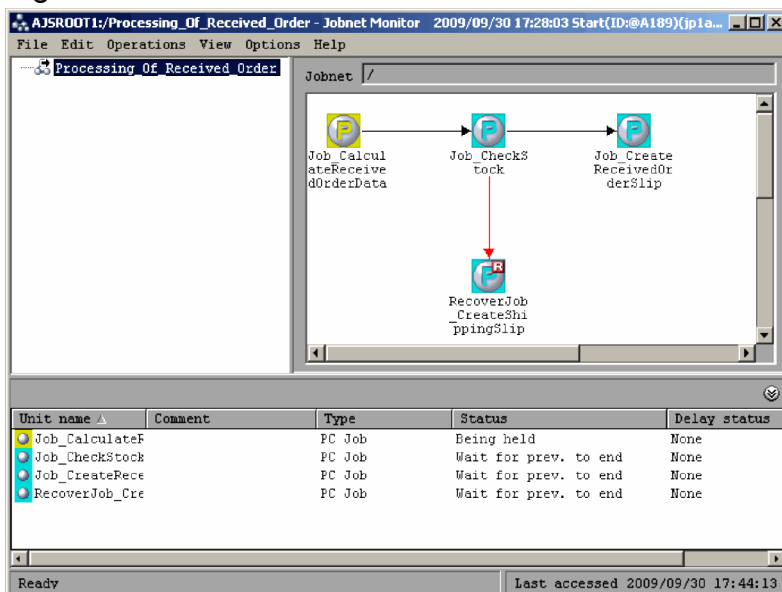
Root jobnet 1 both starts and ends later than scheduled. It is counted as a delayed unit, but the **Delay** count is incremented by 1 only, giving a total of one delayed unit.

## 6.2.2 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 6–16: 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 12.3.42 *Preferences dialog box* in the *JP1/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

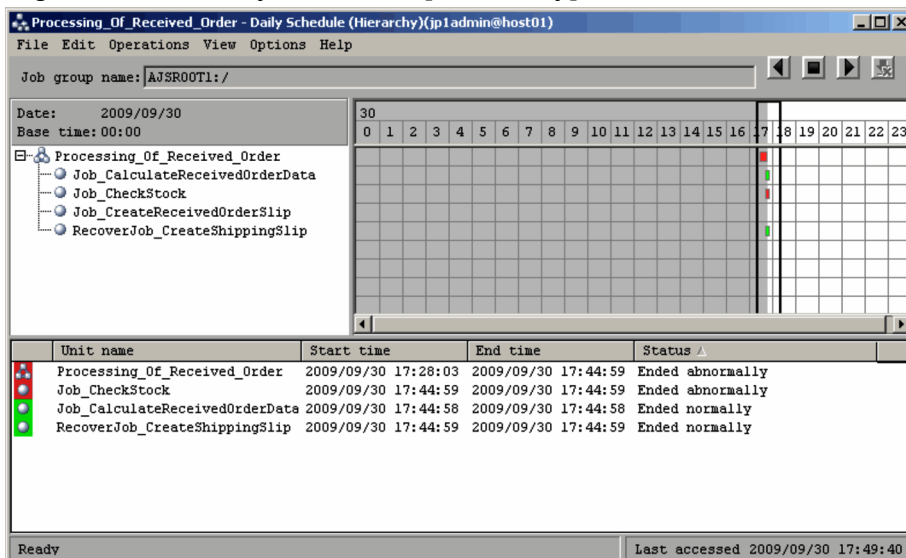
- You can display detailed information for jobnets, remote jobnets and manager jobnets.
- When you specify a manager jobnet, the execution status and results for the jobs referenced by the manager are displayed.

### 6.2.3 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 6–17: 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.

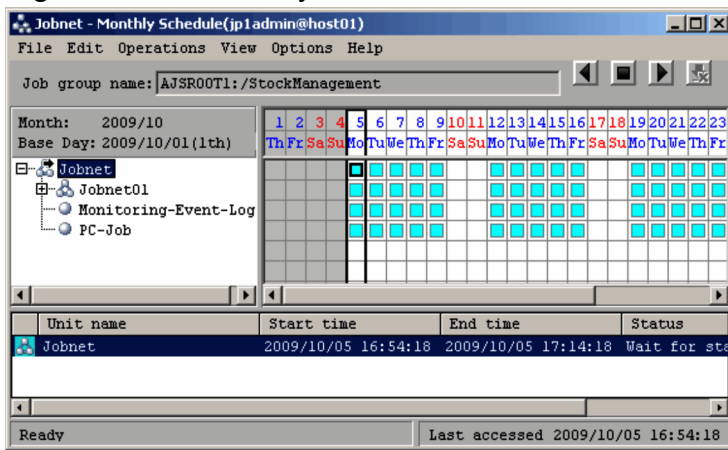
Generations that do not have an execution ID (generations that are scheduled for execution but do not have a defined schedule) are not displayed in the execution result list.

### 6.2.4 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 6–18: 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.

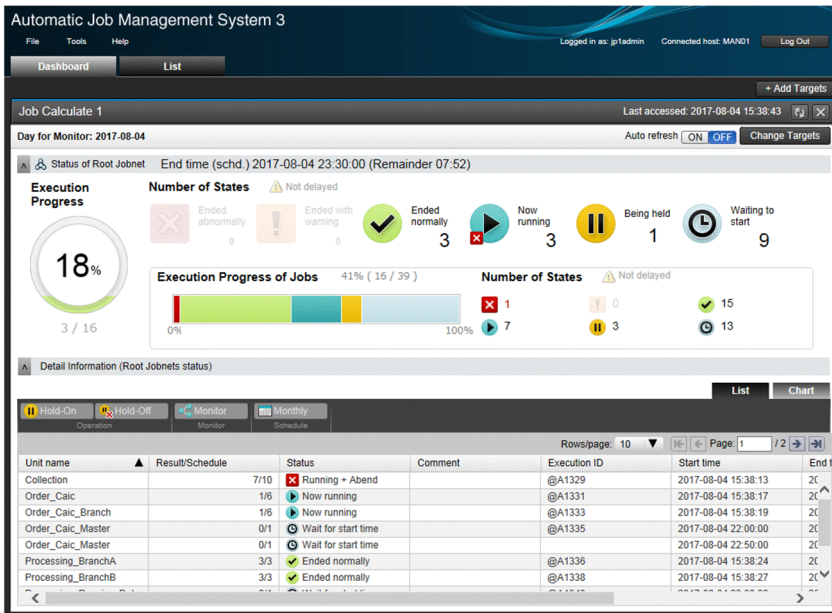
Generations that do not have an execution ID (generations that are scheduled for execution but do not have a defined schedule) are not displayed in the execution result list.



## 6.3 Monitoring jobnets by using a web browser

In JP1/AJS3, you can use a web browser to monitor the execution statuses and results of jobnets. The page displayed in the web browser for jobnet monitoring is called the *Web GUI*.

Figure 6–19: Web GUI



For details about the Web GUI, see [8. Web Interfaces Available in JP1/AJS3](#).

## 6.4 Monitoring jobnets by using user applications

---

HTTP-based APIs for operating jobnets are available to JP1/AJS3 users. JP1/AJS3 users can use these APIs to develop their own applications (user applications) for monitoring the execution statuses and results of jobnets.

For details about the user applications, see [8. \*Web Interfaces Available in JP1/AJS3\*](#).

## 6.5 Features of and differences between jobnet monitoring programs

---

In JP1/AJS3, the following three types of programs can be used to monitor jobnets:

- JP1/AJS3 - View
- Web GUI
- User application

The differences between the above program types are described below.

Differences in the program roles:

JP1/AJS3 - View is a program that allows users to perform a variety of jobnet-related operations, such as definition, execution, and monitoring.

The Web GUI is a program mainly used to execute and monitor jobnets. To define jobnets or jobs, use JP1/AJS3 - View or JP1/AJS3 - Definition Assistant.

User applications are programs that users develop by using the HTTP-based APIs available to JP1/AJS3 users. The role of a user application varies depending on the APIs or design policy.

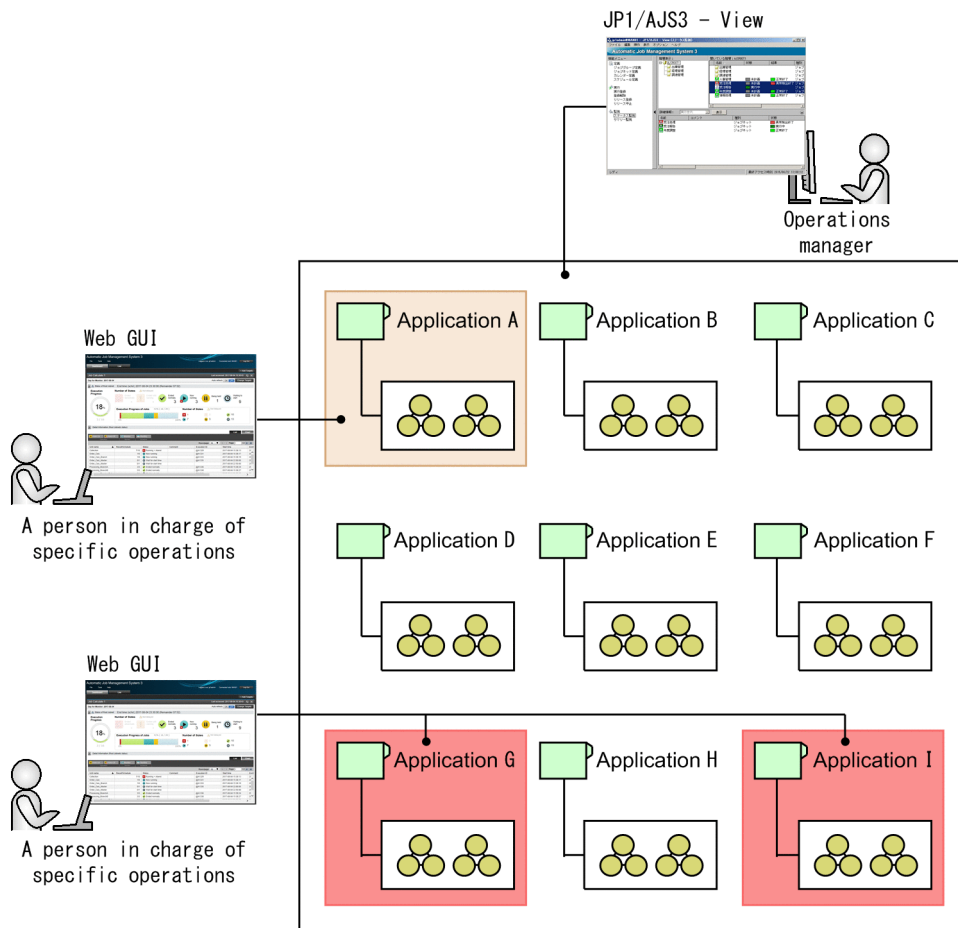
Supplementary note

JP1/AJS3 - View is a program intended to monitor whole jobs and jobnets. JP1/AJS3 - View is suitable as a GUI for the person who monitors whole applications (the operations manager).

The Web GUI is a program intended to monitor specific jobnets. The Web GUI is suitable as a GUI for the person who monitors a specific application (a person in charge of specific operations).

Use JP1/AJS3 - View and the Web GUI appropriately according to their roles.

Figure 6–20: Target usage of JP1/AJS3 - View and Web GUI



Differences in the system configuration and implementation method:

JP1/AJS3 - View must be installed as a separate task after JP1/AJS3 - Manager has been installed. JP1/AJS3 - View can be installed in Windows only.

The Web GUI uses a web browser (for example, Internet Explorer) that has already been installed in the client environment. Therefore, it is easier to prepare the monitoring terminal than to use JP1/AJS3 - View. You can also monitor jobnets on a Linux host if a supported web browser has been installed. In this case, however, you must install and set up JP1/AJS3 - Web Console, which provides server functionality, on the server to which the web browser is connected.

User applications are programs developed by users. Users can implement any functions in any programming language according to the APIs, environment, and design policy. In the same way as for the Web GUI, you must install and set up JP1/AJS3 - Web Console on the server to which the user application is connected.

Differences in the connection environment:

JP1/AJS3 - View connects to JP1/AJS3 - Manager via a LAN or WAN environment. The WAN environment used by JP1/AJS3 - View must meet requirements for communication security and line stability. For security, SSL communication (encrypted communication) can be used.

The Web GUI connects to the JP1/AJS3 system via the Internet. Users can monitor, from clients connected to the Internet, the jobnets used in various places or environments. SSL communication (HTTPS communication) can also be used, if necessary.

The connection environment of a user application varies depending on the design policy. For example, as with the Web GUI, a function for monitoring jobnets via the Internet can be implemented in a user application.

## Reference standard for the number of monitoring targets

In the case of the Web GUI, the time required to draw images on the web browser screen is proportional to the number of jobs and jobnets displayed. The following table describes a reference standard for the number of monitoring targets with the intention to use the Web GUI with an appropriate drawing time. We recommend that you use the Web GUI if the scale of the jobnets and jobs to be monitored is below this estimate, and JP1/AJS3 - View for systems whose scale exceeds it.

Monitoring target	Reference standard	Effect on drawing performance
Number of definitions of the jobnets and jobs under the root jobnet to be monitored	500 <sup>#</sup>	This number of monitoring targets affects the time required to draw the following screens and dialog box: <ul style="list-style-type: none"><li>• Monitor screen and Monitor dialog box</li><li>• Monthly Schedule screen</li></ul>
Number of saved generations of the root jobnet to be monitored and the jobnets and jobs under the root jobnet	10	This number of monitoring targets affects the time required to draw the following screen: <ul style="list-style-type: none"><li>• Monthly Schedule screen</li></ul>

#

Although monitoring on a larger scale is possible, when using the Monthly Schedule screen of the Web GUI, make sure that the number of units being monitored does not exceed the following limit for the browser. If this limit is exceeded, the screen might freeze or information might not be displayed correctly.

- Internet Explorer: 3,000 units
- Firefox ESR: 6,000 units

# 7

## 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.

## 7.1 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.

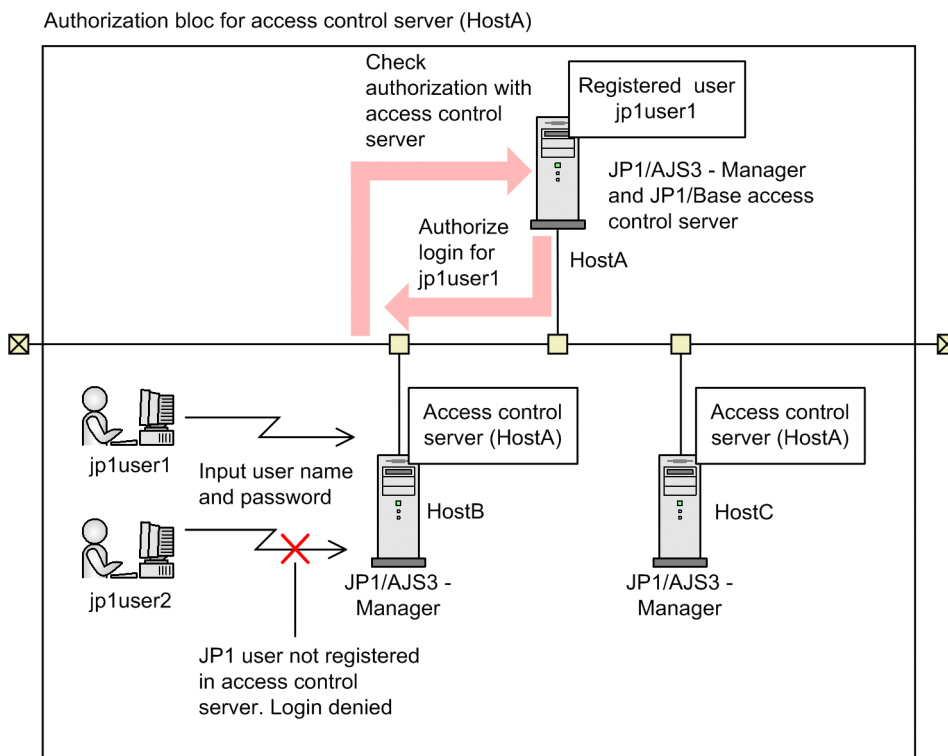
This section provides an overview of each of these functions and the setting of those functions. For details on settings and operational methods within JP1/Base, see the *JP1/Base User's Guide*.

### 7.1.1 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 7–1: 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 submit 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 *6.4.1(2) Determining JP1 permission levels* in the *JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide*.

## Supplementary note

If the OS user mapped to a JP1 user is a user with administrator privileges or superuser privileges, the OS user has the following access permissions for units:

- **JP1\_AJS\_XXX** access permissions (for defining and executing jobnets)  
All operations can be performed regardless of the JP1 permission level.
- **JP1\_JPQ\_XXX** access permissions (for executing and operating jobs, and for accessing agent management information)  
Only operations permitted by the JP1 permission level set for the mapped JP1 user can be performed.

However, if `yes` is explicitly specified for the `ADMACLIMIT` environment setting parameter, only operations permitted by the JP1 permission level set for the mapped JP1 user can be performed, regardless of the type of access. For details about this parameter, see *20.11.2(4) ADMACLIMIT* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

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 Accounting 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 Accounting, and a JP1 permission level set to `JP1_AJS_Editor` for the resource group called Sales. 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 Accounting. 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 Sales 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 Personnel 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 the user `jp1user1` is a user with administrator privileges (in Windows) or superuser privileges (in UNIX), the JP1 permission level of the JP1 resource group applies to the user.

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.

## 7.1.2 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.

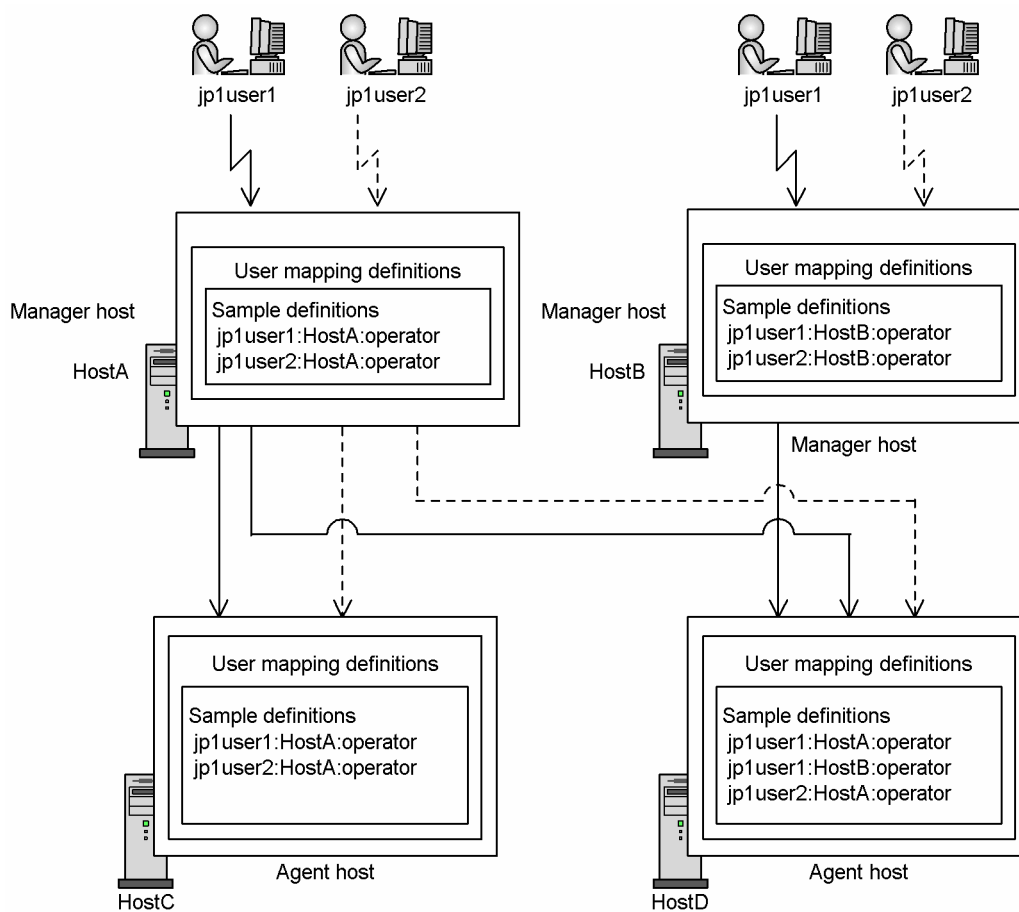
When a job or command is executed, user mapping compares the JP1 user at the source host where the execution request originated with the OS users registered in the OS of the execution target host.

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 7–2: 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.

### 7.1.3 User management using the JP1 administrators group of JP1/Base (UNIX only)

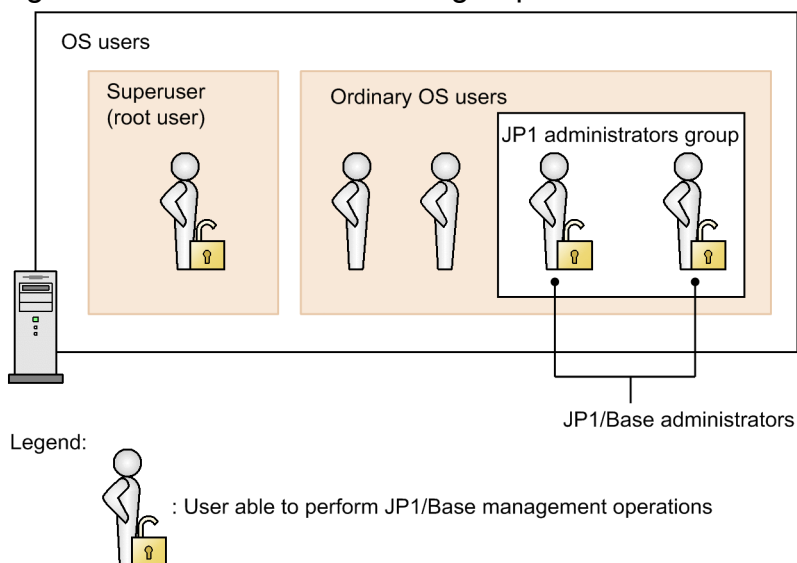
#### (1) What is the JP1 administrators group?

In JP1/Base, the following types of users perform system management operations such as starting and stopping services, changing environment setting parameters, and creating backups:

- OS user having superuser permissions (root user)
- OS user belonging to the JP1 administrators group (JP1/Base administrator)

The *JP1 administrators group* is an OS user group set in JP1/Base. OS users who belong to this group are called *JP1/Base administrators*. JP1/Base administrators are granted access permissions that are almost equivalent to superuser permissions. That is, JP1/Base administrators are ordinary OS users that can perform JP1/Base system management operations such as specifying the JP1/Base environment settings and starting and stopping services.

Figure 7–3: JP1 administrators group and JP1/Base administrators



For details about the JP1 administrators group and JP1/Base administrators, see the *JP1/Base User's Guide*.

#### (2) What is the AJS administrator?

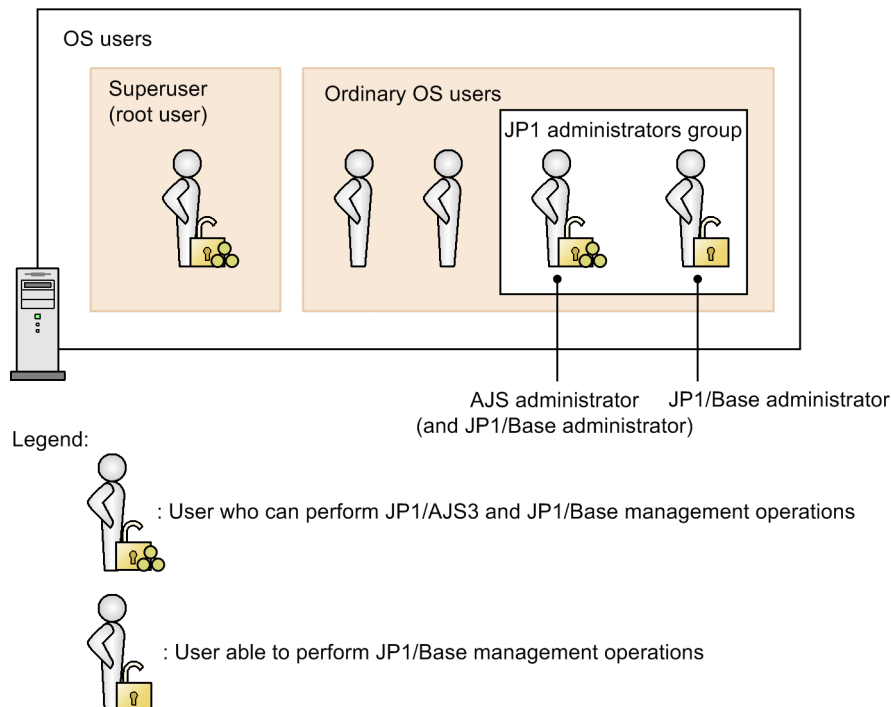
In JP1/AJS3, the following types of users perform system management operations such as starting and stopping services, changing environment setting parameters, and creating backups:

- OS user having superuser permissions (root user)

- AJS administrator (a JP1/Base administrator for whom necessary access permissions are explicitly set in JP1/AJS3)

The *AJS administrator* is a JP1/Base administrator who has been granted JP1/AJS3 access permissions that are almost equivalent to superuser permissions. That is, the AJS administrator is an ordinary OS user that can perform JP1/AJS3 system management operations such as specifying the JP1/AJS3 environment settings and starting services and stopping services.

Figure 7–4: JP1 administrators group and the AJS administrator



Note that, as the above figure shows, the AJS administrator is also a JP1/Base administrator. Therefore, the AJS administrator can also perform JP1/Base system management operations.

### (3) Operations that the AJS administrator can perform

The operations that the AJS administrator can perform are almost equivalent to the operations that users with superuser permissions can perform. These operations include the starting and stopping of JP1/AJS3 services, the creation of backups, and database maintenance. In addition, the AJS administrator can manipulate all units regardless of the JP1 permission level. Note, however, that the AJS administrator cannot perform the operations listed in the following table. These operations require superuser permissions.

Table 7–1: Operations that the AJS administrators cannot perform

No.	Operation	Description
1	Installing and uninstalling JP1/AJS3	Installing and removing JP1/AJS3 require superuser permissions. The AJS administrator cannot perform these operations.
2	Applying and deleting JP1/AJS3 patches	Applying and deleting JP1/AJS3 patches require superuser permissions. The AJS administrator cannot perform these operations.
3	Designating the AJS administrator	Whether an AJS administrator has been designated is set by a user with superuser permissions before JP1/AJS3 is installed. Ordinary OS users cannot perform this operation.
4	Allocating OS resources	OS resource allocation operations, such as registering and removing port numbers, and preparing and deleting shared directories, require

No.	Operation	Description
4	Allocating OS resources	superuser permissions. The AJS administrator cannot perform these operations.
5	Operating linked programs that require superuser permissions	The AJS administrator cannot operate any linked programs that require superuser permissions.
6	Accessing files that require superuser permissions	The AJS administrator cannot access any files that require superuser permissions.
7	Performing user mapping for users who have superuser permissions	Using the JP1/Base user mapping functionality to map a user to a user who has superuser permissions requires superuser permissions. The AJS administrator cannot perform this operation.
8	Operating cluster software	Operating a cluster system requires access permissions for the cluster software. An AJS administrator who does not have access permissions for the cluster software cannot operate the cluster software. This kind of AJS administrator cannot execute the <code>jajs_killall.cluster</code> command used to forcibly terminate the JP1/AJS3 processes running in a cluster system, and cannot change or back up the <code>jajs_killall.cluster</code> command file.
9	Backing up and restoring files that require superuser permissions	The AJS administrator cannot back up or restore files that require superuser permissions. (An example of such a file is the <code>jajs_killall.cluster</code> command file.)
10	Setting up and operating the Web Console server	The AJS3 administrator cannot perform setup and server management tasks for JP1/AJS3 - Web Console.

If operations that the AJS administrator cannot perform must be performed, an OS user that has superuser permissions must be asked to perform the operations.

For details about operation of JP1/AJS3 by the AJS administrator, see *E. Operating JP1/AJS3 By Using an AJS Administrator Account (UNIX Only)* in the *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

## 7.2 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 *JP1/Automatic Job Management System 3 Operator's Guide*.

### 7.2.1 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. For a unit owner to be able to make these changes, the unit owner needs to change the JP1 resource group, as another JP1 user, to a JP1 resource group for which the unit owner has the view 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 owner of a job is changed, the value set for the type of the job execution user changes from **User who owns** to **User who registered**. This prevents the job from being unintentionally executed by the new owner. If the type of user who can execute a job changes to **User who registered**, the job is executed by the JP1 user who registered the job for execution. Even if you change the owner in the cases below, the type of the job execution user is not changed to **User who registered**.

- The ADMACLIMIT environment setting parameter is omitted or no is set for it, and the JP1 user who performs the operation is mapped to one of the following OS users who have administrative permissions.
  - In Windows  
A JP1 user mapped to an OS user who has administrator privileges.
  - In UNIX  
A JP1 user mapped to an OS user who has superuser privileges.
- A resource group and owner are set for the job targeted by the operation, and the JP1 user who performs the operation has the JP1\_AJS\_Admin permission for that resource group.

### 7.2.2 Setting the JP1 resource group

The access permission for a unit (JP1 permission level) is assigned to all JP1 users for each JP1 resource group in JP1/Base. The JP1 permission level is enabled by setting a JP1 resource group for the unit.

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.

Supplementary note

If the OS user mapped to a JP1 user is a user with administrator permissions or superuser privileges, the OS user has the following access permissions for units:

- JP1\_AJS\_XXX access permissions (for defining and executing jobnets)  
All operations can be performed regardless of the JP1 resource group and JP1 permission level.
- JP1\_JPQ\_XXX access permissions (for executing and operating jobs, and for accessing agent management information)  
Only operations permitted by the JP1 resource group and JP1 permission level that are set can be performed.

However, if `yes` is explicitly specified for the `ADMACLIMIT` environment setting parameter, only operations permitted by the JP1 permission level set for the mapped JP1 user can be performed, regardless of the type of access. For details about this parameter, see 20.11.2(4) *ADMACLIMIT* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

## 7.2.3 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. If no owner is specified for a standard job, the JP1 user that has the same name as the OS user that started the JP1/AJS3 service on the manager host becomes the execution user.

If no owner is specified for a queueless job, the system assumes that no JP1 user is specified. In such cases, the KAVS1881-E message is output, and the job will change to the *Failed to start* status. To prevent this problem, specify an owner for queueless jobs.

In the case of an event job, because event jobs are executed with the user permission of the account from which JP1/AJS3 on the agent host 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 on the manager host.

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.

## 7.2.4 Setting the User name (for PC jobs, Unix jobs, and HTTP connection jobs)

When PC jobs, Unix jobs, or HTTP connection 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, Unix jobs, and HTTP connection 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

For PC jobs or HTTP connection jobs that are executed on Windows hosts, only the user permissions of the specified OS user can be used for the OS user information. 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.



## 7.3 Settings for restricting connection to JP1/AJS3

Restricting the hosts that are able to connect to JP1/AJS3 is called a *connection source restriction*. You can restrict connection to JP1/AJS3 - Manager or JP1/AJS3 - Agent to protect against operations or job execution from unauthorized hosts.

To use the connection source restriction function, in the connection permission configuration file, specify the IP address of a host whose connection you want to permit. There are two types of connection permission configuration file: the *manager connection permission configuration file* and the *agent connection permission configuration file*.

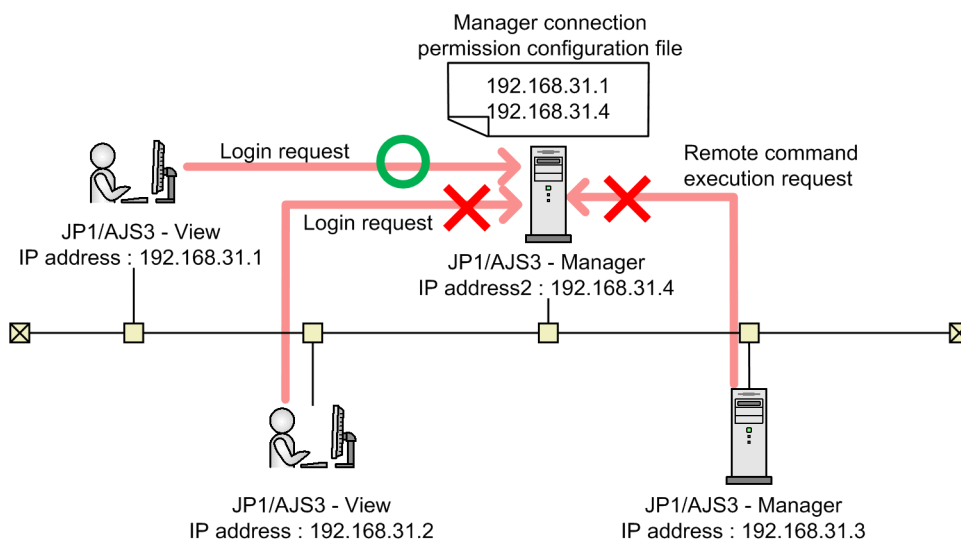
The following are examples of restricting connection to JP1/AJS3 - Manager and restricting connection to JP1/AJS3 - Agent.

### Example of restricting connection to JP1/AJS3 - Manager

To permit connection in JP1/AJS3 - Manager for a login request from JP1/AJS3 - View, or to permit a request to execute a remote jobnet from another JP1/AJS3 - Manager, in the manager connection permission configuration file, specify both the IP address of the host whose connection to JP1/AJS3 - Manager you want to permit and the IP address of the local host. These operations allow you to restrict connection from any host that does not have the specified IP address. Make sure that you specify the IP address of the local host. If you do not do this, the JP1/AJS3 service will not be able to start.

The following figure shows an example of restricting connection to JP1/AJS3 - Manager.

Figure 7–5: Example of restricting connection to JP1/AJS3 - Manager

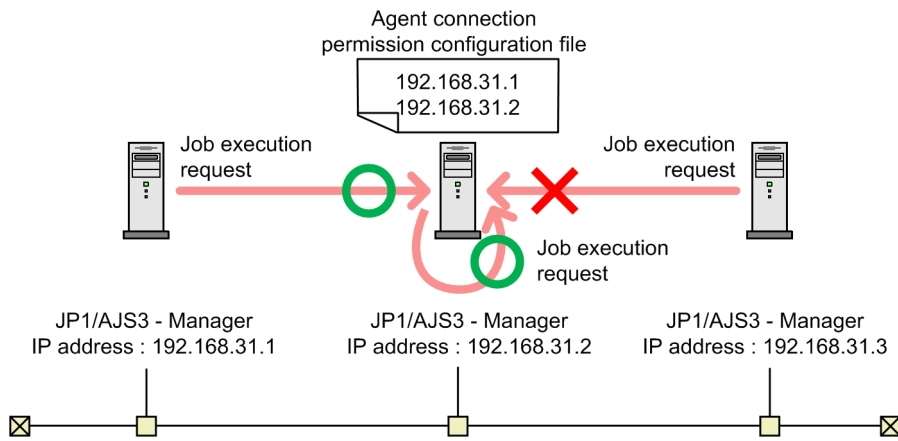


In the above figure, the IP address (192.168.31.1) of the host for which connection is permitted and the IP address (192.168.31.4) of the local host are specified in the manager connection permission configuration file. In this case, login or remote jobnet execution cannot be performed from a host whose IP address is not 192.168.31.1.

To restrict connection for a request to execute a job in JP1/AJS3 - Manager, in the agent connection permission configuration file, specify the IP address of the host whose connection you want to permit and the IP address of the local host. Make sure that you specify the IP address of the local host. If you do not do this, it will not be possible to execute a passing information setting job.

The following figure shows an example of restricting connection for a job execution request in JP1/AJS3 - Manager.

Figure 7–6: Example of restricting connection for a job execution request in JP1/AJS3 - Manager



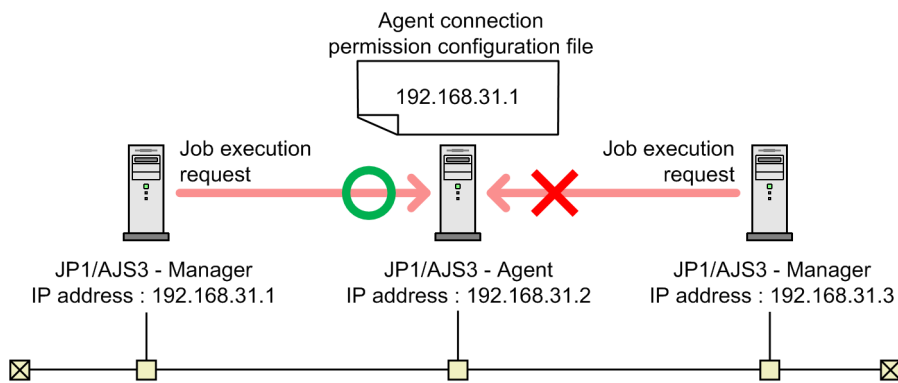
In the above figure, the IP address (192.168.31.1) of the host for which connection is permitted and the IP address (192.168.31.2) of the local host are specified in the agent connection permission configuration file. In this case, a job cannot be executed from any host that is not the local host or the host whose IP address is 192.168.31.1.

#### Example of restricting connection to JP1/AJS3 - Agent

When you permit connection in JP1/AJS3 - Agent for a job execution request from JP1/AJS3 - Manager, in the agent connection permission configuration file, specify the IP address of the host whose connection to JP1/AJS3 - Agent you want to permit. This operation allows you to restrict connection from any host that does not have the specified IP address. In contrast to JP1/AJS3 - Manager, you do not need to specify the IP address of the local host in JP1/AJS3 - Agent.

The following figure shows an example of restricting connection to JP1/AJS3 - Agent.

Figure 7–7: Example of restricting connection to JP1/AJS3 - Agent



In the above figure, the IP address (192.168.31.1) of the host for which connection is permitted is specified in the agent connection permission configuration file. In this case, a job cannot be executed from any host whose IP address is not 192.168.31.1.

For details about the connection source restriction function, see *2.3.9 Restricting hosts that can access JPI/AJS3* in the *JPI/Automatic Job Management System 3 System Design (Configuration) Guide*.

For details about how to set the connection source restriction function, see *21.2 Restricting connections to JPI/AJS3* in the *JPI/Automatic Job Management System 3 Configuration Guide*.

# 8

## Web Interfaces Available in JP1/AJS3

JP1/AJS3 users can use the Web GUI to execute and monitor units via the Internet. In addition, JP1/AJS3 users can use the HTTP-based API provided by JP1/AJS3 to develop web-based user applications that operate JP1/AJS3.

This chapter describes the web interfaces that are available in JP1/AJS3.

## 8.1 Overview of web interfaces available in JP1/AJS3

JP1/AJS3 users can monitor the execution status and results of jobnets from various locations or environments by using the Internet.

The following two programs can be used on the monitoring terminal (client) to access the Internet:

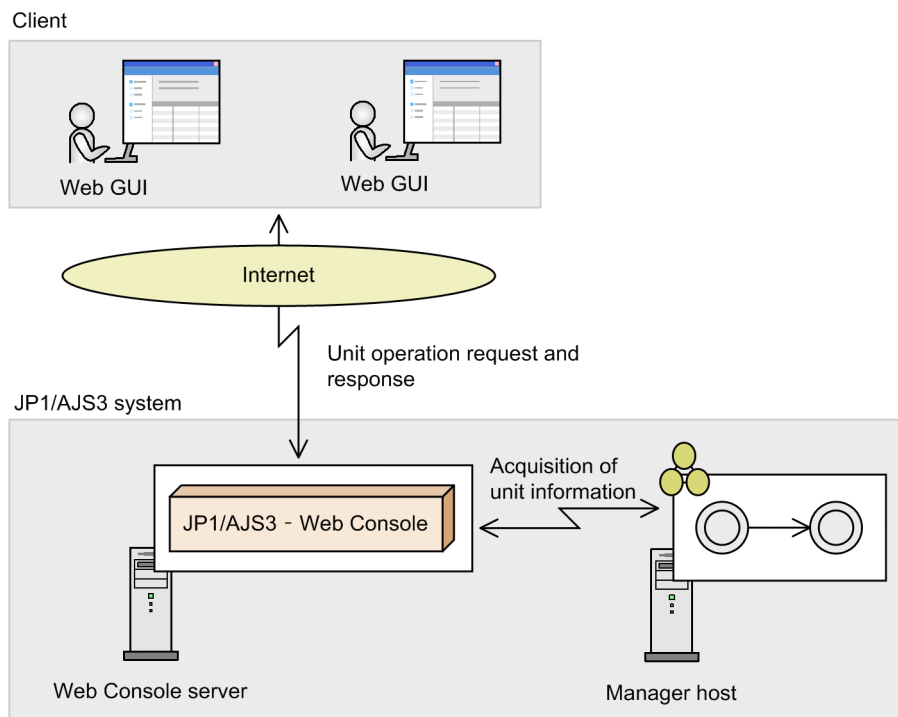
- Web browser (Web GUI)  
Web browsers installed in the client environment can be used to monitor the execution statuses and results of units. Jobnets can also be executed or suspended from the web browser.
- User application  
JP1/AJS3 provides HTTP-based APIs for users. JP1/AJS3 users can use these APIs to develop their own applications. For example, an API that obtains the execution information of units and an API that registers units for execution are available.

To monitor jobnets via the Internet by using the above programs, JP1/AJS3 - Web Console (a component of JP1/AJS3 - Manager) is required. The host on which JP1/AJS3 - Web Console is installed is called the *Web Console server*.

The Web Console server serves as a web server and web application server to control HTTP communication and web applications. In addition, when a client requests unit information, the Web Console server obtains the unit information from JP1/AJS3 - Manager and then sends the information to the client as a response.

The following figure shows an overview of communication via the Web Console server.

Figure 8–1: Overview of communication via the Web Console server



For details about the system configuration, see 2.2.5 *Examples of system configurations that include a Web Console server* in the *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

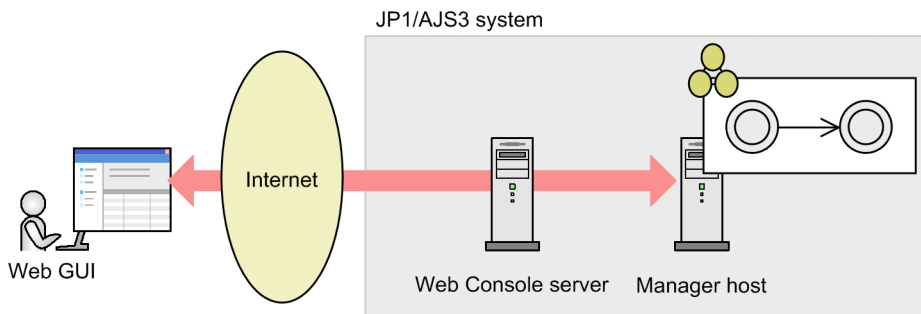
## 8.2 Operating JP1/AJS3 by using the Web GUI

This section describes how to operate JP1/AJS3 by using the Web GUI.

### 8.2.1 What you can do with the Web GUI

The Web GUI allows users to monitor the progress and execution statuses of jobnets via the Internet. The Web GUI can be used in many ways, such as for monitoring (and operating) jobnets from outside the company, and monitoring JP1/AJS3 applications built into a cloud environment.

Figure 8–2: Overview of the Web GUI



The following table describes the operations that can be performed by using the Web GUI.

Table 8–1: Operations that can be performed by using the Web GUI

No.	Operation		Description
1	Executing units	Immediate execution registration	You can register jobnets for immediate execution.
2		Planned execution registration	You can register jobnets for planned execution.
3		Fixed execution registration	You can register jobnets for fixed execution.
4		Cancellation of registration for execution	You can unregister jobnets that have been registered for execution.
5		Addition of an execution schedule <sup>#1</sup>	You can add an execution schedule to a unit.
6	Monitoring units <sup>#2,#3,#4</sup>	Monitoring the execution status <sup>#5</sup>	You can check the execution status, execution result, and execution schedule of each unit. You can also check details about the execution results and the results of passing macro variables.
7		Monitoring the progress of processing <sup>#6</sup>	You can check the total number of units that are scheduled for execution and the total number of terminated units in the root jobnet. You can also check the processing progress calculated from these values. In addition, you can check the number of delayed units and abnormally terminated units.
8		Checking the execution schedule	You can check the execution schedule of each unit.
9		Checking the monthly execution schedule <sup>#7</sup>	You can check the monthly execution schedule of a unit.

No.	Operation	Description
10	Operating units	Manipulating the hold attribute <sup>#1</sup>
11		Interrupt
12		Kill
13		Rerun
14		Change job status
15		Change plan (change time) <sup>#1</sup>
16		Change plan (execute immediately) <sup>#1</sup>
17		Change plan (execution prohibited) <sup>#1</sup>
18		Change plan (release change) <sup>#1</sup>
19		Change delay monitor <sup>#1</sup>

#1

You cannot view or re-manipulate temporarily changed information by using the Web GUI. If you need to view or re-manipulate temporarily changed information, use JP1/AJS3 - View.

#2

Manager job groups and manager jobnets are not displayed.

#3

The jobs created directly under a job group are not displayed.

#4

You cannot monitor the units under a remote jobnet or view the schedule and execution result information on those units.

#5

For custom jobs, a common custom job icon is displayed regardless of the type of custom job.

#6

You cannot specify a planning group as a monitoring target. If you want to monitor the jobnets under a planning group, specify those jobnets as monitoring targets.

#7

You cannot display a monthly schedule by selecting a job group or planning group.

## 8.2.2 Web GUI screens for monitoring

This subsection describes the screens provided by the Web GUI.

### (1) Dashboard screen

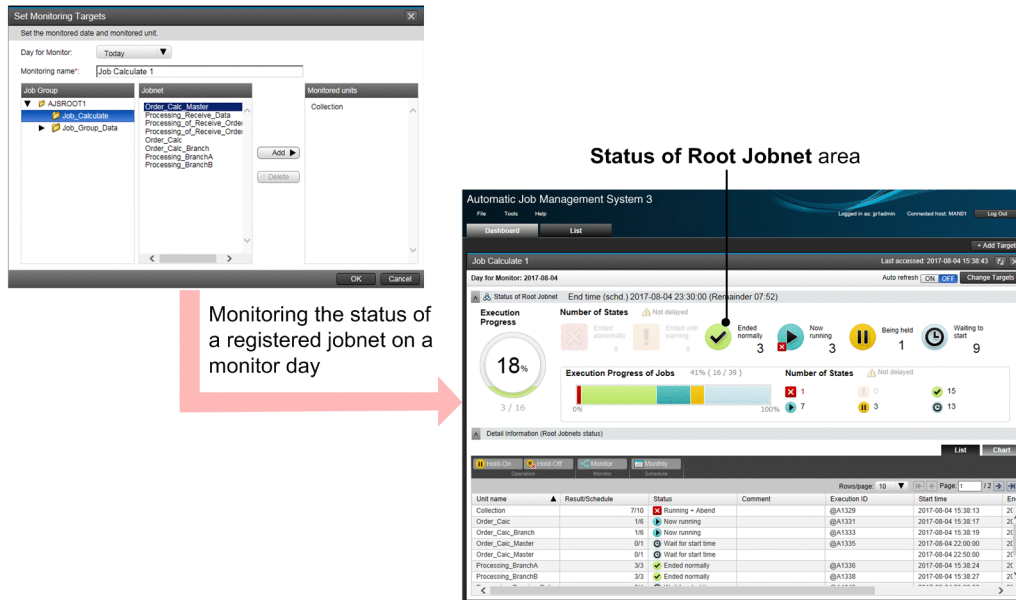
After registering monitoring-target root jobnets, you can monitor the execution status and progress of each registered root jobnet and of all units (that is, registered root jobnets and the jobnets and jobs under the root jobnets). In addition

to the execution status of a root jobnet, the number of units in the root jobnet is displayed for each execution status. Progress is displayed as a percentage of terminated units.

You can also display (in chart form) the execution status and schedule of a registered root jobnet on a specific monitor day.

The Dashboard allows you to specify a certain day (for example, the current day or a previous day). This is useful when you want to perform monitoring on a per-day basis.

Figure 8–3: Dashboard screen



Monitoring the status of a registered jobnet on a monitor day

The **Status of Root Jobnet** area on the Dashboard screen displays the number of units by status for a specified root jobnet or all units. The following table describes classification of unit statuses.

Table 8–2: Classification of unit statuses

No.	Classified status	Status
1	<b>Ended abnormally</b>	<ul style="list-style-type: none"> <li>Ended abnormally</li> <li>Invalid exe. seq.</li> <li>Interrupted</li> <li>Killed</li> <li>Failed to start</li> <li>Unknown end status</li> <li>Skipped so not exe.#</li> <li>Monitor terminated</li> </ul>
2	<b>Ended with warning</b>	<b>Ended with warning</b>
3	<b>Ended normally</b>	<ul style="list-style-type: none"> <li>Ended normally</li> <li>Normal end + False</li> <li>Unmonitored + Ended</li> <li>Interrupted monitoring</li> <li>Monitor-end normal</li> </ul>
4	<b>Now running</b>	<ul style="list-style-type: none"> <li>Now running</li> <li>Running + Warning</li> <li>Running + Abend</li> <li>Now monitoring</li> </ul>

No.	Classified status	Status
5	<b>Being held</b>	<b>Being held</b>
6	<b>Wait for start time</b>	<ul style="list-style-type: none"> <li>• <b>Wait for start time</b></li> <li>• <b>Wait for prev. to end</b></li> <li>• <b>Waiting to execute</b></li> <li>• <b>Now queuing</b></li> <li>• <b>Wait for start cond.</b></li> </ul>

## Note

Units in the following statuses are not monitored:

- Not registered
- Not sched. to exe.
- Bypassed
- Shutdown
- Not executed + Ended

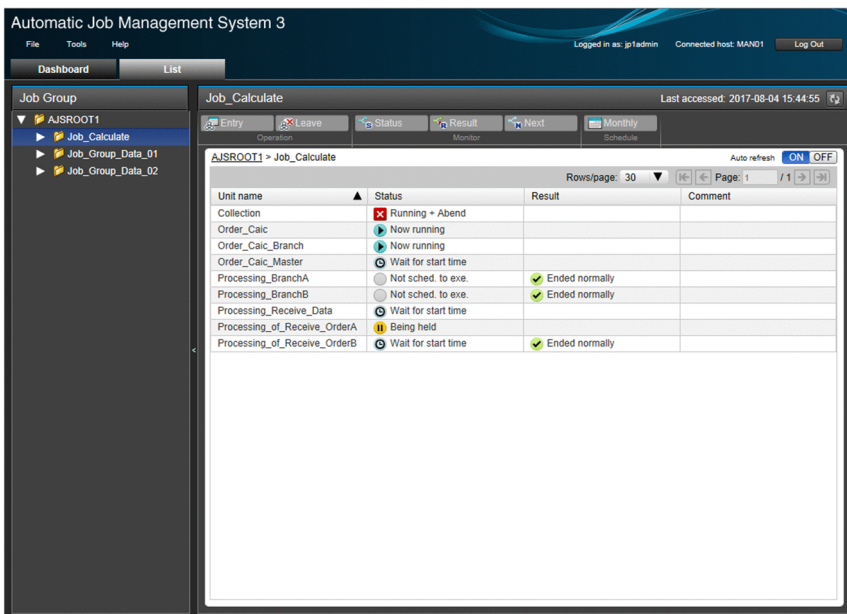
#

A unit enters *Not sched. to exe.* status if the execution schedule of its nested jobnets differs from that of its higher-level jobnet or if the unit is prohibited from being executed (and has thereby no execution schedule). A unit in *Not sched. to exe.* status transitions to *Skipped so not exe.* status if its root jobnet is skipped without execution. In this case, the unit is not monitored although it is in *Skipped so not exe.* status.

## (2) List screen

The List screen displays the job groups and jobnets defined in JP1/AJS3 in a tree or list. You can register root jobnets for execution on the List screen, and then monitor the registered root jobnets on the Dashboard screen.

Figure 8–4: List screen

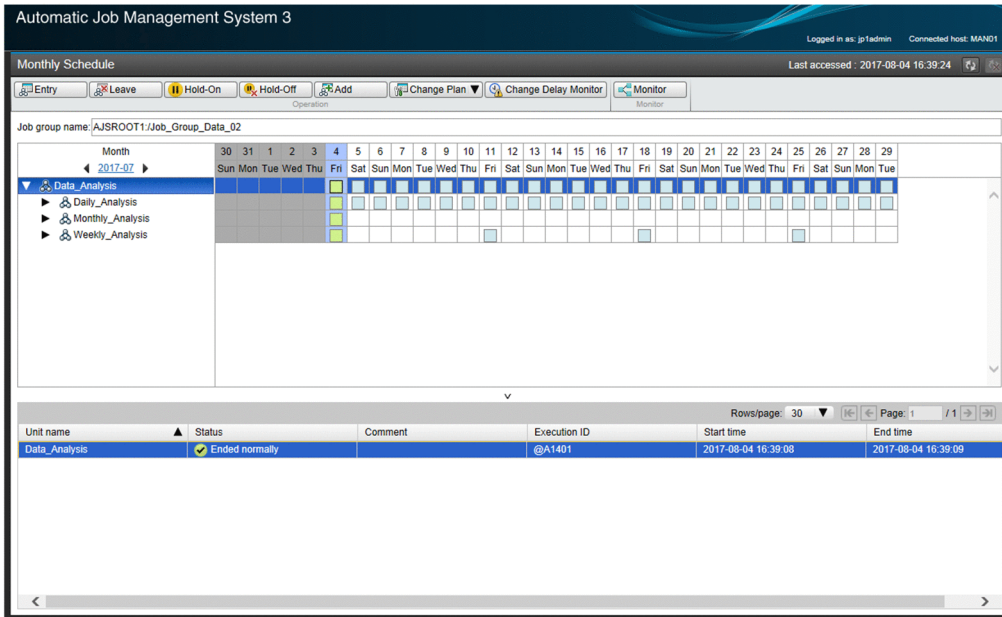




### (3) Monthly Schedule screen

The Monthly Schedule screen allows you to view monthly execution schedules of jobnets and jobs. On this screen, you can also add and change execution schedules.

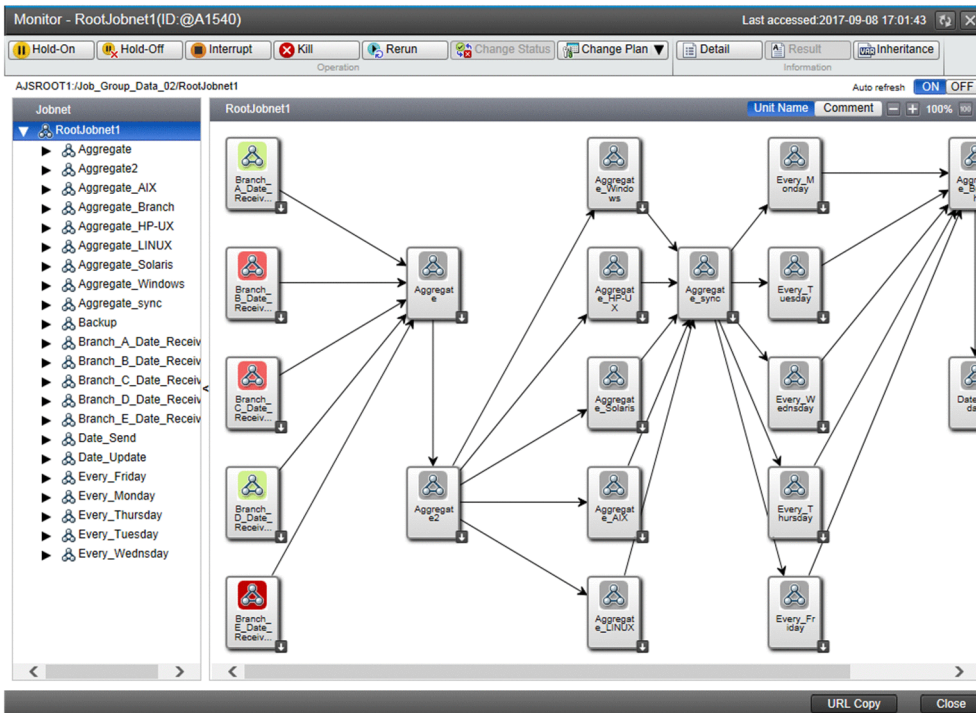
Figure 8–5: Monthly Schedule screen



### (4) Monitor dialog box

The Monitor dialog box allows you to monitor and check the execution status and results of the jobs or nested jobnets defined for a jobnet in the form of a job flow. Using the dialog box, you can also interrupt job or jobnet execution, rerun a job or jobnet, and change hold attributes.

Figure 8–6: Monitor dialog box

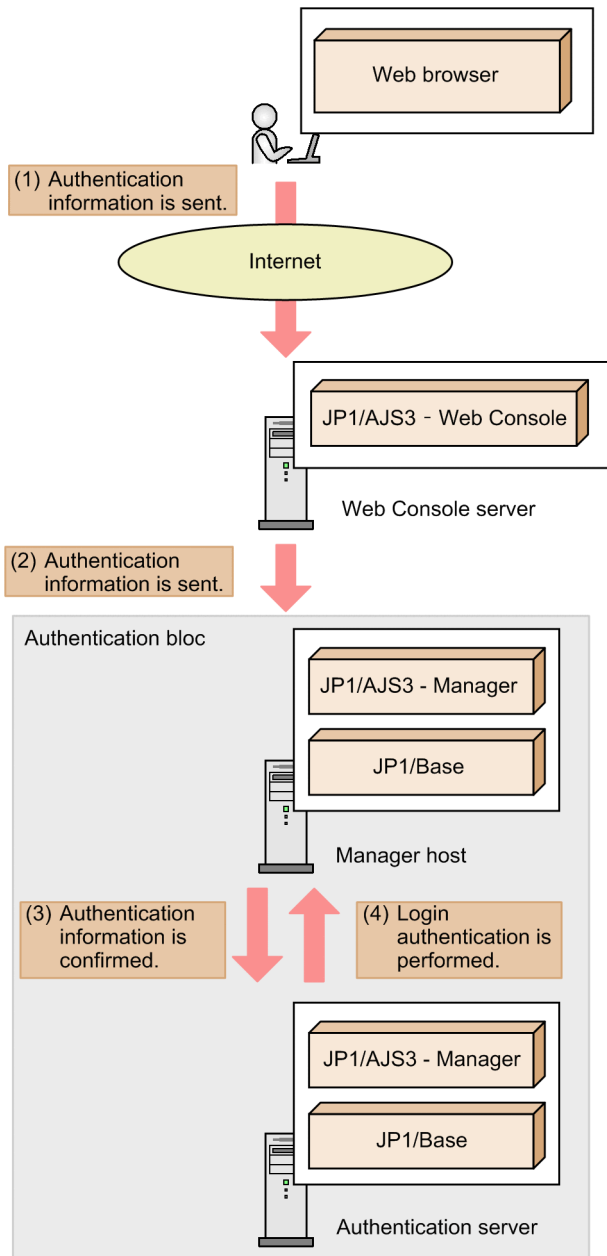


## 8.2.3 Authentication and operation mechanisms of the Web GUI

Before you can use the Web GUI to operate units of a jobnet, you must log in from the Web GUI to JP1/AJS3 - Manager as a JP1 user. Use the user authentication function of JP1/Base (authentication server) to log in.

When you enter your authentication information (JP1 user name and password) in the Web GUI, the information is forwarded via the Web Console server to the authentication server, where you are authenticated as a JP1 user. The authentication server to be used is set on the manager host to which the Web GUI connects.

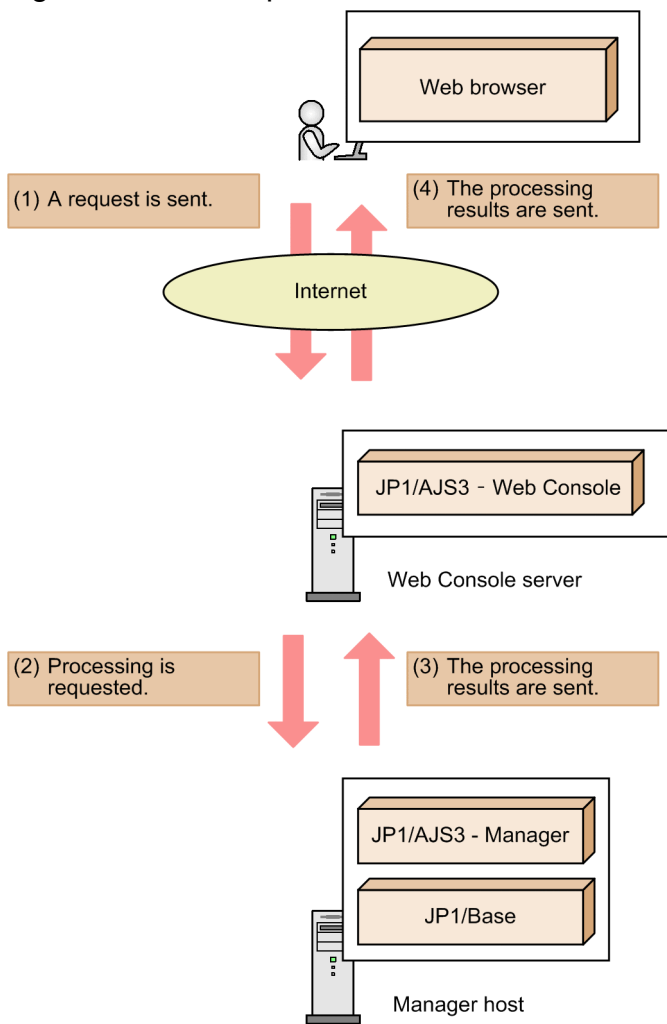
Figure 8–7: Authentication mechanism of the Web GUI



If authentication is successful, you will be able to operate the units in a jobnet within the limits of the access permissions of the JP1 user account that you used to log in. When you perform an operation on a unit from the Web GUI, a request corresponding to the operation is sent to the Web Console server. The Web Console server connects to JP1/AJS3 - Manager and requests for JP1/AJS3 - Manager to execute the operation. When JP1/AJS3 - Manager finishes processing, the execution result is sent to the Web GUI through the Web Console server.

The following figure shows the unit operation mechanism of the Web GUI.

Figure 8–8: Unit operation mechanism of the Web GUI



## 8.3 Operating JP1/AJS3 by using user applications

This section provides an overview of operating JP1/AJS3 by using user applications that were developed from the APIs provided by JP1/AJS3.

### 8.3.1 What you can do with user applications

A user application is a user-created program that is capable of interacting with a web (HTTP) interface. By sending requests from the user application to the web (HTTP) API provided by JP1/AJS3, you can perform various operations via the Internet, such as batch processing to register jobs for execution or to collect information about units.

Because user applications use the APIs provided by JP1/AJS3, the functions that can be implemented depend on the APIs that are available.

The following table describes the operations that can be performed by using user applications.

Table 8–3: Operations that can be performed by using user applications

No.	Operation	Description	
1	Executing units	Immediate execution registration	You can register jobnets for immediate execution.
2		Planned execution registration	You can register jobnets for planned execution.
3		Fixed execution registration	You can register jobnets for fixed execution.
4		Cancellation of registration for execution	You can unregister jobnets that have been registered for execution.
5	Operating units	Change plan (change time)	You can temporarily change the execution time for a unit that has already been registered for execution.
6		Change plan (execute immediately)	You can immediately execute a unit that has already been registered for execution by temporarily changing the execution schedule of the unit.
7		Change plan (execution prohibited)	You can cancel execution of a unit that has already been registered for execution by temporarily changing the execution schedule of the unit.
8		Change plan (release change)	You can undo a change that was temporarily made to the execution schedule of a unit.
9		Change hold attribute	You can set the hold attribute for a unit or remove the hold attribute from a unit.
10		Interrupt	You can interrupt the execution of a unit.
11		Kill	You can forcibly terminate the execution of a unit.
12		Rerun	You can rerun a unit.
13		Change delay monitor	You can temporarily change the delay monitoring settings for a unit that has already been registered for execution.
14		Change job status	You can change the job status.
15	Obtaining unit information	Obtaining a list of units	You can obtain the unit name, unit type, execution status, and other unit information about a specified unit or about the units belonging to that specified unit.

No.	Operation		Description
16	Obtaining unit information	Obtaining unit information	You can obtain the unit information about a specified unit.
17		Obtaining the detailed execution result of a unit	You can obtain the detailed execution result of a unit that has already been executed.
18	Other operations	Obtaining version information	You can obtain information about the versions of products.
19		Obtaining protocol version of JP1/AJS3 - Manager	You can obtain the protocol version of JP1/AJS3 - Manager to which you are connecting.

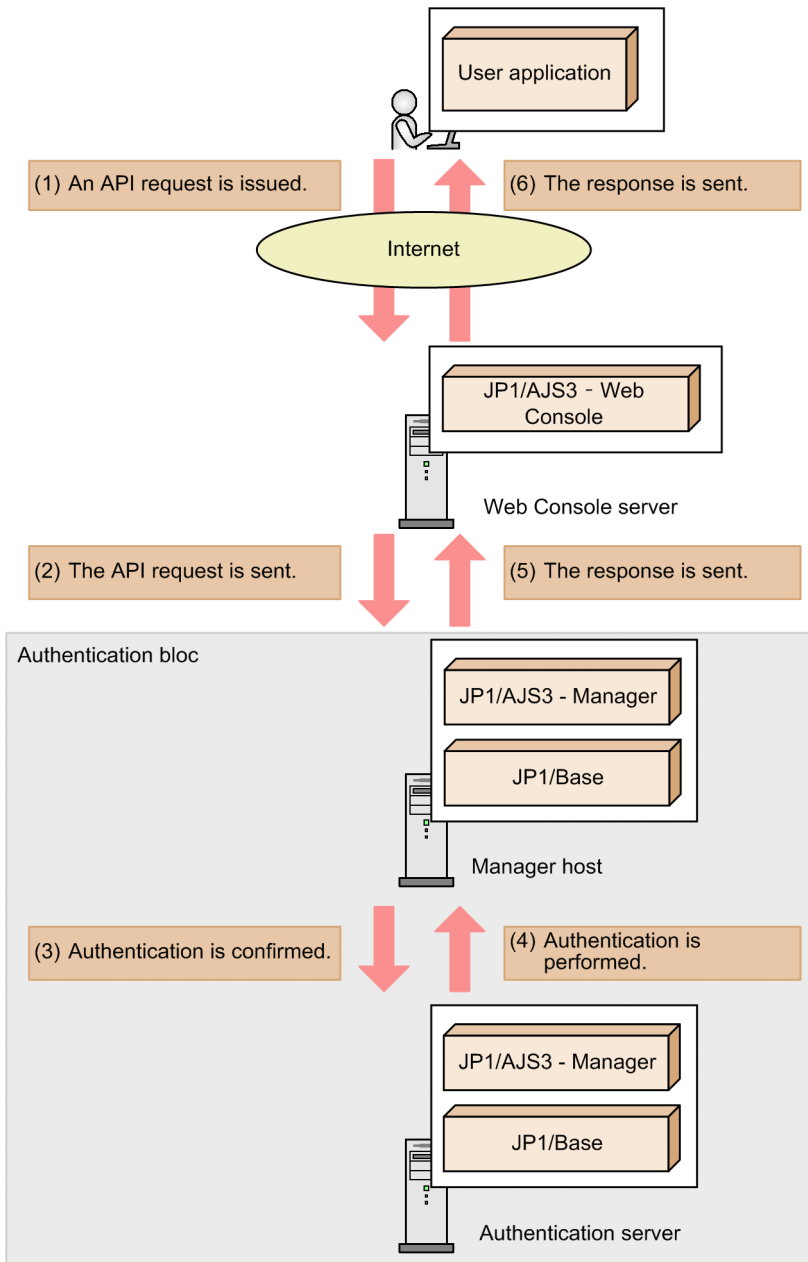
### 8.3.2 Authentication and operation mechanisms of user applications

Before you can use a user application to operate units in JP1/AJS3, you must be authenticated as a JP1 user. Use the user authentication function of JP1/Base (authentication server) for this purpose.

To be authenticated as a JP1 user, specify your authentication information (JP1 user name and password) in the API request headers. The API request to which your authentication information has been added is forwarded via the Web Console server to the authentication server, where you are authenticated as a JP1 user. The authentication server to be used is set on the manager host to which the user application connects.

If authentication is successful, JP1/AJS3 - Manager sends the result of requested processing as a response. The response is then forwarded to the user application via the Web Console server.

Figure 8–9: Authentication mechanism of user applications



Authentication for a user application is invalidated each time after a request is processed, regardless of whether the response is normal or abnormal. Therefore, if you use a user application to operate units, you must undergo authentication each time you issue an API request.

For details about APIs, see 6. *Overview of APIs* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

# 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

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 *JP1/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"> <li>• SAP R/3 systems</li> <li>• SAP BW systems</li> </ul>	<p>Run ERP applications automatically:</p> <p>You can use JP1/AJS3 to automatically run processes created using SAP R/3 systems or SAP BW 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 unit 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>	For details on linking JP1/AJS3 with an ERP system, see 5. <i>Linking ERP Systems</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .
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>	For details on linking JP1/AJS3 with HP NonStop Server, see 7. <i>Linking hp NonStop Server (Windows Only)</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .
3	JP1/AJS3 - Definition Assistant	<p>Define jobnets and jobs as a batch operation:</p> <p>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>	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> .
4	JP1/AJS3 - Web Operation Assistant	<p>Monitor and manipulate applications on different servers from a web browser:</p> <p>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.</p> <p>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>	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> .
5	HP NNM	<p>Monitor the execution status of JP1/AJS3 processes:</p> <p>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>	For details on linking JP1/AJS3 with HP NNM, see the following: <ul style="list-style-type: none"> <li>• <i>A. Monitoring Jobnets Using HP NNM</i> in the <i>JP1/Automatic Job</i></li> </ul>



No.	Linked product	Functionality in JP1/AJS3	Relevant manual
5	HP NNM	JP1/AJS3 cannot be linked with HP Network Node Manager i Software v8.10 or later.	<p><i>Management System 3 Linkage Guide</i></p> <ul style="list-style-type: none"> <li>• <i>B. Events Reported to HP NNM in the JP1/Automatic Job Management System 3 Linkage Guide</i></li> <li>• <i>C. Commands Used When Linking with HP NNM in the JP1/Automatic Job Management System 3 Linkage Guide</i></li> </ul>
6	JP1/FTP	<p>Automate file transfer jobs:</p> <p>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>	For details on linking JP1/AJS3 with JP1/FTP, see 4. <i>Linking JP1/FTP in the JP1/Automatic Job Management System 3 Linkage Guide.</i>
7	JP1/IM	<p>Monitor the execution status of JP1/AJS3 processes:</p> <p>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"> <li>• <b>Monitoring JP1 events</b> <p>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.</p> </li> <li>• <b>Monitoring execution status icons</b> <p>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> </li> </ul> <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 or Web GUI from the JP1/IM window.</p>	For details on linking JP1/AJS3 with JP1/IM, see 10. <i>Monitoring Jobnets Using JP1/IM in the JP1/Automatic Job Management System 3 Linkage Guide.</i>
8	JP1/NQSEXEC	<p>Execute processing with JP1/NQSEXEC:</p> <p>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.</p> <p>You can queue jobs from JP1/AJS3 to JP1/NQSEXEC as either QUEUE jobs or submit jobs.</p>	For details on linking JP1/AJS3 with JP1/NQSEXEC, see 9. <i>Setup for Linking JP1/NQSEXEC in the JP1/Automatic Job Management System 3 Linkage Guide.</i>
9	JP1/OJE	<p>Execute processing in conjunction with a mainframe:</p> <p>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"> <li>• VOS3</li> <li>• Another vendor's mainframe (MVS, OSIV/MSP, etc.)</li> </ul>	For details on linking JP1/AJS3 with JP1/OJE, see 6. <i>Linking Mainframes Using JP1/OJE in the JP1/Automatic Job Management System 3 Linkage Guide.</i>

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
9	JP1/OJE	<ul style="list-style-type: none"> <li>AS/400</li> </ul> <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>	For details on linking JP1/AJS3 with JP1/OJE, see <i>6. Linking Mainframes Using JP1/OJE</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .
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>	For details on linking JP1/AJS3 with JP1/Script, see <i>8. Linking JP1/Script</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .
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>	For details on linking JP1/AJS3 with JP1/Software Distribution, see <i>11. Automatic Distribution Using JP1/Software Distribution</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .
12	TELstaff	<p>Automatically report errors:</p> <p>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.</p> <p>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:</p> <p>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 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 <i>2. Linking Mail Systems</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .
14	Message queue system	<p>Define the sending and receiving of messages in a jobnet:</p> <p>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>	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.

No.	Linked product	Functionality in JP1/AJS3	Relevant manual
14	Message queue system	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.	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.
15	JP1/DH - AJE	Automate data transmission to and from JP1/DH - Server: By linking JP1/AJS3 with JP1/DH - AJE, you can automate data transmission to and from JP1/DH - Server. This linkage allows you to incorporate the sending and receiving of data into a series of operations. You can also use the scheduler to perform automatic execution. JP1/AJS3 provides a standard custom job dedicated to JP1/DH - AJE. You can define a custom job dedicated to JP1/DH - AJE in a jobnet to execute the sending and receiving of data.	For details on linking JP1/AJS3 with JP1/DH - AJE, see <i>12. Linking JP1/DH - AJE</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .
16	JP1/PFM	Facilitate investigation of the causes of job delays and errors due to system performance: By linking JP1/AJS3 with JP1/PFM, you can examine the causes of errors occurring during execution of JP1/AJS3 jobs or display a report on the performance of the manager host. This enables the implementation of more advanced and stable job operations by taking into account system performance.	For details on linking JP1/AJS3 with JP1/PFM, see <i>13. Linking JP1/PFM</i> in the <i>JP1/Automatic Job Management System 3 Linkage Guide</i> .

# 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

---

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 *10.1.1(4) Examples of configuring planning group definitions*.

### 10.1.1 Overview of planning group functions and operations

This subsection provides an overview of planning group functions and operations.

#### (1) Functions of planning groups

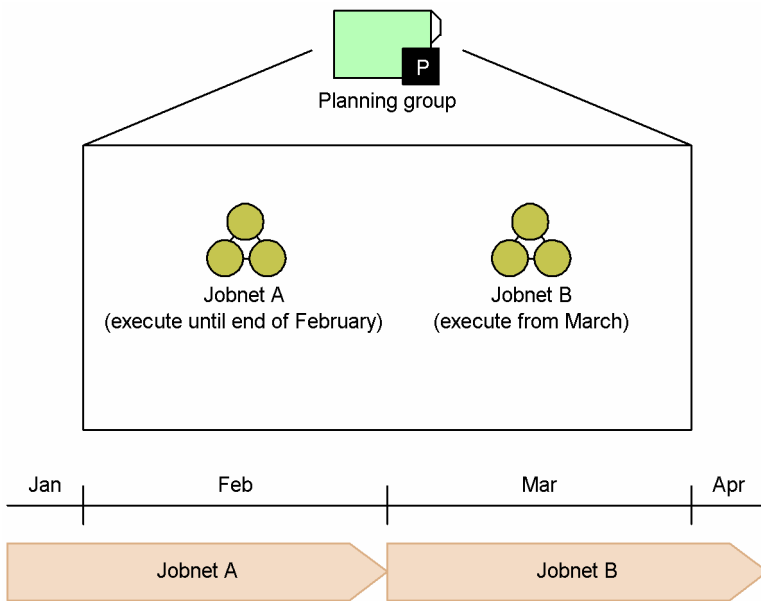
By using a planning group, you can switch smoothly between different root jobnets during system operation, without having to use a different name for each root jobnet in the GUI or in a command.

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.

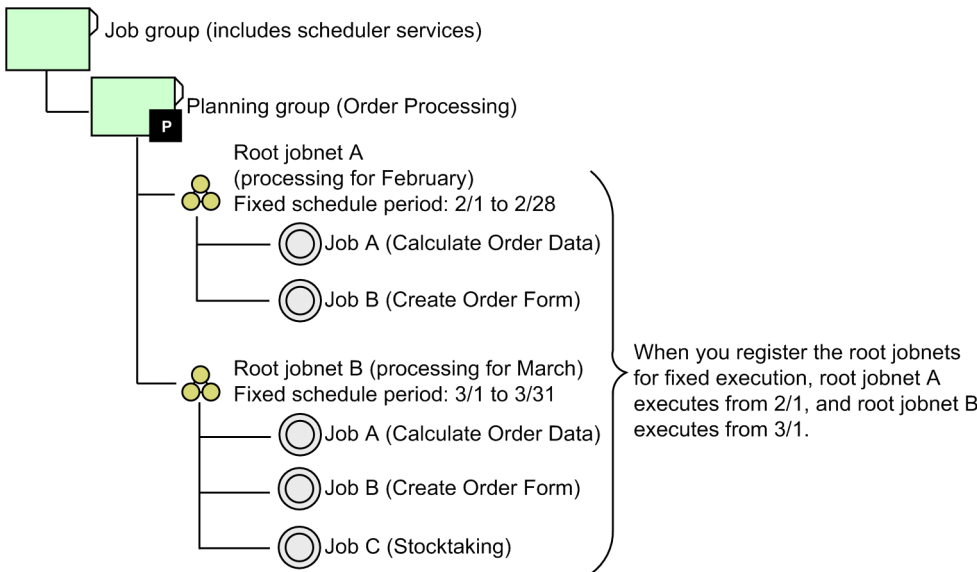
Figure 10–1: Operation of a planning group



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.

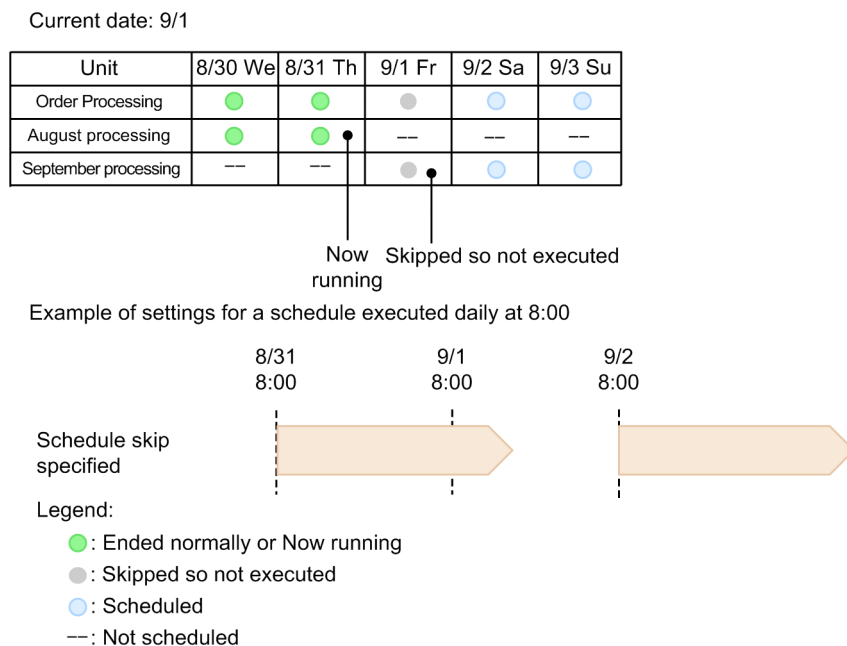
#### ■ Concurrent execution and the Schedule option

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



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

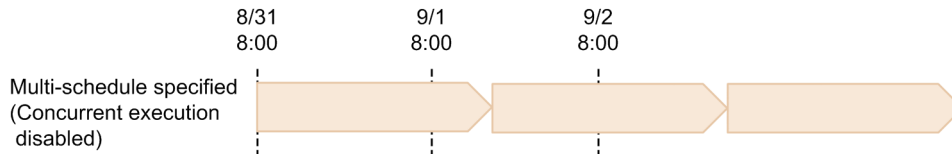
Current date: 9/1

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

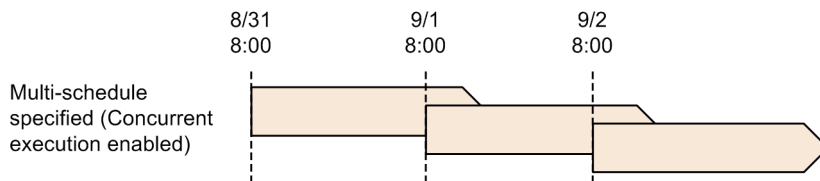
Current date: 9/1

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	---	---	●	●	●

Legend:

- : Ended normally
- : Being held
- : Scheduled
- : Not scheduled

Being held

Because the previous generation is not yet running, this generation is not executed.

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

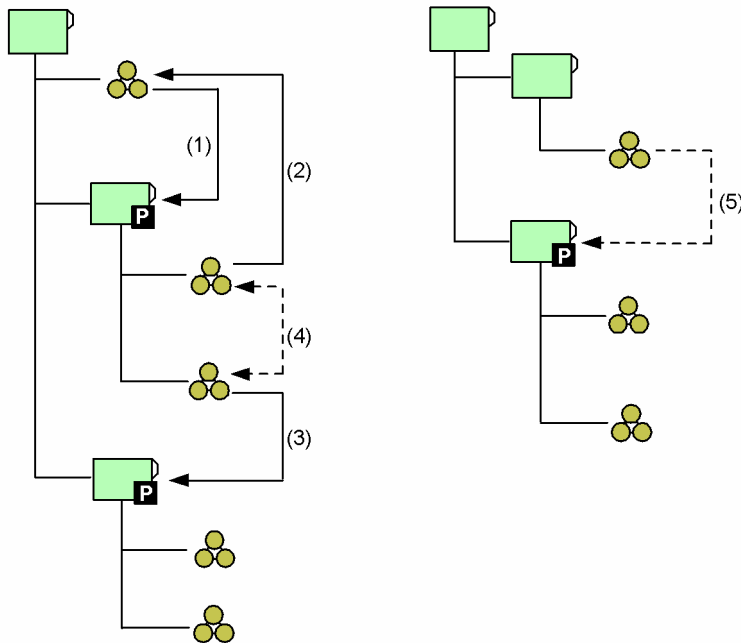
The generation scheduled for 8/31 ended abnormally before the scheduled start time of the generation scheduled for 9/1 was reached. The generation scheduled for 9/1 shifts to the Being held status.

### (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

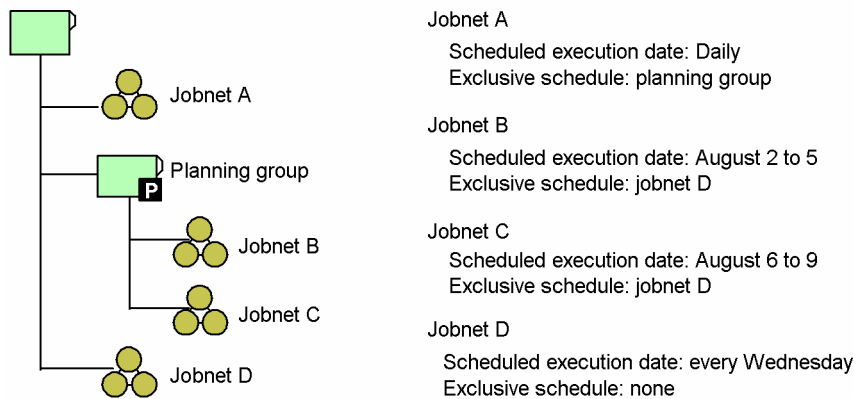
Number in figure	Valid?	Type of jobnet for which the exclusive schedule is defined	Unit specified in the exclusive schedule
(1)	Y	Jobnet	Planning group on the same hierarchical level
(2)	Y	Jobnet under a planning group	Jobnet on the same hierarchical level as the planning group
(3)	Y	Jobnet under a planning 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	--	●	--	--	--	--	--	--	●	--	--

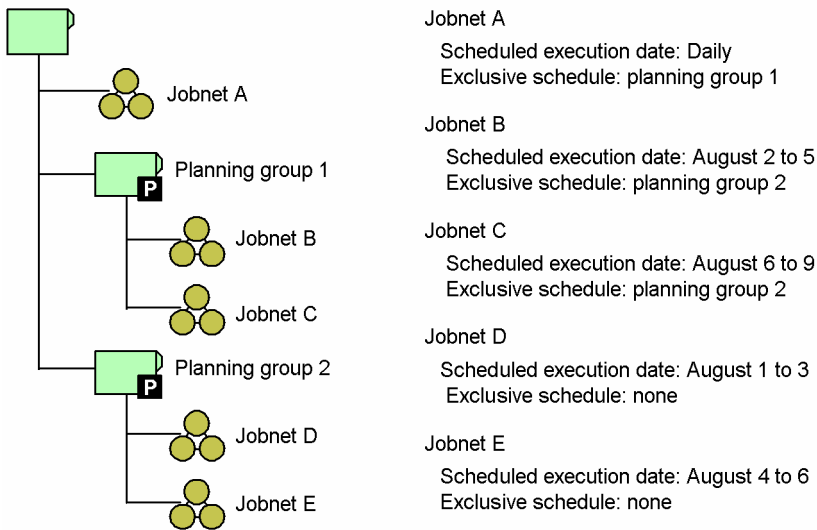
Legend:

- : 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](#).

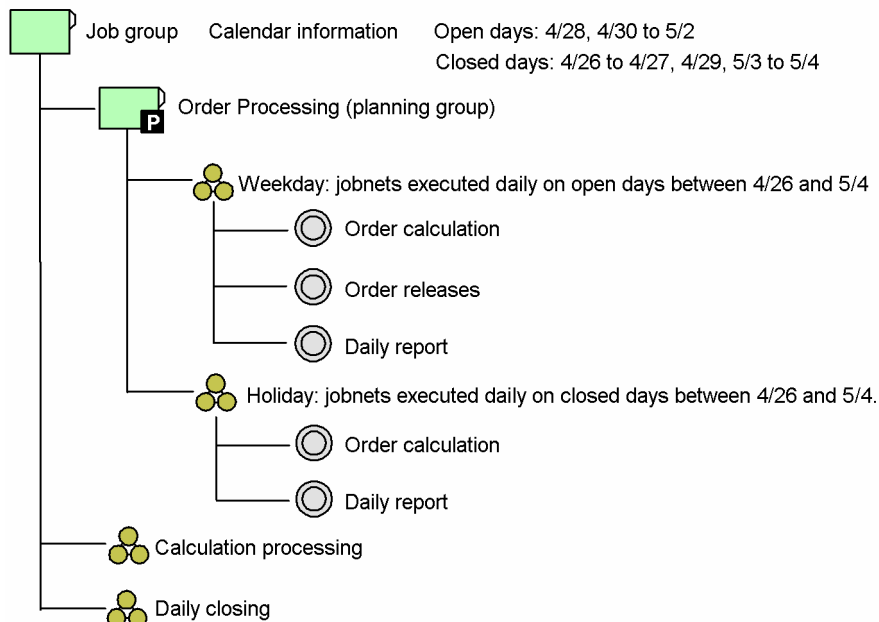
2. Define a root jobnet that executes in the period where you canceled jobnet registration or prohibited jobnet execution.

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 <sup>#</sup>	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 <sup>#</sup>	EN	EN	EN	EN	SE	SE	SE	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

#

*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 *12.3.42 Preferences dialog box* in the *JP1/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	--	--
Daily report	--	--	--	EN	EN	EW	--	--

Legend:

EN : Ended normally

EW : Ended with warning

SE: Scheduled for execution (waiting for start time)

-- : Not scheduled

#

*Order processing* is a planning group.

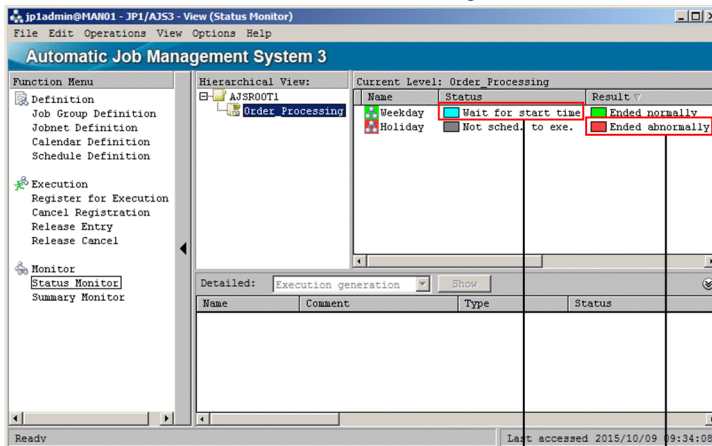
If you compare the results of the jobnet **Holiday** and the jobnet **Weekday**, you can see that the results of the jobnet **Holiday** are closer to the current date and time. Imagine that you now select an upper-level unit to the planning group to display the results of the planning group at the end 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.

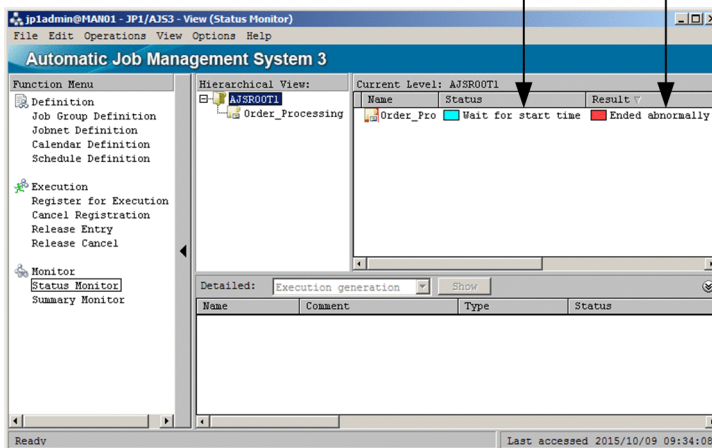
The following figure shows an example of how the status and execution results for a planning group are displayed in the list area of the JP1/AJS3 - View window.

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

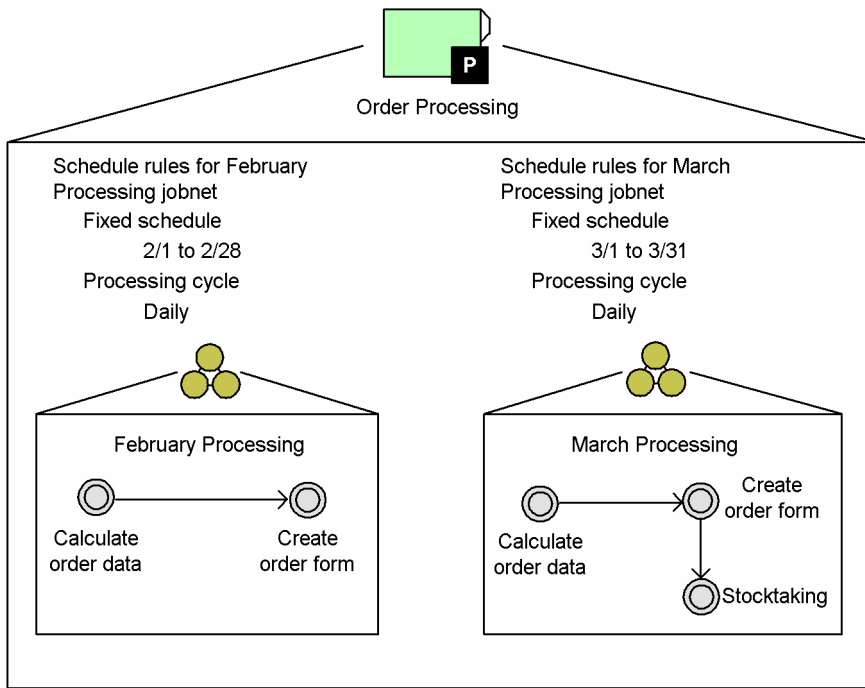


Table 10–5: Image displayed in the schedule area of the Monthly Schedule window for a planning group that switches jobnets every month

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 <sup>#</sup>	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

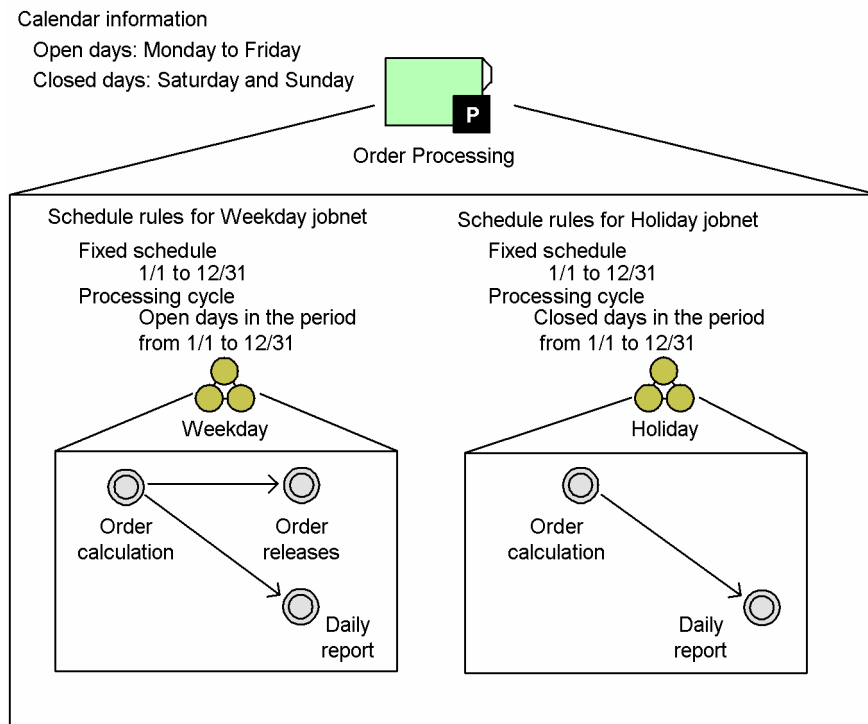


Table 10–6: Image displayed in the schedule area of the Monthly Schedule window for a planning group that alternates between jobnets

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 <sup>#</sup>	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

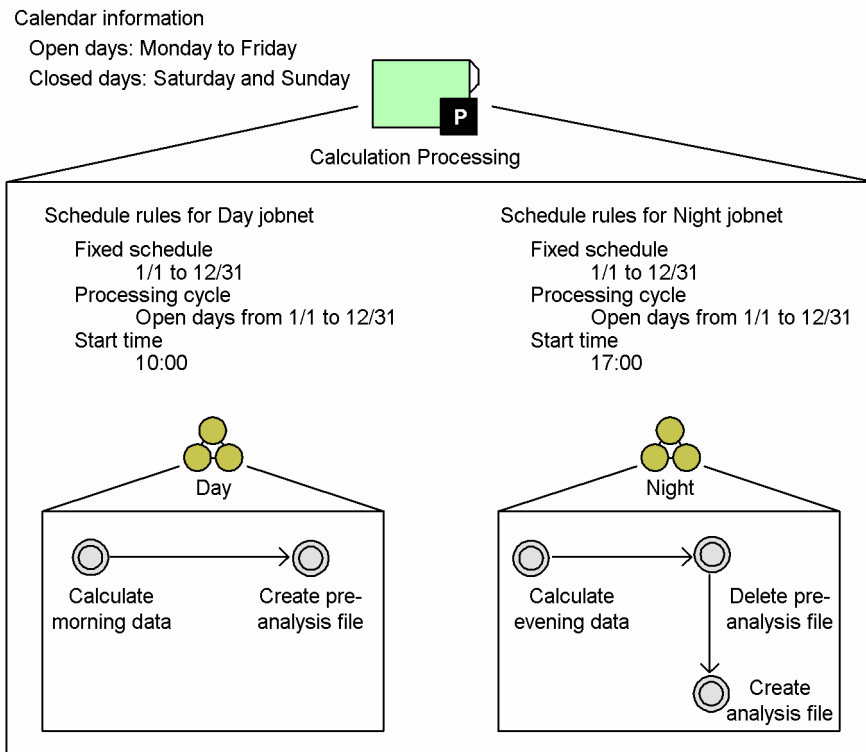


Table 10–7: Image displayed in the schedule area of the Monthly Schedule window for a planning group that switches between jobnets within the same day

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 <sup>#</sup>	SE	--	--	SE	SE	SE	SE	SE	--	--	SE	SE
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

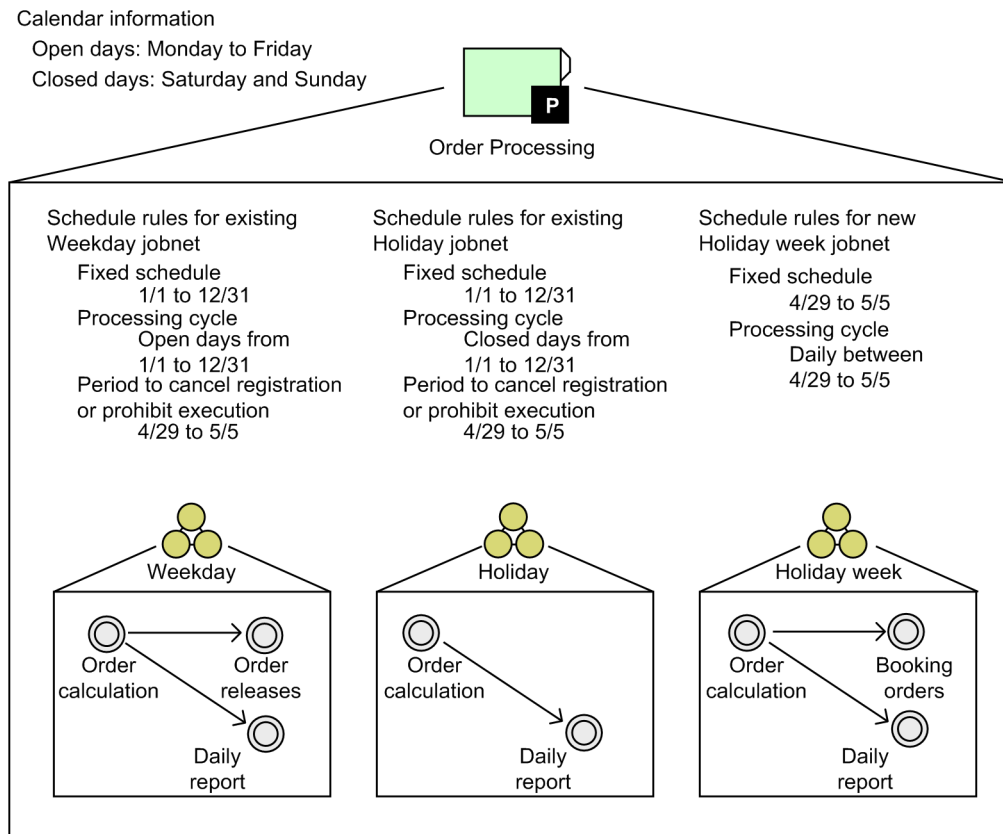


Table 10–8: Image displayed in the schedule area of the Monthly Schedule window when a jobnet is added

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 <sup>#1</sup>	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE
Weekday	--	--	SE	--	--	--	--	--	--	--	SE	SE
Holiday	SE	SE	--	--	--	--	--	--	--	--	--	--
Holiday week <sup>#2</sup>	--	--	--	SE	SE	SE	SE	SE	SE	SE	--	--

Legend:

SE : Scheduled for execution

-- : Not scheduled

#1

*Order Processing* is the planning group.

#2

*Holiday week* is the added jobnet.

## (5) Supplementary notes

This subsection contains additional information about planning groups.

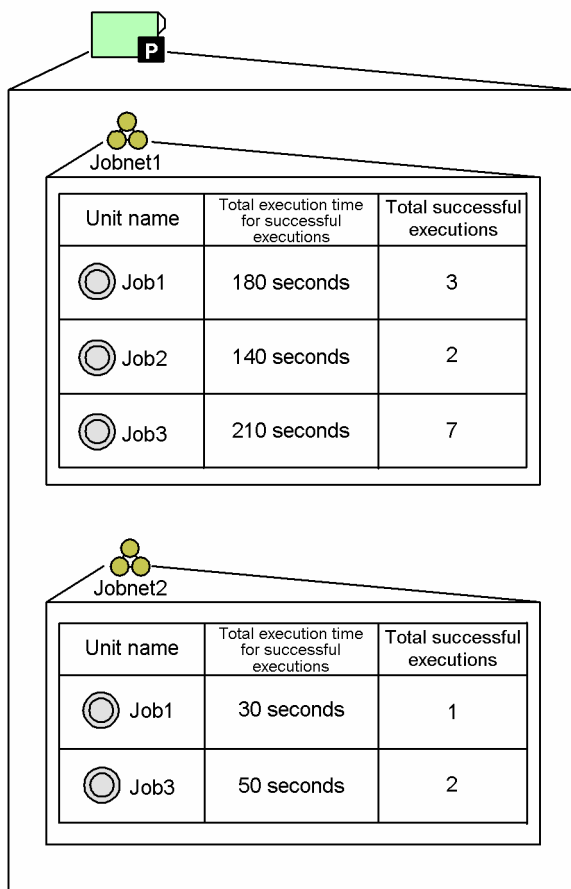
### (a) Specifying unit names when using a command to manipulate a planning group

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 *JP1/Automatic Job Management System 3 Command Reference*.

### (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 character codes of the root jobnet names. For alphanumeric characters, generation numbers are assigned in the order of the ASCII character codes. For Japanese characters, generation numbers are assigned in the order of the Japanese Shift JIS, Japanese EUC or UTF-8 character codes. The following figure shows an example of generation numbers assigned in the order of the character 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.

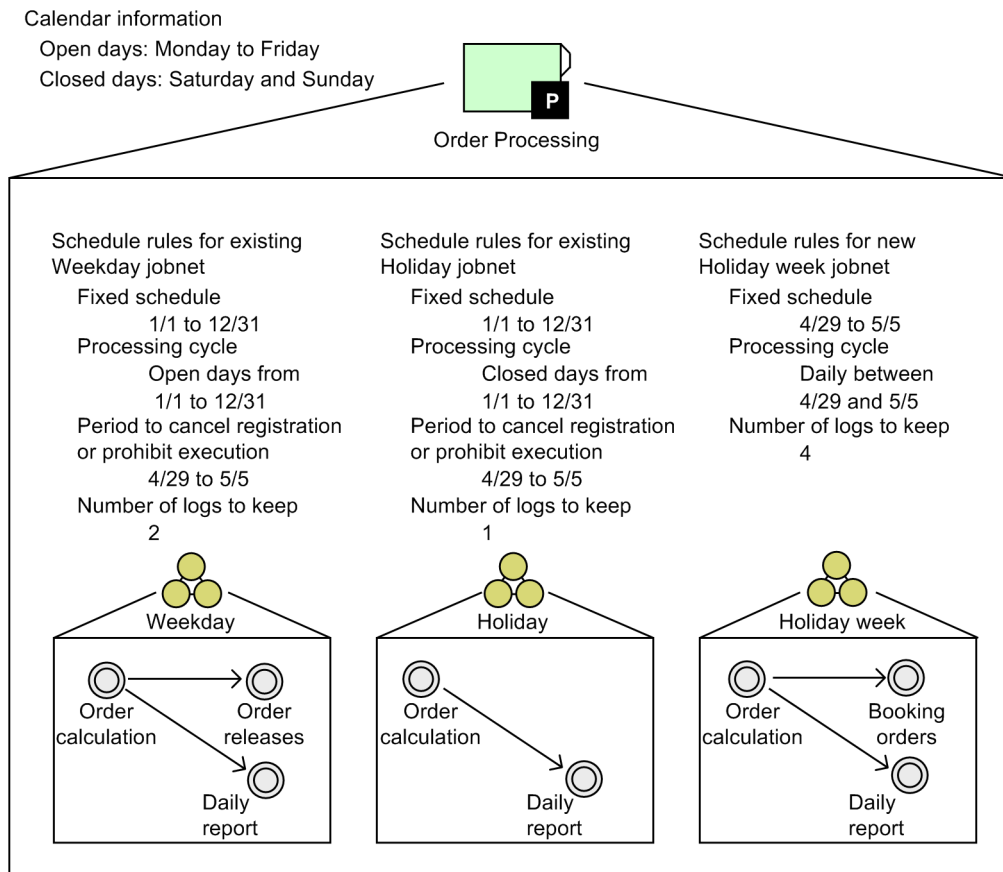
When you register jobnets for execution in order from the jobnet with the earliest scheduled start time, you ensure that the generation number of the registered jobnet remains 1 because when the first jobnet ends, the jobnet with the next earliest start time executes. Therefore, you need only specify 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 <sup>#</sup>	EN	--	--	--	--	EN	EN	EN	EN	--	EN	EN
Weekday	--	--	--	--	--	--	--	--	--	--	EN	EN
Holiday	EN	--	--	--	--	--	--	--	--	--	--	--
Holiday week	--	--	--	--	--	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

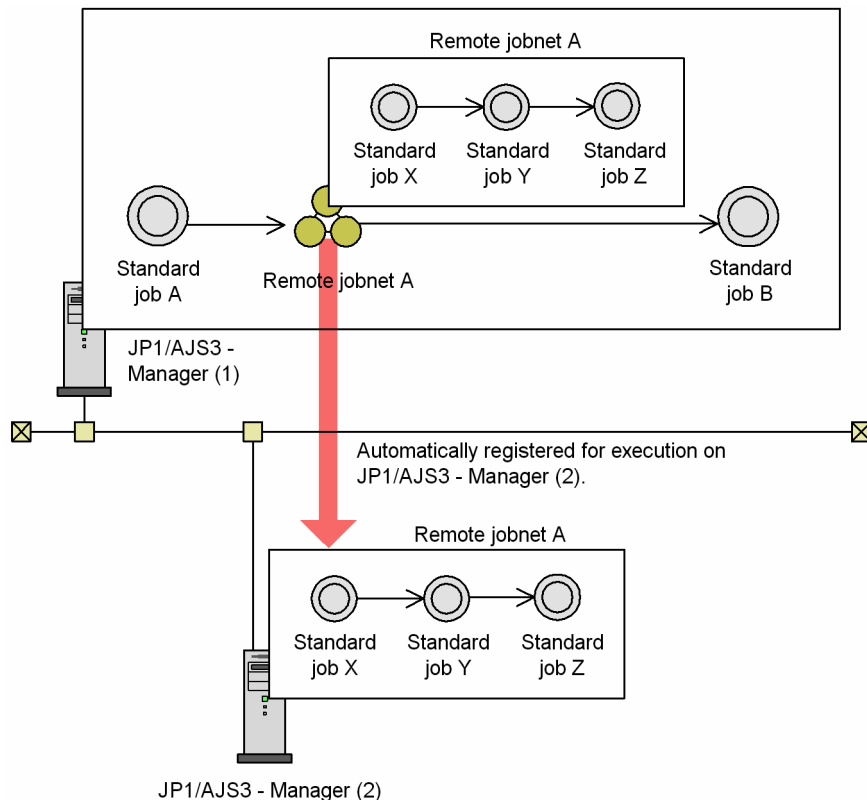
A *remote jobnet* is a jobnet defined on the local manager host, but transferred to a different manager host for execution. The use of remote jobnets allows the workload of JP1/AJS3 - Manager, when executing jobnets or jobs defined under the remote jobnet, to be distributed over multiple hosts.

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 destination 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 return 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 *JP1/Automatic Job Management System 3 Configuration Guide* (Windows) or see *15.1.1 Settings for starting multiple scheduler services* in the *JP1/Automatic Job Management System 3 Configuration Guide* (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 *20.4 Setting up the scheduler service environment* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

- 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.
- The scheduler service to be used for executing remote jobnets is determined by the settings for the REMNETSERVICENAME environment setting parameter for JP1/AJS3 - Manager on the transfer destination host.

## 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.

### 10.4.1 Overview and feature description of QUEUE jobs and submit jobs

This subsection provides an overview of QUEUE jobs and submit jobs, and describes their features.

#### (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?

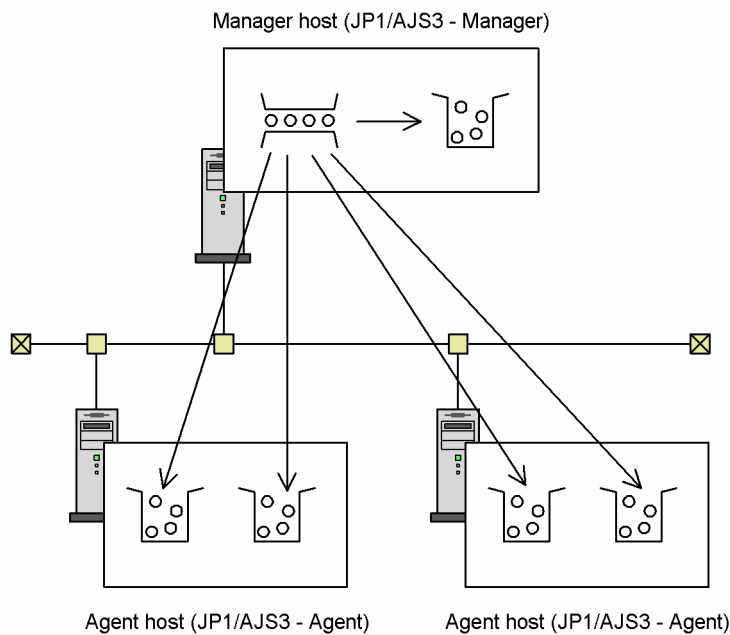
In JP1/AJS3, jobs can be defined and executed in a jobnet, or they can be executed singly by command, via direct registration in a queue in JP1/AJS3 - Manager. The jobs in the queue are sent to an agent connected to the queue for execution. If you specify the agent when directly registering a job for 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 `jpqjobs` 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 `jpqjbsub` 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.

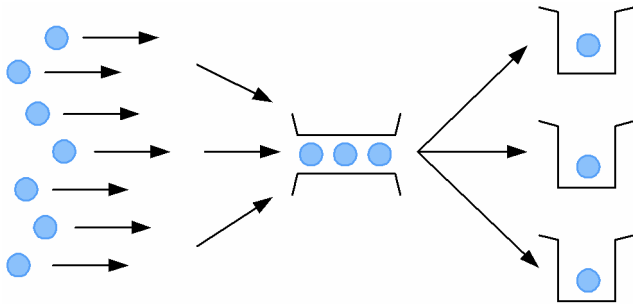
By using a queue, these sorts of problems can be avoided and jobs can be executed more efficiently. When the number of jobs to be executed concurrently reaches a set limit, subsequent jobs are made to wait and are passed in turn to the agent as preceding jobs finish.

#### 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 *JP1/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 *JP1/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	<p>Manager</p>	<p>Agent</p>	The manager registers the job as requested, and places it in the queue. The job status becomes <i>Now queuing</i> .
Now running	<p>Manager</p>	<p>Agent</p>	The manager takes the job from the queue and transfers it to an agent, requesting the agent to execute the job.

Job status		Description
Now running		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.
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 `jpqimport` command. You can change the queue and agent configuration while JP1/AJS3 services are active, using the `jpqqueadd` 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 *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

For details about the `jpqimport` and `jpqqueadd` commands, see *1.5.9(1) Commands available in an execution environment for queue or submit jobs* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

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 *JP1/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 *JP1/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` (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 execution environment configuration definition file for QUEUE jobs and submit jobs (`jpgqsetup.conf`) 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 4. *Commands Used for Special Operation* in the manual *JP1/Automatic Job Management System 3 Command Reference*.

## (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 *JP1/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<sup>#</sup> 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:

Agent usage rate = *number of active jobs / maximum number of concurrent jobs*

#### Cautionary note

When jobs are to be distributed, there might be multiple execution agents that have the same priority and agent usage rate. In this case, to balance the load equally, the best way is to distribute the jobs by selecting the execution agent whose idle time is longest.

For details on defining priorities, see 7.1.7 *Method of determining the agent host to which a job is distributed* in the *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

### **(b) Detecting failures and distributing jobs to execution hosts (agent hosts)**

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 `jqimport` or `jqresadd` 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 `jqjobs` 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/NQSEXEC 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 **Exec-agent** in the detailed definition of a job. If the host name specified in **Exec-agent** is set in an execution agent profile, the function for execution agent restriction can also be used for queueless jobs. For details about execution agent restriction, see *2.5.1(6) Execution agent restriction* in the *JP1/Automatic Job Management System 3 System Design (Configuration) Guide*.

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 *20.10 Setting up the queueless job execution environment* in the *JP1/Automatic Job Management System 3 Configuration Guide*.

### 10.5.1 Overview and feature description of queueless jobs

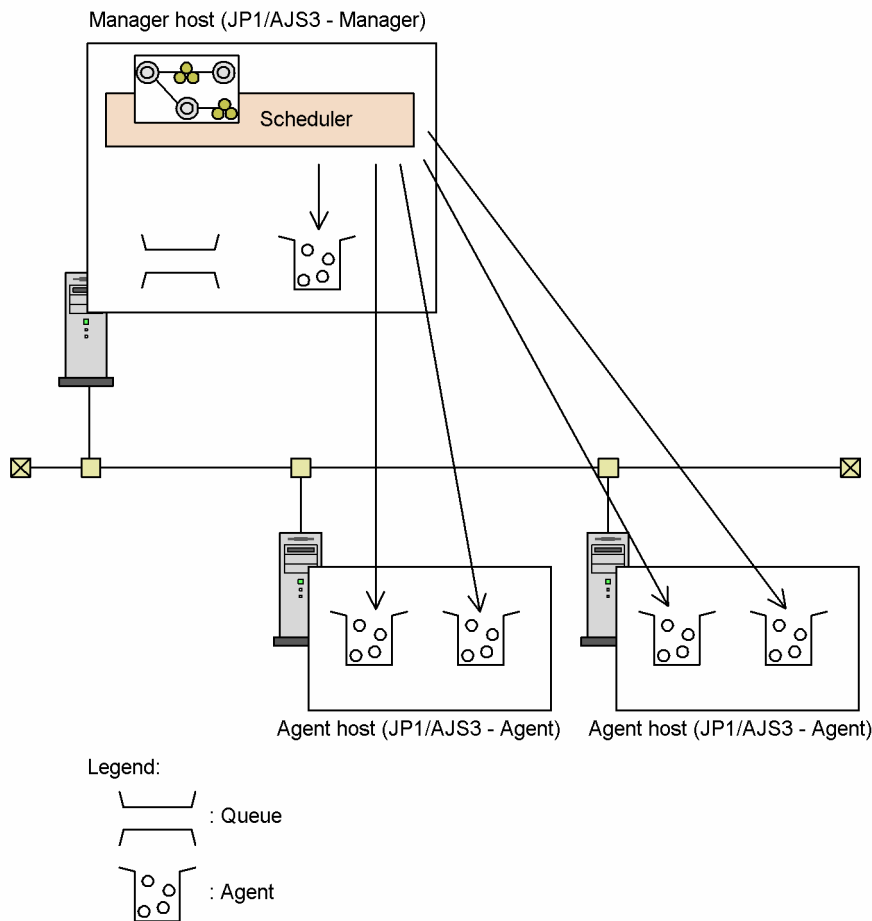
This subsection provides an overview of queueless jobs and describes their features.

#### (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 provides an overview of executing queueless jobs.

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 `jajs_config` command to set these parameters. For details about the parameters, see *20.10 Setting up the queueless job execution environment* in the *JPI/Automatic Job Management System 3 Configuration Guide*.

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 *4. Commands Used for Special Operation* in the manual *JPI/Automatic Job Management System 3 Command Reference*.

### (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 *JPI/Automatic Job Management System 3 Configuration Guide* (Windows) or see *15.4.1 Executing jobs with a class specified in a queueless job environment* in the *JPI/Automatic Job Management System 3 Configuration Guide* (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.

## (4) OS user when queueless jobs are executed

In the same way as for the jobs for which **Standard** is specified in **Exec. Service**, the OS user who corresponds to the JP1 user by user mapping at agent host executes the queueless jobs. When a file is transferred by using a queueless job, user mapping is performed on the manager host. Therefore, you must ensure that the JP1 user was mapped by an OS user via user mapping on the manager host.

Cautionary notes:

1. For server host name of the user mapping definitions, you must specify the manager host name resolved by the agent host.

2. The user mapping is cached when the queueless jobs are executed. After you change the user mapping or the information about the OS user, you must clear the user mapping cache. The cache can be cleared by one of the following operations.
  - Restart the queueless agent service and queueless file transfer service.
  - Execute the `ajsqlalter` command with the `-r` option specified.

# Appendixes

## A. Functions that Cannot Be Used in the Standard Configuration Without ISAM or in the Compatible ISAM Configuration

---

This appendix describes the functions that cannot be used in the standard configuration without ISAM, and the functions that cannot be used in the compatible ISAM configuration.

### A.1 Functions that cannot be used in the standard configuration without ISAM

The following functions cannot be used in the 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 (j<sub>p</sub>q<sub>xxxx</sub> commands).  
For details about the commands used in the execution environment for QUEUE jobs and submit jobs, see *1.5.9(1) Commands available in an execution environment for queue or submit jobs* in the manual *JP1/Automatic Job Management System 3 Command Reference*.
- Execute jobs on the local host from JP1/AJS2 - Client Toolkit or JP1/AJS2 - SDK on a remote host.
- Execute jobs on the local host from another system such as JP1/NQSEXEC or JP1/OJE for VOS3.

### A.2 Functions that cannot be used in the compatible ISAM configuration

The following functions cannot be used in the 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.
- Specify unit names, comments, and other items with the same number of characters as in the Shift-JIS environment, even in a UTF-8 environment.
- Disaster recovery operations.
- Use an operation profile for an operation.
- Use the connection-source restriction function.
- Using wait conditions to control the order of unit execution.
- Restrict the maximum number of allowed concurrent sessions for scheduler services.
- Display the execution schedule for unregistered units in the Monthly Schedule window of JP1/AJS3 - View.
- Prevent scheduler services for which the user has no access privileges from appearing in JP1/AJS3 - View.



- Change the passwords of JP1 users by using JP1/AJS3 - View.
- Set a hold attribute when registering a jobnet for execution.
- Reschedule jobnet execution on an all-day basis.
- Execute JP1/AJS3 commands from JP1/AJS3 - View.
- Operations related to replicating a JP1/AJS3 - Manager virtual machine.
- Operation for changing the agent host information used for name resolution (`ResolveAgentInfo` environment setting parameter).
- Perform connection and operations by using JP1/AJS3 - Web Console.
- Functions for defining and executing flexible jobs.
- Functions for defining HTTP connection jobs.
- Functions that use encrypted communication.
- Use the job distribution delay reduction function.
- Execute the `ajscalutil` command.
- Transition the job meeting specific conditions to *Wait for prev. to end* or *Not executed + Ended* status, when the scheduler service is restarted, and not to *Unknown end status*.
- Increase the maximum number of execution agents that can be registered in a manager host to 2,048.

## B. Version Revisions

---

This appendix lists the changes in each version of the JP1/AJS series programs.

### B.1 Revisions in 11-50

The following lists the revisions in 11-50 for each program.

#### (1) JP1/AJS3 - Manager

- The Web GUI now support the following functions:
  - Viewing monthly schedules
  - Adding execution schedules
  - Change plan (change time)
  - Change plan (execute immediately)
  - Change plan (execution prohibited)
  - Change plan (release change)
  - Change delay monitor
- The API now supports the function to acquire unit information.
- The maximum size of the Java heap on the Web Console server can now be changed, and the default value was changed from 512 MB to 2,048 MB.
- A function was added to enable you to change the colors indicating unit statuses displayed in the windows and dialog boxes of the Web GUI.
- The map area in the Monitor dialog box and the Monitor screen of the Web GUI can now switch between displaying unit names and comments.
- The following functions were added to improve job throughput when multiple scheduler services are active:
  - Function to reduce the number of ports that are used during job execution
  - Function to output trace logs for each scheduler service

In addition to the changes to internal processing, the JP1/AJS3 performance processing that runs when multiple root jobnets start at the same time was improved. As a result, job throughput when a single scheduler service is active was also improved.

- The maximum number of execution agents that can be registered on one manager host can now be increased to 2,048.
- To enable the comments on units to be viewed in JP1/IM windows, commenting on units was added as an extended attribute (specific information) of a JP1 event whose OBJECT\_TYPE is JOBNET or JOB.
- Commands to specify settings of environment setting parameters were integrated into the `jajs_config` command.
- When the scheduler service is restarted, a job that meets specific conditions now transitions to *Wait for prev. to end* or to *Not executed + Ended* status (and *not to Unknown end status*). This enables you to determine, from the job status after the restart of the scheduler service, whether a user program has started.

## (2) JP1/AJS3 - Agent

- Commands to specify settings of environment setting parameters were integrated into the `jajs_config` command.

## (3) JP1/AJS3 - Agent Minimal Edition

- Descriptions were added concerning JP1/AJS3 - Agent Minimal Edition, which limits the number of jobs that can be executed per day to 10.

## B.2 Revisions in 11-10

The following lists the revisions in 11-10 for each program.

### (1) JP1/AJS3 - Manager

- The Web GUI and API now support the following functions:

#### Web GUI

- Check execution schedule
- Interrupt
- Kill
- Rerun
- Change job status
- Change detailed definitions for PC jobs and Unix jobs
- Filter monitoring targets by execution status

#### API

- Fixed execution registration (specifying a specific date and time)
  - Change plan (change time)
  - Change plan (execute immediately)
  - Change plan (execution prohibited)
  - Change plan (release change)
  - Change hold attribute
  - Interrupt
  - Kill
  - Rerun
  - Change delay monitor
  - Change job status
  - Obtain execution result details for units
  - Obtain the protocol version of JP1/AJS3 - Manager
- JP1/IM and JP1/AJS3 - Web Console can now be linked so that the Web GUI can be displayed from an email sent by an automated action of JP1/IM - View or JP1/IM.

- JP1/PFM and JP1/AJS3 - Web Console can now be linked so that PFM - Web Console can be launched from the Web GUI.
- The `ajscalutil` command was added. This command calculates multiple calendars and outputs the results.
- A function was added to set a hold attribute for the first execution generation created by registering a jobnet for planned execution or fixed execution.
- The job distribution delay reduction function was added. (When a job is to be executed concurrently on multiple execution agents, if a communication failure occurs on some execution agents, this function reduces the delay in job distribution to the normally operating execution agents.)
- In HP-UX, AIX, and Linux, the architecture of most JP1/AJS3 - Manager processes was changed to 64-bit architecture. Note that the following processes were not changed to 64-bit architecture: `jpqman32`, `jqimport`, `jqexport`, `jqdbcond`, `jajs_maintain`, and `jajs_migrate`.
- The broadcast function was added. This function distributes a job to multiple execution agents by using a flexible job.
- When the `ajswbsetup` command is run with the `-s` option (option for setting up the standby system) specified, the command now ends normally even if no shared disk is mounted.

## (2) JP1/AJS3 - Agent

- In HP-UX, AIX, and Linux, the architecture of most JP1/AJS3 - Agent processes was changed to 64-bit architecture.
- The broadcast function was added. This function distributes a job to multiple execution agents by using a flexible job.

## (3) JP1/AJS3 - View

- A function was added to set a hold attribute for the first execution generation created by registering a jobnet for planned execution or fixed execution.
- The broadcast function was added. This function distributes a job to multiple execution agents by using a flexible job.
- The multiuser remote desktop connection function was added. This function allows multiple users to use JP1/AJS3 - View via remote desktop connections.
- A function was added that allows each JP1 user to use different custom jobs in JP1/AJS3 - View.

## B.3 Revisions in 11-00

The following lists the revisions in 11-00 for each program.

### (1) JP1/AJS3 - Manager

- The supported language types were changed. The specific combinations of languages that are supported when using JP1/AJS3 in multi-language environments was also changed.
- JP1/AJS3 - Web Console was added as a component of JP1/AJS3 - Manager. JP1/AJS3 - Web Console allows you to monitor units over the Internet from a web browser, and develop custom programs that use APIs to perform operations on units.
- JP1/AJS3 - Print Option was incorporated into JP1/AJS3 - Manager. You can use JP1/AJS3 - Print Option to display and print unit information with a customized layout.
- Messages between hosts such as agent hosts and JP1/AJS3 - View hosts can now be encrypted using SSL.

- A *flexible job* was added as a job type. This job allows processing to be executed on an agent host that cannot be managed directly from the manager host. For example, you might use a flexible job with an execution agent in an auto-scaling environment. Although the definition of a flexible job is not OS-dependent, flexible jobs can only be executed in Windows and Linux environments.
- An *HTTP connection job* was added as a job type. This job can send requests to HTTP interfaces such as web APIs provided by cloud environments and web servers. Although the definition of an HTTP connection job is not OS-dependent, HTTP connection jobs can only be executed in Windows and Linux environments.
- The initial values of the following environment setting parameters were changed to reflect the recommended values:
  - UNITDEFDIVIDEUPDATE
  - AJSPRINTSORTUNITINF
  - ReceiveFileSizeStatus
  - LoadUserProfile
  - IsAccessTokenCache
  - MergeEnvMode
  - NotifyJobStateCount
  - PassingInfoUsePreRc
  - FileWriteMode
  - ResolveAgentInfo
  - RetryAgtStartNotification
- The environment setting parameters that require assessment in the design stage of JP1/AJS3 were reviewed, and corresponding improvements were made to the manual.
- An option was added that causes JP1/AJS3 - Manager to continue waiting when the status of a unit whose end is being waited for changes after re-execution of its succeeding unit. The behavior of JP1/AJS3 - Manager when waiting is enabled after a new installation of JP1/AJS3 was also changed.
- The following OS is supported:
  - CentOS 6(x64)
  - CentOS 7
  - Linux 6.1(x64)
  - Linux 7.1
  - Oracle Linux 6(x64)
  - Oracle Linux 7
  - SUSE Linux 12
- The following OS is no longer supported:
  - Windows Server 2008 (excluding Windows Server 2008 R2)
  - Windows Server 2003
  - Windows Server 2003(x64)
  - Linux 6.1(x86) or later
  - Linux 5.1(x86) or later
  - Linux 5.1(AMD/Intel 64) or later

- Support for Outlook 2003 linkage was eliminated. JP1/AJS3 - Manager now supports linkage with Outlook 2013 (32 bit) and Outlook 2010 (32 bit).
- JP1/AJS3 Console is now compatible with earlier versions.

## (2) JP1/AJS3 - Agent

- The supported language types were changed. The specific combinations of languages that are supported when using JP1/AJS3 in multi-language environments was also changed.
- Messages exchanged with JP1/AJS3 - Manager can now be encrypted using SSL.
- Flexible jobs can now be executed (Windows and Linux only).
- HTTP connection jobs can now be executed (Windows and Linux only).
- The initial values of the following environment setting parameters were changed to reflect the recommended values:
  - LoadUserProfile
  - IsAccessTokenCache
  - MergeEnvMode
  - NotifyJobStateCount
  - FileWriteMode
  - ResolveAgentInfo
  - RetryAgtStartNotification
- The environment setting parameters that require assessment in the design stage of JP1/AJS3 were reviewed, and corresponding improvements were made to the manual.
- The following OS is supported:
  - CentOS 6(x64)
  - CentOS 7
  - Linux 6.1(x64)
  - Linux 7.1
  - Oracle Linux 6(x64)
  - Oracle Linux 7
  - SUSE Linux 12
- The following OS is no longer supported:
  - Windows Server 2008 (excluding Windows Server 2008 R2)
  - Windows Server 2003
  - Windows Server 2003(x64)
  - Linux 6.1(x86) or later
  - Linux 5.1(x86) or later
  - Linux 5.1(AMD/Intel 64) or later
- Support for Outlook 2003 linkage was eliminated. JP1/AJS3 - Manager now supports linkage with Outlook 2013 (32 bit) and Outlook 2010 (32 bit).

### (3) JP1/AJS3 - View

- The supported language types were changed. The specific combinations of languages that are supported when using JP1/AJS3 in multi-language environments was also changed.
- Messages exchanged with JP1/AJS3 - Manager can now be encrypted using SSL.
- Flexible jobs can now be defined in the Define Details dialog box.
- HTTP connection jobs can now be defined in the Define Details dialog box.
- The paste commands in the menu of the Jobnet Editor window were replaced with two new commands. One is the **Paste (Units and relations)** command, which pastes icons, the coordinates of the icons, and the relation lines between icons. The other is the **Paste (Units only)** command which pastes icons and their coordinates.
- An option was added that allows you to set the display order of items in the tree area and list area of the JP1/AJS3 - View window (Main window).
- An option was added that allows you to change where the current day is positioned in the Time area and Schedule area of the Monthly Schedule window.
- The settings below now apply as the default settings for all windows. You can customize these settings in JP1/AJS3 - View.
  - A question mark (?) is used as the icon for unregistered custom jobs.
  - Units in the tree area are displayed in the same order as the list area.
- The settings below now apply as the default settings in JP1/AJS3 - View windows. You can customize these settings in JP1/AJS3 - View.
  - Job groups and planning groups are selected in the list area when the JP1/AJS3 - View window (Main window) is displayed
  - The same units remain selected and the scroll position is maintained when you refresh the window contents or specify list filtering
  - A confirmation message is displayed before deleting a defined schedule rule for a nested jobnet for which the **Depends on upper-level jobnet** check box is selected
  - Jobnets that have been registered for execution appear in the list area when **Jobnet Definition** is selected in **Function Menu**
- The following settings now apply as the default settings in the Jobnet Editor window. You can customize these settings in JP1/AJS3 - View.
  - Nested jobnets and nested remote jobnets are selected in the map area when the Jobnet Editor window is displayed
  - Users are prevented from opening multiple Jobnet Editor windows
- The following settings now apply as the default settings in the Jobnet Monitor window. You can customize these settings in JP1/AJS3 - View.
  - The contents of the Execution Result Details dialog box are not automatically refreshed
  - The Execution Result Details dialog box is displayed in large size
  - Users are prevented from opening multiple Jobnet Monitor windows
  - Jobs are displayed directly in the map area when displaying the Jobnet Monitor window from JP1/IM - View
- Color images are now used to represent the execution status, execution results, and delays in **Status** and **Result** in the list area and **Status** in the detailed information area of the JP1/AJS3 - View window (Main window).
- The following units are now hidden by default in the icon list in the Jobnet Editor window:
  - Remote jobnet

- Receive MQ Message job
- Receive MSMQ Message job
- Send MQ Message job
- Send MSMQ Message job
- Support for Windows 10 has been added.
- The following OS is no longer supported:
  - Windows Server 2008 (excluding Windows Server 2008 R2)
  - Windows Vista
  - Windows Server 2003
  - Windows Server 2003(x64)
  - Windows XP Professional

## B.4 Revisions in 10-50

The following lists the revisions in 10-50 for each program.

### (1) JP1/AJS3 - Manager

- A virtual machine on which JP1/AJS3 has been installed and configured can now be duplicated.
- Functionality was expanded so that a disaster recovery environment can be set up with the main and remote hosts whose logical host names are the same.
- A setting for shifting the start day by a number of days (counting both open and closed days) was added.
- A function that holds jobnet execution during immediate execution registration was added.
- A function that can execute some commands from JP1/AJS3 - View to JP1/AJS3 - Manager was added.
- The files `jajs_log.bat` and `jajs_log`, which have the same functionality as the data collection tool (`_04.bat` and `_04`), are now available. In addition, an option was added to the Windows version of the data collection tools that allows you to specify the location to which data is output.
- An option was added to the `ajsp rint` command so that relation line information can be output in order of the unit name.
- The procedure for changing the IP address of an agent host was changed.
- UTF-8 was added to the list of character encodings that can be used in AIX, HP-UX, and Solaris 10.
- The number of characters that can be used when specifying a logical host name for the command `jajs_killall.cluster` was increased.

### (2) JP1/AJS3 - Agent

- The files `jajs_log.bat` and `jajs_log`, which have the same functionality as the data collection tool (`_04.bat` and `_04`), are now available. In addition, an option was added to the Windows version of the data collection tools that allows you to specify the location to which data is output.
- The procedure for changing the IP address of an agent host was changed.
- UTF-8 was added to the list of character encodings that can be used in AIX, HP-UX, and Solaris 10.



- The number of characters that can be used when specifying a logical host name for the command `jajs_killall.cluster` was increased.

### (3) JP1/AJS3 - View

- A setting for shifting the start day by a number of days (counting both open and closed days) was added to the **Schedule by days from start** settings in the **Advanced** tab of the Schedule Rule dialog box.
- A setting for holding jobnet execution if the registration method is **Immediate execution** was added to the Register for Execution dialog box.
- A function that can execute some commands from JP1/AJS3 - View to JP1/AJS3 - Manager was added.
- The files `jajs_log.bat` and `jajs_log`, which have the same functionality as the data collection tool (`_04.bat` and `_04`), are now available. In addition, an option was added to the Windows version of the data collection tools that allows you to specify the location to which data is output.

## B.5 Revisions in 10-10

The following lists the revisions in 10-10 for each program.

### (1) JP1/AJS3 - Manager

- A function was added that uses IP addresses to restrict the hosts that are able to connect to a manager host.
- A function was added for checking the execution schedule of unregistered units in the Monthly Schedule window of JP1/AJS3 - View.
- A function was added for preventing scheduler services for which the JP1 user logged in to JP1/AJS3 - View has no access privileges from appearing in JP1/AJS3 - View.
- A function was added for restricting the maximum number of allowed JP1/AJS3 - View concurrent sessions for a scheduler service.
- A function was added for changing the passwords of JP1 users by using JP1/AJS3 - View.
- The following OS is supported:
  - Linux 5.1 (x86) or later
  - Linux 5.1 (AMD/Intel 64) or later
- Content related to Solaris 11 (SPARC) was added.

### (2) JP1/AJS3 - Agent

- A function was added that uses IP addresses to restrict the hosts that are able to connect to an agent host.
- Content related to Solaris 11 (SPARC) was added.

### (3) JP1/AJS3 - View

- A function was added for checking the execution schedule of unregistered units in the Monthly Schedule window of JP1/AJS3 - View.
- A function was added for preventing scheduler services for which the JP1 user logged in to JP1/AJS3 - View has no access privileges from appearing in JP1/AJS3 - View.

- A function was added for restricting the maximum number of allowed JP1/AJS3 - View concurrent sessions for a scheduler service.
- A function was added for changing the passwords of JP1 users by using JP1/AJS3 - View.
- The way in which the Register Custom Job dialog box and the Set Properties of Custom Job dialog box are displayed was changed.
- A change was made so that when **Jobnet Definition** is selected from **Function Menu**, jobnets for which execution has been registered are displayed in the list area of the JP1/AJS3 - View window (main window).
- The function for hiding the icons of unused units in the Jobnet Editor window (a setting in the Set Default Values dialog box) was extended.
- A function was added for preventing the history of previously-used login-user names and connection-destination host names from appearing on the Login screen.

## B.6 Revisions in 10-00

The following lists the revisions in 10-00 for each program.

### (1) JP1/AJS3 - Manager

- IPv6 addresses are now supported for communication.
- Mirroring by copying a disk is now supported to enable disaster recovery operations.
- The AJS administrator, with JP1/AJS3 access permissions that are almost equivalent to superuser permissions, can now be designated when JP1/AJS3 is installed as a new installation.
- Wait conditions can now be used as a means for controlling the execution order of units across jobnets.
- The number of definition items for which macro variables can be specified has been increased, and the passing information setting job, which uses macro variables to pass information, has been added.
- A function that assigns a created or copied unit the attributes inherited from the upper-level unit, and a function that can permanently assign (fix) an execution user to a job have been added.
- A function that prevents jobs from being executed on unauthorized execution agents has been added.
- A function that allows users to save information about temporary change operations and to re-execute temporary change operations has been added.
- A function that automatically retries execution of a job when the executable file defined for the job terminates abnormally has been added.
- The dependent job of a judgment job can now be re-executed when the judgment job terminates normally.
- Mutually exclusive conditions and range conditions can now be specified as judgment conditions for judgment jobs.
- The Monitoring Event Log Job now can monitor the types of log data and events that were added in Windows Server 2008.
- An option that can be used when the Interval Control Job is defined as a start condition has been added to forcibly assume that the start condition is satisfied immediately after the monitoring of the start condition starts.
- The email sending job now supports a function that sends emails without using Outlook.
- SNMP traps can now be issued in Windows Server 2008.
- Processes for managing the embedded database have been improved.
- The startup type of the JP1/AJS3 Database service has been changed to **Manual** so that the JP1/AJS3 Database ClusterService service will not be created.

- An option has been added that checks the permissions of JP1 users mapped to OS users that have administrator permissions based on the JP1 permission level.
- A time period can now be specified when the `aj_sentry` command is used to register a jobnet for fixed execution.
- Options that can be specified when the `jajs_setup` or `jajs_setup_cluster` command is used to set up the embedded database have been added.
- The `ajsembdbidlist` and `ajsembdbstatus` commands can now be used to display the status of the embedded database.
- The embedded database settings file is now generated when an embedded database is set up. If this file is specified when the `ajsembdbbuild` command is used to build another embedded database, the other embedded database will be set up with the settings in the file.
- *Execution user name* and *Execution time* have been added as extended attributes that can be output for JP1 events (specific information) so that JP1 events indicating job termination can be used as job operating information.
- JP1/AJS3 - Manager can now link with JP1/DH - AJE, and DHAJEW has been added as a standard custom job.
- The following OS is no longer supported:
  - Solaris 9(SPARC)
  - Linux 5 (x86)
  - Linux 5 (AMD/Intel 64)
  - Linux 5 (IPF)
- The following OS is supported:
  - Windows Server 2012
  - Solaris 11(SPARC)
  - Linux 6 (x86)
  - Linux 6 (x64)

## (2) JP1/AJS3 - Agent

- IPv6 addresses are now supported for communication.
- The AJS administrator, with JP1/AJS3 access permissions that are almost equivalent to superuser permissions, can now be designated when JP1/AJS3 is installed as a new installation.
- An option that can be used when the Interval Control Job is defined as a start condition has been added to forcibly assume that the condition is satisfied when the monitoring of the start condition starts.
- The email sending job now supports a function that sends emails without using Outlook.
- SNMP traps can now be issued in Windows Server 2008.
- The Monitoring Event Log Job now can monitor the types of log data and events that were added in Windows Server 2008.
- The following OS is no longer supported:
  - Solaris 9(SPARC)
  - Linux 5 (x86)
  - Linux 5 (AMD/Intel 64)
  - Linux 5 (IPF)
- The following OS is supported:

- Windows Server 2012
- Solaris 11(SPARC)
- Linux 6 (x86)
- Linux 6 (x64)

### (3) JP1/AJS3 - View

- The Wait Conditions Settings List window and the Wait Conditions Statuses window have been added as means for managing units with wait conditions and units whose end is being waited for.
- A dialog box which lists temporary change operations that have already been performed and which allows users to re-execute (re-apply) specific temporary change operations has been added.
- Retry information can now be displayed in the following windows:
  - Daily Schedule window
  - Monthly Schedule window
  - Jobnet Monitor window
  - Detailed Schedule dialog box
  - Monitor Details - [*icon-name*] dialog box
- Whether retries are enabled is now displayed in the list area of the Jobnet Editor window.
- The Jobnet Monitor window was provided with a function that grays out all relation lines and units other than the day's units (the units that were executed and are to be executed on the day).
- **Specify period**, **Reference calendar**, **Waiting-target**, and **Execution type** have been added as search conditions that can be specified in the Search window.
- In the Search window, **Retry Settings** and **Retry execution** can now be specified as search conditions. In addition, the search results displayed in the Search window now include items related to automatic retry.
- The procedure for displaying the units found in the Search window in the Summary Monitor window has been simplified.
- The Display Item Setup dialog box can now be used to set the items to be displayed in the list areas of the Jobnet Editor window and the Jobnet Monitor window.
- A function that allows users to define all-at-one-time schedule rules that start execution at regular intervals, and a function that allows users to delete multiple schedule rules at one time have been added.
- Definition items related to automatic retry have been added to the detailed definitions of Unix jobs, PC jobs, QUEUE jobs, and custom jobs.
- Mutually exclusive conditions and range conditions can now be specified as judgment conditions for judgment jobs.
- The types of log data and events that were added in Windows Server 2008 can now be specified as **Log type** and **Event type** in the detailed definition of the Monitoring Event Log Job.
- The **Expire right after starting** option has been added in the detailed definition of the Interval Control Job. When the job has been defined as a start condition, this option can be used to forcibly assume that the condition is satisfied immediately after the monitoring of the start condition starts.
- An option ensuring that the job groups and planning groups displayed in the list area are initially selected when the JP1/AJS3 - View window (Main window) opens has been added. In addition, an option ensuring that the nested jobnets and nested remote jobnets displayed in the map area are initially selected when the Jobnet Editor window or the Jobnet Monitor window opens has been added.

- An option has been added that displays a nested jobnet as being selected at the location at which the jobnet existed before it was moved to the upper layer in the Jobnet Editor window or the Jobnet Monitor window.
- When the **Depends on upper-level jobnet** check box is selected in the schedule settings of a nested jobnet, whether to display a confirmation message before defined schedule rules are deleted can now be selected by using an option.
- The **Start** button, which allows users to launch arbitrary programs, has been added to the Monitor Details - [*custom-job-name*] dialog box.
- Because of the user authentication function added in JP1/AJS3 - Definition Assistant 09-10, specifying the JP1 user name is now mandatory for starting JP1/AJS3 - Definition Assistant from JP1/AJS3 - View. With this specification change, **User name** has been added as an item that can be selected from the **Replace** drop-down list in the Tool Entry dialog box.
- The following OS is supported:
  - Windows 8.1
  - Windows 8
  - Windows Server 2012
- An option for preventing the opening of multiple Jobnet Editor windows and Jobnet Monitor windows has been added.

## B.7 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 (jpm) and event/action control manager process (jpac) 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 `ajsshows` command for outputting the standard output file name.
- The Manager Environment Settings dialog box is no longer provided. Instead, you can use the `jaajs_config` command to set up the manager environment.
- The following environment setting parameters have been added:
  - `FixedHostnameForAgent`
  - `NotificationConstantRetry`
  - `NotificationRetryInterval`
  - `NotificationRetryCount`
  - `ClientConnectTimeout`
- 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 `ajsslogprint` 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 `jaajs_config` command to set up the agent environment.
- Linkage with a message queue system is no longer supported.
- The following environment setting parameters have been added:
  - `FixedHostnameForAgent`
  - `NotificationConstantRetry`
  - `NotificationRetryInterval`
  - `NotificationRetryCount`
  - `ClientConnectTimeout`
- 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.



## C. Reference Material for This Manual

---

This appendix provides reference information, including various conventions, for this manual.

### C.1 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:

- *JP1 Version 11 Job Management: Getting Started (Job Scheduler)* (3021-3-B11(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 System Design (Configuration) Guide* (3021-3-B13(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 System Design (Work Tasks) Guide* (3021-3-B14(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 Configuration Guide* (3021-3-B15(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 Administration Guide* (3021-3-B16(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 Troubleshooting* (3021-3-B17(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 Operator's Guide* (3021-3-B18(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 Command Reference* (3021-3-B19(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 Linkage Guide* (3021-3-B20(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 Messages* (3021-3-B21(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 - Print Option* (3021-3-B26(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 - Definition Assistant* (3021-3-B25(E))
- *JP1 Version 11 JP1/Automatic Job Management System 3 for Enterprise Applications Description, User's Guide and Reference* (3021-3-B27(E))
- *Job Management Partner 1 Version 10 Job Management Partner 1/Automatic Job Management System 3 - Web Operation Assistant Description, Operator's Guide and Reference* (3020-3-S18(E))

About JP1:

- *JP1 Version 11 JP1/Base User's Guide* (3021-3-A01(E))
- *JP1 Version 11 JP1/Base Messages* (3021-3-A02(E))
- *JP1 Version 11 JP1/Base Function Reference* (3021-3-A03(E))
- *JP1 Version 11 Integrated Management: Getting Started (Integrated Console)* (3021-3-A06(E))
- *JP1 Version 11 JP1/Integrated Management - Manager Overview and System Design Guide* (3021-3-A07(E))
- *JP1 Version 11 JP1/Integrated Management - Manager Configuration Guide* (3021-3-A08(E))
- *JP1 Version 11 JP1/Integrated Management - Manager Administration Guide* (3021-3-A09(E))
- *JP1 Version 11 JP1/Integrated Management - Manager GUI Reference* (3021-3-A10(E))
- *JP1 Version 11 JP1/Integrated Management - Manager Command and Definition File Reference* (3021-3-A11(E))
- *JP1 Version 11 JP1/Integrated Management - Manager Messages* (3021-3-A12(E))
- *JP1 Version 11 JP1/Performance Management Planning and Configuration Guide* (3021-3-A37(E))
- *JP1 Version 11 JP1/Performance Management User's Guide* (3021-3-A38(E))

- *JP1 Version 11 JP1/Performance Management Reference* (3021-3-A39(E))
- *JP1 Version 11 Job Management: Getting Started (Scripting Language)* (3021-3-B31(E))
- *JP1 Version 11 JP1/Advanced Shell* (3021-3-B32(E))
- *Job Management Partner 1 Version 10 Job Management Partner 1/Script Description and Reference* (3021-3-135(E)), for Windows Systems
- *JP1 Version 11 JP1/File Transmission Server/FTP Description, Reference, and Operator's Guide* (3021-3-B36(E)), for Windows systems
- *JP1 Version 11 JP1/File Transmission Server/FTP Description, Reference, and Operator's Guide* (3021-3-B37(E)), for UNIX systems
- *JP1 Version 11 Job Management: Getting Started (High-speed Transfer of Huge Amounts of File)* (3021-3-B41(E))
- *JP1 Version 11 JP1/Data Highway - Automatic Job Executor Operation manual* (3021-3-B46(E))
- *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/Consolidated Management 2/Extensible SNMP Agent Description, Operator's Guide and Reference* (3020-3-L04(E)), for UNIX systems
- *Job Management Partner 1/NQSEXEC System Administrator's Guide* (3020-3-F30(E))
- *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))

## C.2 Conventions: Abbreviations for product names

This manual uses the following abbreviations for product names:

Abbreviation		Full name or meaning
JP1/AJS3	JP1/AJS3 - Manager	JP1/Automatic Job Management System 3 - Manager
	JP1/AJS3 - Agent	JP1/Automatic Job Management System 3 - Agent
	JP1/AJS3 - View	JP1/Automatic Job Management System 3 - View
JP1/AJS3 - Web Console		JP1/Automatic Job Management System 3 - Web Console
JP1/AJS3 - Print Option	JP1/AJS3 - Print Option Manager	JP1/Automatic Job Management System 3 - Print Option Manager
	JP1/AJS3 - Print Option	JP1/Automatic Job Management System 3 - Print Option
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/AJS3 - Definition Assistant		JP1/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		JP1/Automatic Job Management System 3 for Enterprise Applications
JP1/DH - AJE		JP1/Data Highway - Automatic Job Executor
JP1/DH - Server		JP1/Data Highway - Server
JP1/FTP		JP1/File Transmission Server/FTP
JP1/IM	JP1/IM - Manager	JP1/Integrated Management - Manager
	JP1/IM - View	JP1/Integrated Management - View
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/OJE for VOS3		JP1/Open Job Entry for VOS3
JP1/PFM	PFM - Manager	JP1/Performance Management - Manager
	PFM - Base	JP1/Performance Management - Base
	PFM - Web Console	JP1/Performance Management - Web Console
	PFM - Agent Option for Platform	JP1/Performance Management - Agent Option for Platform(For UNIX Systems)
JP1/Performance Management - Agent Option for Platform(For Windows Systems)		
JP1/SES		Job Management Partner 1/System Event Service
NNM	HP NNM	HP Network Node Manager Software version 7.5 or earlier

Abbreviation		Full name or meaning
NNM	HP NNM	HP Network Node Manager Software Starter Edition version 7.5 or earlier
AIX		AIX V6.1
		AIX V7.1
Firefox	Firefox ESR	Mozilla(R) Firefox(R) Extended Support Release
HP-UX	HP-UX (IPF)	HP-UX 11i V3(IPF)
Linux		CentOS 6(x64)
		CentOS 7
		Linux 6.1(x64)
		Linux 7.1
		Oracle Linux 6(x64)
		Oracle Linux 7
		SUSE Linux 12
SAP BW		SAP Business Information Warehouse
SAP R/3		SAP R/3(R)
Solaris		Solaris 10(SPARC)
		Solaris 11(SPARC)

- In this manual, *JP1/AJS* is sometimes used generically, referring to JP1/AJS3 and JP1/AJS2.
- *UNIX* is sometimes used generically, referring to HP-UX, Solaris, AIX and Linux.

### C.3 Conventions: Acronyms

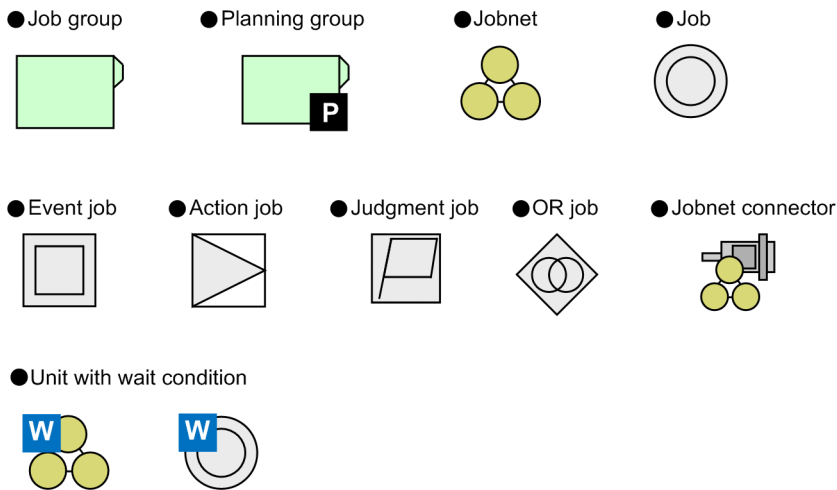
This manual also uses the following acronyms:

Acronym	Full name or meaning
ACL	Access Control List
API	Application Programming Interface
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
HTTP	Hyper Text Transfer Protocol

Acronym	Full name or meaning
HTTPS	Hyper Text Transfer Protocol Security
IME	Input Method Editor
IPF	Itanium(R) Processor Family
ISAM	Indexed Sequential Access Method
LAN	Local Area Network
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
PDF	Portable Document Format
RDB	Relational Database
REST	Representational State Transfer
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management 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

## C.4 Conventions: Diagrams

This manual uses the following conventions in diagrams:



## C.5 Conventions: Fonts and symbols

The following table explains the text formatting conventions used in this manual:

Text formatting	Convention
<b>Bold</b>	<p>Bold characters indicate text in a window, other than the window title. Such text includes menus, menu options, buttons, radio box options, or explanatory labels. For example:</p> <ul style="list-style-type: none"> <li>From the <b>File</b> menu, choose <b>Open</b>.</li> <li>Click the <b>Cancel</b> button.</li> <li>In the <b>Enter name</b> entry box, type your name.</li> </ul>
<i>Italic</i>	<p>Italic characters indicate a placeholder for some actual text to be provided by the user or system. For example:</p> <ul style="list-style-type: none"> <li>Write the command as follows: <code>copy source-file target-file</code></li> <li>The following message appears: A file was not found. (file = <i>file-name</i>)</li> </ul> <p>Italic characters are also used for emphasis. For example:</p> <ul style="list-style-type: none"> <li>Do <i>not</i> delete the configuration file.</li> </ul>
Monospace	<p>Monospace characters indicate 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"> <li>At the prompt, enter <code>dir</code>.</li> <li>Use the <code>send</code> command to send mail.</li> <li>The following message is displayed: <code>The password is incorrect.</code></li> </ul>

The following table explains the symbols used in this manual:

Symbol	Convention
	<p>In syntax explanations, a vertical bar separates multiple items, and has the meaning of OR. For example: <code>A B C</code> means A, or B, or C.</p>
{ }	<p>In syntax explanations, curly brackets indicate that only one of the enclosed items is to be selected. For example: <code>{A B C}</code> means only one of A, or B, or C.</p>

Symbol	Convention
[ ]	In syntax explanations, square brackets indicate that the enclosed item or items are optional. For example: [A] means that you can specify A or nothing. [B   C] means that you can specify B, or C, or nothing.
...	In coding, an ellipsis (...) indicates that one or more lines of coding have been omitted. In syntax explanations, an ellipsis indicates that the immediately preceding item can be repeated as many times as necessary. For example: A, B, B, ... means that, after you specify A, B, you can specify B as many times as necessary.
x	Multiplication sign
/	Division sign
↑↑	The calculation result is rounded up to the next whole number. Example: The result of ↑ 34 / 3↑ 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.
(( (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.

## C.6 Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.
- 1 MB (megabyte) is 1,024<sup>2</sup> bytes.
- 1 GB (gigabyte) is 1,024<sup>3</sup> bytes.
- 1 TB (terabyte) is 1,024<sup>4</sup> bytes.

## C.7 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.

## C.8 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*.

## C.9 About administrator privileges

In this manual, the term "administrator privileges" refers to the execution privileges of the built-in administrator account for a local PC.

If a user has administrator privileges for the local PC, operations do not vary in local user, domain user, and Active Directory environments.

## C.10 About the Start menu in Windows

If you are using Windows Server 2016, Windows 8.1, Windows 8, or Windows Server 2012, instead of using the **Start** menu, perform the operations as described below.

In Windows Server 2016, Windows 8.1 or Windows Server 2012 R2:

Click the down-pointing arrow icon to display the Apps screen.

In Windows 8 or Windows Server 2012 (except Windows Server 2012 R2):

Right-click on the Start screen to display the All Apps screen.



## C.11 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 (x86)\HITACHI\JP1AJS2`

The default installation folders of JP1/AJS3 Web Console are as follows:

`system-drive\Program Files\HITACHI\JP1AJS3WEB`

Default installation folder of JP1/AJS3 - Agent:

`system-drive\Program Files (x86)\HITACHI\JP1AJS2`

Default installation folder of JP1/AJS3 - View:

`system-drive\Program Files (x86)#\HITACHI\JP1AJS2V`

#

For 32-bit versions of Windows, replace `Program Files (x86)` with `Program Files`.

## C.12 Online manual

JP1/AJS3 - View comes with an online manual that you can read in browsers.

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.

## C.13 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 *JP1/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 *JP1/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.

## **C.14 About the 3rd level and 4th level Kanji characters added in JIS 2004 (JIS X 0213:2004)**

JP1/AJS3 cannot use the 3rd level and 4th level Kanji characters added in JIS 2004. If these characters are used, operation might not be performed correctly.

## **C.15 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

## 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 *JP1/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.

### AJS administrator

A user that has access permissions almost equivalent to superuser permissions for JP1/AJS3. Only one of the JP1/Base administrators in the JP1 administrators group can be designated as the AJS administrator. An ordinary user designated as the AJS administrator can perform such JP1/AJS3 system management operations as specifying the environment settings and starting and stopping services.

### 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.

### automatic retry

A function that automatically retries execution of a job if the executable file defined for the job terminates abnormally. Because jobs that have terminated abnormally due to a temporary error might be able to run normally when automatically retried, this function can improve the system availability.

### 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.

## broadcast agent

An agent that broadcasts a job to multiple destination agents for broadcast execution. The broadcast agent automatically manages the destination agents to which a job is to be broadcast. For example, in an environment in which the number of execution agents is increased or decreased by automatic scaling, the broadcast agent automatically expands or reduces the job distribution range. Therefore, in such an environment, you do not need to modify the job definition, or add or delete jobs manually.

## broadcast execution

To broadcast a flexible job to multiple execution agents where the flexible job is to be executed. Use a broadcast execution if you want multiple execution agents to run the same processing.

## 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.

### **data collection tool**

These tools are provided by JP1 for batch collection of log data and other information required to investigate problems. In addition to problems in JP1/AJS3, problems might occur in the OS or user programs or as the result of operational mistakes. The data collection tools collect JP1/AJS3 log data and other information, such as OS log data, at the same time.

### **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.

### **destination agent**

An agent that executes a flexible job. A destination agent might be an execution agent in an auto-scaling environment.

### **disaster recovery**

Disaster recovery refers to measures that are taken when there are unexpected occurrences such as system shutdown due to a large-scale disaster.

JP1/AJS3 supports a disaster recovery operation that copies JP1/AJS3 data on the shared disk to a shared disk at a remote site. If the JP1/AJS3 system fails because of an event such as a disaster, the JP1/AJS3 system at the remote site can continue operation by using the copied data. The disk copy and mirroring functionality of hardware is used to copy data between shared disks.

### **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 agent restriction**

A function that suppresses execution of jobs on an execution agent that the administrator has not permitted to execute the jobs. The execution agents permitted to execute jobs can be set for each unit.

When the execution agent profile is enabled, this function checks for whether the execution agent is permitted to execute jobs at the following times:

- When a unit is defined in JP1/AJS3 - View
- When the `ajschkdef` command is used to conduct a definition pre-check
- When a job is executed

### **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.

### **flexible job**

A standard job that executes processing on an agent host that is not managed directly by the manager host.

Because flexible jobs allow you to execute jobs in an environment where processing is distributed by a load balancer, you can use them to execute processing via an execution agent in a cloud environment (auto-scaling environment).

### **HP NNM**

A suite of integrated network management tools from Hewlett-Packard Co. for managing network configuration, performance, and failures.

### **HTTP connection job**

A job that uses the HTTP protocol to send requests and receive responses. By using a HTTP connection job, you can request processing by an API with an HTTP interface, such as the REST API. This allows you to integrate a business system based on JP1/AJS3 with other web-based business systems.

### **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 submit jobs are used, the ISAM database and environment setting parameters are used as the job execution environment for the QUEUE jobs and submit 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 - Agent Minimal Edition

A program that can execute a maximum of 10 jobs per day by request from JP1/AJS3 - Manager. JP1/AJS3 - Agent Minimal Edition has the same functionality as JP1/AJS3 - Agent, except for some limitations. JP1/AJS3 - Agent Minimal Edition is suitable for use on a computer that executes only a few jobs per day.

## 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 - Print Option

A component of JP1/AJS3 - Manager.

This program allows you to display or print jobnet or schedule information formatted as needed (for example, as jobnet definition information, an execution schedule table, or an execution result confirmation table).

You can also output jobnet or schedule information in CSV format so that you can edit the information.

## 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/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/DH - AJE**

A program that automatically sends data to and receives data from JP1/DH - Server.

## **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/PFM**

A program that collects operating information from numerous monitoring targets throughout the system for more stable system operation.

## **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 unit 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 used to pass information from one job to another job during execution.

A macro variable name and passing information (or for an event job, a passing information name) are specified during unit definition. A macro variable can be used during job execution by specifying the name of the macro variable in the succeeding job.

### **macro variable name**

A character string in ?AJS2xxxx? format that indicates the name of a macro variable. If a macro variable name is specified during the definition of a unit, the macro variable set in the preceding job can be used during job execution.

### **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.

**passing information**

Values specified for macro variables during registration for execution, event IDs, event issuance dates, and other information (values) that is dynamically set in macro variables and passed to succeeding jobs.

### passing information name

A variable name indicating the event information to be received by an event job.

Variable names that can be used for this purpose include `EVID` (event ID) and `EVDATE` (event issuance date). These variable names are specified during the definition of an event job.

### 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 JP1/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.

### **relay agent**

An agent that relays the execution requests associated with a flexible job. The execution requests relayed by the relay agent are executed by the destination agent of the flexible job.

By using relay agents, you can execute jobs defined for local execution in a cloud environment.

### **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.

**submit job**

A standard job registered using the `jpqjobs` command.

If descriptions in this manual do not explicitly distinguish between QUEUE jobs and submit jobs, descriptions for submit jobs also apply to QUEUE jobs.

**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.

**unit whose end is being waited for**

A unit specified as a wait condition of another unit. The unit for which the unit whose end is being waited for is specified as a wait condition starts when the unit whose end is being waited for terminates.

**unit with wait condition**

A unit for which another unit (unit whose end is being waited for) is specified as a wait condition. The unit with the wait condition starts execution when the associated unit (unit whose end is being waited for) terminates.

**virtual machine**

A virtual computer system (host) created on a physical host to run an OS is called a *virtual machine*. Multiple virtual machines can operate on a single physical host to execute multiple processes simultaneously or to run different OSs concurrently.

By replicating a virtual machine that has already been set up, you can easily increase hosts that have the same environment.

**wait condition**

A condition set for a unit that allows the execution order of units across jobnets to be controlled. A unit with another unit specified as a wait condition starts execution when the associated unit terminates.

**warning threshold**

A value for evaluating whether a job ended with a warning.

**Web Console server**

A host on which JP1/AJS3 - Web Console is installed.



A Web Console server operates as a web server and a web application server, and controls HTTP communication and the behavior of web applications. It also acquires unit information from JP1/AJS3 - Manager in response to client requests, and sends the information as a response to the client.

The Web Console server also provides an API that can be used to perform operations in JP1/AJS3. You can use this API to develop custom applications that interact with JP1/AJS3.

## Web GUI

An interface used on a client terminal to monitor units defined on the manager host. The Web GUI is displayed in a web browser, and connects to a Web Console server from a client terminal.

You can use the Web GUI to perform various tasks over the Internet, such as monitoring the execution status and progress of units and executing jobnets.

## 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 76
- 48-hour schedule 76, 82

## A

- abbreviations for products 370
- abnormal end 379
- abnormal threshold 379
- about administrator privileges 376
- About the 3rd level and 4th level Kanji characters added in JIS 2004 (JIS X 0213:2004) 378
- About the Start menu in Windows 376
- absolute day 79
- absolute time 80
- access permission 288
- access permission to executable files and script files 240
- acronyms 372
- action 33
- action job 55, 379
- agent 343
  - configuration 342
- agent host 25, 379
- agent management control 33
- AJS administrator 292, 379
- AJSPATH 379
- AND condition 96
- API control 34
- authentication server 287
- automatic distribution 314
- automatic retry 379
- automation procedures 44

## B

- backup box 379
- backup file 379
- base day 73, 380
- base time 73, 102, 380
- Being held 252
- Benefits of deploying JP1/AJS3 29
- broadcast agent 380
- broadcast execution 380
- Bypassed 252

## C

- calendar information 380
- changing the execution schedule for nested jobnet 168
- checking suspension status 213
- closed day 80, 380
- cluster system 380
- commands
  - specifying generation 189
- common user profile 380
- concurrent execution 93, 233
- concurrent execution of execution generations 103
- connection source restriction 297
- conventions
  - abbreviations for products 370
  - acronyms 372
  - diagrams 373
  - fonts and symbols 374
  - KB, MB, GB, and TB 376
  - meaning of directory and folder 376
  - version numbers 376
- correlation ID 381
- custom job 57, 381

## D

- Daily Schedule window 381
  - checking schedules 156
  - monitoring 279
- database administrator 382
- data collection tool 381
- default queue 381
- defaults
  - installation folders of JP1/AJS3 for Windows 377
- Defining a start condition 96
- delay monitor setting
  - temporarily changing 177
- dependent job 52, 381
- dependent jobnet 381
- destination agent 381
- diagram conventions 373
- directory
  - term defined 376
- disaster recovery 381
- distributing jobs among execution hosts (agent hosts) 247, 345

dummy schedule 158, 159

## E

- embedded database
  - administrator, glossary definition 382
  - glossary definition 382
  - operation commands, glossary definition 382
  - service, glossary definition 382
  - system administrator, glossary definition 382
- embedded database control 34
- End delay 256
- end delay monitoring 49
- Ended abnormally 253
- Ended normally 253
- Ended with warning 253
- end with warning 382
- environment setting parameters
  - glossary definition 382
- environment settings file 382
- ERP systems 312
- event 33, 382
- event/action control 33
- event job 54, 382
- examples of job flow 67
- exclusive control 346
- Exclusive schedule 90
- execution agent 230, 383
  - default 236
  - restriction 239
- execution agent group 234, 383
- execution agent profile 239
- execution agent restriction 239, 383
- execution ID 140, 383
- execution-locked resource 346
- execution-locked resource 383
- execution order control 32
- execution registration numbers 191
- execution schedule
  - adding to jobnet 166
- execution schedule management 33
- execution service 49
- execution simulation 160
- execution start time
  - temporarily changing 167

## F

- Failed to start 255
- features 23
- file access permission check (UNIX only) 241
- file transfer 313
- fixed execution 130
- fixed execution registration 383
- fixed schedule 383
- fixed schedule period 130
- flexible job 50, 383
- flexible job execution control 35
- folder
  - term defined 376
- font conventions 374
- Functions 30
- future generation 131

## G

- GB meaning 376
- generation 59, 140
- generation number 140
- Glossary 379

## H

- hold 49, 61
- HP NNM 312, 383
  - term defined 378
- HP NonStop Server 312
- HP-UX OS user groups 242
- HTTP connection job 58, 383

## I

- immediate execution 128
- immediate execution registration 384
- installation folders
  - default for JP1/AJS3 for Windows 377
- Interrupted 254
- Interrupted monitoring 255
- interval control job 54
- ISAM
  - compatible ISAM configuration, glossary definition 380
  - database, glossary definition 384

## J

- job 24, 48, 384
    - changing status 188
    - distributing among execution hosts (agent hosts) 247, 345
    - killing 248
    - status levels 252
    - status monitoring 244
  - job execution
    - file access permission check (UNIX only) 241
    - system 229
  - job execution control 33
    - stopping JP1/AJS3 services 249
  - job execution environment 228, 384
    - system operation 240
  - job execution environment settings file
    - specification 244
    - using 243
  - job group 62, 384
  - jobnet 24, 58
    - adding execution schedule 166
    - canceling registration 165
    - determining generation automatically 189
    - executing immediately 172
    - interrupting 179
    - killing 179
    - operations 164
    - prohibiting execution 175
    - releasing 193
    - rerunning 181
    - suspending 205
    - temporarily changing delay monitor settings 177
    - temporarily changing hold attribute 176
    - temporarily changing priority 178
  - jobnet connector 61, 384
  - jobnet creation examples 64
  - Jobnet Editor window 384
  - jobnet monitoring 61
  - Jobnet Monitor window 384
    - monitoring 278
  - jobnets
    - glossary definition 384
  - jobnets, monitoring
    - features of and differences between jobnet monitoring programs 283
    - using user application 282
    - using web browser 281
  - jobnet with start condition
    - managing saved generations 144
  - jobnet with start conditions
    - suppressing further executions after abnormal termination 110
  - job network element 384
  - job network management hierarchy 48
  - job transfer
    - restricting 238
  - JP1/AJS3 - Agent 36
  - JP1/AJS3 - Agent Minimal Edition 385
  - JP1/AJS3 - Definition Assistant 312, 385
  - JP1/AJS3 for Enterprise Applications 312, 385
  - JP1/AJS3 - Manager 36
    - database 37
  - JP1/AJS3 - Print Option 385
  - JP1/AJS3 - Print Option control 35
  - JP1/AJS3 - View 36
    - displaying hold plan 257
    - priorities of status level 265
  - JP1/AJS3 - View windows
    - monitoring 259
  - JP1/AJS3 - Web Console 36
  - JP1/AJS3 - Web Operation Assistant 312
  - JP1/Base 36, 385
    - user authentication function 287
  - JP1/Base administrator 291
  - JP1/Base user mapping 290
  - JP1/DH - AJE 315, 386
  - JP1/FTP 313, 386
  - JP1/IM 313, 386
  - JP1/NQSEXEC 313, 386
  - JP1/OJE 313
    - JP1/OJE for Midrange Computer 386
    - JP1/OJE for VOS3 386
  - JP1/PFM 315, 386
  - JP1/Script 314, 386
  - JP1/Software Distribution 314, 386
  - JP1 administrators group 291
  - JP1 event 385
  - JP1 permission level 288, 385
  - JP1 resource group 288, 385
  - JP1 user 385
  - JP1 users 288
- judgment condition 52
  - judgment job 52, 386
  - judgment value 386

## K

- KB meaning 376
- kill 386
- Killed 254
- killed processes (UNIX execution agent host) 248
- killing job 248
  - created by JP1/Script 249
- killing job or jobnet 179

## L

- list file 387
- local power control job 56
- logical host 387
- login shell
  - precautions 243
- login shell at job startup (Unix jobs, flexible jobs, HTTP connection jobs, and action jobs) 243

## M

- macro variable 387
  - specifying (during registration for execution) 136
- macro variable name 387
- mail filtering application 387
- mail receipt parameter file 387
- mail send parameter file 387
- mail system linkage 314
- mainframe 313
- manager/agent configuration 25
- manager host 25, 387
- manager job group 388
- manager jobnet 388
- managing saved generations
  - for jobnet with start condition 144
- MAPI 388
- max. shiftable days 85, 89, 388
- maximum number of concurrently executable jobs 388
- maximum shiftable days 85
- MB meaning 376
- message ID 388
- message queue system linkage 314
- MIME 388
- monitor delay 86
- Monitor-end normal 256
- monitoring event log job 54
- monitoring files job 54
- monitoring jobnets

- features of and differences between jobnet monitoring programs 283
- using user application 282
- using web browser 281
- monitoring log files job 54
- Monitor terminated 255
- Monthly Schedule window 388
  - checking schedules 157
  - monitoring 279
- multi-platform support 28
- multi-schedule 93

## N

- nested jobnet 58, 388
  - time format 77
- Nested jobnet delayed end 256
- Nested jobnet delayed start 256
- NNM linkage
  - supported products 378
- node switching system 388
- normal end 388
- Normal end + False 253
- Not executed + Ended (not scheduled) 252
- Not sched. to exe. 252
- Now monitoring 255
- Now queuing 252
- Now running 252
- number of logs to keep 142
  - overview 142
- Number of logs to keep 59

## O

- online manual
  - contents 377
- open day 79, 388
- OpenView Status Report job 56
- operating JP1/AJS3
  - by using Web GUI 301
  - by using user application 308
- operation profile
  - execution agent profile 239
- OR condition 97
- OR job 51
- OS user environment for job execution 240
- overview
  - number of logs to keep 142

## P

passing information 388  
passing information name 389  
passing information setting job 57  
PC job 50  
physical host 389  
planned execution 128  
planned execution registration 389  
planning group 63, 317, 389  
    definition and execution 324  
preceding job 49, 389  
preceding jobnet 389  
priority (jobnet) 60  
processing cycle 83, 389  
prohibiting job or jobnet execution 175

## Q

queue 342, 389  
    processing 343  
queue/agent configuration 341  
QUEUE job 50, 341  
QUEUE jobs and submit jobs  
    execution flow 343  
queueless job 347, 389  
queueless job execution control 35  
queueless job execution environment 390  
queuing job 390

## R

receive JP1 event job 54  
receive mail job 54  
receive MQ message job 54  
receive MSMQ message job 54  
recovery job 49, 390  
recovery jobnet 49, 61, 390  
recovery units 49, 61  
re-execution of a temporary change 196  
Reference Material for This Manual 369  
refer to a calendar of another job group 90  
registered day 79  
registering JP1 users 288  
registration 126  
registration for execution  
    specifying macro variable 136  
regular expressions available in JP1/AJS3 377  
Related publications 369

relative day 79  
relative time 80  
relay agent 390  
release 193  
release entry 193  
release ID 194  
release source jobnet 193  
release target jobnet 193  
releasing suspension 213  
remote power control job 56  
rerunning job or jobnet 181  
retain 107  
revisions  
    09-00 365  
    10-00 362  
    10-10 361  
    10-50 360  
    11-00 356  
    11-10 355  
    11-50 354  
root jobnet 58  
Running + Abend 252  
Running + Warning 252

## S

SAP BW systems 312  
SAP R/3 systems 312  
schedule by days from start 88, 390  
scheduled generation 140  
schedule information file 390  
schedule option 93  
schedule setting options 90  
scheduler control 32  
scheduler service 48, 390  
schedule rule 390  
    defining 78  
schedule rules 76  
schedule simulation 158  
schedule skip 93  
scheduling after releasing suspension 221  
send JP1 event job 56  
send mail job 56  
send MQ message job 56  
send MSMQ message job 56  
Settings for restricting access to units 294  
settings for restricting connection to JP1/AJS3 297  
shift days 390

- Shutdown 255
- shutdown status 391
- Skipped so not exe. 253
- SMTP 391
- specific date and time 131
- standard configuration 391
- standard error output file 243
- standard job 50
- standard output file 243
- start condition 391
- Start condition 86
- start day options 79
- Start delay 256
- start time, setting 78
- Statistical information 162
- status levels of jobnets, jobs, and jobnet connectors 252
- status monitoring of registered jobs 244
- subject 391
- submit 341, 391
- submit job 341, 391
- submit job execution control 34
- substitute schedule 391
- Substitute schedule of closed day job 84
- succeeding job 49, 391
- succeeding jobnet 391
- suspend 206, 391
- suspend function
  - activating 212
  - executing 213
- symbol conventions 374
- system management control 32

## T

- TB meaning 376
- TELstaff 314
- temporarily changing job or jobnet hold attribute 176
- temporary change
  - displaying 196
  - re-executing 196
- temporary change information 196
- threshold 392
- timeout period 49, 60, 392
- TP1/Server Base 392
- type 49, 61

## U

- umask value for standard output file and standard error output file (UNIX only) 243
- unit 392
- unit definition parameter file 392
- unit ID 392
- unit whose end is being waited for 392
- unit with wait condition 392
- Unix job 50
- Unknown end status 255
- Unmonitored + Ended 255
- user account for job execution 240
- user application 300
  - operating JP1/AJS3 by using 308
- user application (for monitoring jobnets) 282
- user name 295
- User who owns 295
- User who registered 295

## V

- valid range of start condition 99
- version number conventions 376
- version revisions 354
- View control 34
- virtual machine 392

## W

- wait condition 69, 392
- Wait for prev. to end 252
- Wait for start cond. 255
- Wait for start time 252
- Waiting to execute 252
- warning threshold 392
- web browser (for monitoring jobnets) 281
- Web Console server 300, 392
- Web GUI 281, 300, 393
  - operating JP1/AJS3 by using 301
- Web GUI control 34
- Windows Messaging 393

---

 **Hitachi, Ltd.**

6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 100-8280 Japan

---