

Fujitsu Software Systemwalker Operation Manager V17.0.1

Design Guide

UNIX/Windows(R)

J2X1-8182-06ENZ0(00)
June 2024

Preface

Purpose of This Document

This document describes how to design the Systemwalker Operation Manager as below.

- Summary of designing the system
- Designing business operations

Intended Readers

This document is intended to be used by users who design system and business using Systemwalker Operation Manager.

The reader of this document should be familiar with the general operating system and GUI operations.

Conventions Used in This Document

For descriptions and abbreviations of specific articles, refer to the "[A.1 Abbreviations and Generic Terms Used](#)".

Trademarks

For information about registered trademarks, refer to the "[A.2 Trademarks](#)".

Export Restriction

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Copyright 1995-2024 Fujitsu Limited

Contents

Chapter 1 Overview.....	1
1.1 Design Flow.....	1
Chapter 2 Overview of System Design.....	5
2.1 System Configuration.....	5
2.2 Operating Environment.....	5
2.3 Estimates.....	6
2.4 Limit Values.....	7
2.5 Environment Definitions.....	7
2.6 Measures to Address Problems.....	8
2.7 Security Measures.....	8
2.8 Maintenance.....	9
2.9 Migrating from Earlier Versions.....	9
2.10 Notes on Design.....	9
Chapter 3 Determining Power Schedules.....	11
3.1 Using the SYSTEM_CALENDAR.....	11
3.1.1 Effects of Changing the Calendar.....	11
3.2 Determining the Power Schedule.....	12
3.2.1 Determining the Power Control Mode.....	12
3.2.2 Determining the Power Schedule.....	14
3.2.3 Tips on Power Schedules.....	15
3.2.3.1 Configuration Details and Operations.....	15
3.2.3.2 Timing Server Shutdown to Smoothly Coincide with Batch Jobs Completion.....	15
Chapter 4 Determining How to Create Jobs and Job Nets.....	16
4.1 Designing Projects.....	16
4.2 Designing Jobs.....	17
4.2.1 Designing Executable Files.....	18
4.2.2 Determining Job Types.....	19
4.2.3 Determining Job Information.....	22
4.2.4 Notes on Creating Jobs.....	24
4.3 Designing Job Nets.....	25
4.3.1 Execution Attributes and Executable Jobs for Job Nets.....	25
4.3.2 Job Flow Operations.....	26
4.3.3 Basic Patterns for Job Flows.....	27
4.3.3.1 Flow Task.....	28
4.3.3.2 Conditional Branching.....	28
4.3.3.3 Consolidate-and-branch.....	30
4.3.3.4 Event Wait.....	31
4.3.3.5 Layering.....	32
4.3.3.6 Recovery Jobs.....	33
4.3.3.7 Advanced Usage.....	34
4.3.4 Determining Job Net Properties Information.....	35
4.4 Tips on How to Create Jobs and Job Nets.....	35
4.4.1 Creating Job Nets that Can Be Shared as Jobs between Different Job Nets.....	36
Chapter 5 Determining How to Start Jobs and Job Nets.....	37
5.1 Determining the Day Change Time.....	37
5.1.1 Effects of Altering the Day Change Time.....	37
5.2 Determining Holiday Calendars for Job Nets.....	38
5.3 Determining the Startup Method for Job Nets.....	39
5.3.1 Determining Execution Conditions.....	39
5.3.1.1 Waiting for Message Event Occurrence as the Execution Condition.....	42
5.3.2 Determining Startup Days.....	43
5.3.2.1 Basic Startup Day Settings.....	43

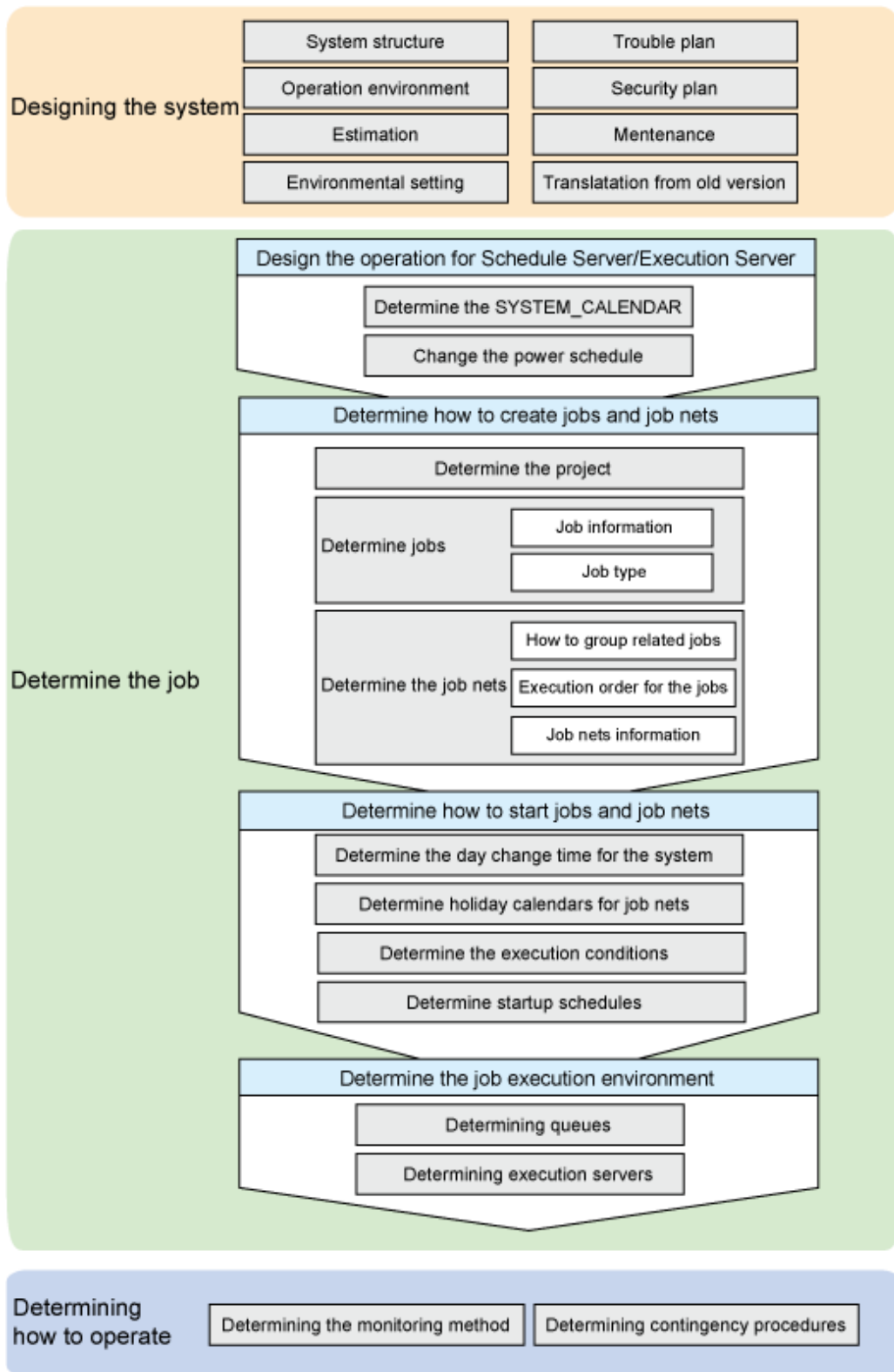
5.3.2.2 Detailed Settings for Startup Days.....	44
5.3.2.3 Exception Days.....	44
5.3.2.4 Startup Days of Layered Job Nets.....	44
5.3.3 Determining Concurrent Execution.....	45
5.4 Tips on How to Start Job Nets.....	45
5.4.1 Relationship between Startup Days and the Day Change Time.....	46
5.4.2 Executing Job Nets in Order.....	46
5.4.3 Using the Master Schedule Management Function.....	47
Chapter 6 Designing the Job Execution Environment.....	48
6.1 Queue Environment.....	48
6.2 Queue Design Examples.....	49
6.2.1 When Simultaneously Executing a Large Number of Jobs	50
6.2.2 When Executing Online Jobs.....	53
6.2.3 When Terminating Jobs that Have Been Running for a Certain Amount of Time.....	53
6.2.4 When Setting Up an Operation where Jobs Do not Have to Wait before Executing.....	53
6.3 When the Execution Server is Different than the Schedule Server.....	54
6.4 Notes for Adjusting the Job Execution Environment.....	55
6.4.1 Using "Max Execution Jobs" to Improve Response Times.....	55
6.4.2 Executing Jobs with Automatic Load Distribution.....	56
Chapter 7 Determining the Monitoring Method.....	57
7.1 Using Systemwalker Operation Manager Clients for Monitoring.....	57
7.1.1 Execution Status Monitoring.....	57
7.1.2 Delayed Monitoring.....	58
7.2 Using the Web Console for Monitoring.....	59
7.3 Using Multi-Server Monitoring.....	60
Chapter 8 Determining Contingency Procedures.....	61
8.1 Addressing Problems.....	61
8.2 Determining Backup and Recovery Methods.....	61
8.2.1 Notes on Using Backup Products [Windows].....	62
8.3 Determining Redundancy.....	62
8.4 Determining Business Continuity when Faults Occur.....	64
8.4.1 How to Continue Business Operations when the Schedule Server Fails (Enabling Continuous Execution Mode)	65
8.4.2 How to Continue Business Operations when Problems Occur with Execution Servers (Execution Server Redundancy for Network Jobs).....	65
Chapter 9 Items for Consideration.....	66
9.1 Startup Schedules for Services and Applications.....	66
9.2 Event Monitoring and Action Control [Windows].....	66
9.3 Audit Logs.....	67
Appendix A Conventions Used in This Document and Trademarks.....	68
A.1 Abbreviations and Generic Terms Used.....	68
A.2 Trademarks.....	69

Chapter 1 Overview

This chapter provides an overview of design tasks for installing Systemwalker Operation Manager.

1.1 Design Flow

The following diagram shows the flow of the basic design tasks for installing Systemwalker Operation Manager.



Designing the system

Determine how to design the Systemwalker Operation Manager system.

Determine what kind of system configuration to adopt, depending on the mode of operation. Also, consider which operating system to use, and what software and hardware is required.

This document mainly focuses on designing business operations. Consider the following reference for information on how to design the system.

Reference: "[Chapter 2 Overview of System Design](#)"

Designing business operations

Determine how to design the business operations.

Determine the operations for schedule servers and execution servers

Determine the operations for the schedule servers and the execution servers where Systemwalker Operation Manager will be installed.

To turn the power to servers on and off, and reboot the system automatically, determine the power schedule, power-up and power-down control method, and holiday settings for the SYSTEM_CALENDAR.

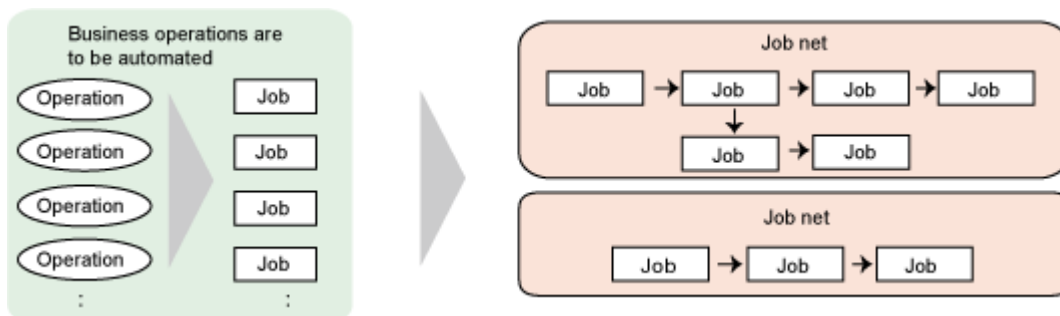
There is no need to consider these items if power schedules are not to be used.

Reference: "[Chapter 3 Determining Power Schedules](#)"

Determine how to create jobs and job nets

Identify which business operations are to be automated, and determine what kind of job to create in each case. A "job" is the smallest unit of processing.

Then, determine how to group related jobs, and decide the execution order for the jobs within these groups. The flow of related jobs is referred to as a "job net".



Reference: "[Chapter 4 Determining How to Create Jobs and Job Nets](#)"

Determine how to start jobs and job nets

Determine how to run jobs and job nets. In other words, determine the business operation schedule.

Jobs are basically run as part of job nets. For each job net, determine the startup days and execution conditions that determine the time for starting the job net.

Also, in relation to setting up startup schedules, determine the day change time for the system, and holiday calendars for job nets.

Jobs that are executed individually are managed separately as "demand jobs".

Reference: "[Chapter 5 Determining How to Start Jobs and Job Nets](#)"

Determine the job execution environment

Determine the execution environment for jobs, such as where to create the execution servers that execute jobs, and the queues where waiting jobs are managed.

Reference: "[Chapter 6 Designing the Job Execution Environment](#)"

Determining how to operate

Determine how to monitor the automated business operations when a production run has started. At the same time, also determine what kind of countermeasures to take in the unlikely event that a fault occurs.

Reference: "[Chapter 7 Determining the Monitoring Method](#)"

"[Chapter 8 Determining Contingency Procedures](#)"

Chapter 2 Overview of System Design

This chapter provides an overview of system design.

2.1 System Configuration

Determine the following items in relation to the system configuration.

Check the appropriate references as you consider these issues.

Item to be considered	Description	Reference
System configuration	Determine what kind of system configuration to adopt.	<ul style="list-style-type: none">- "Operation Configurations for Systemwalker Operation Manager" in the <i>Systemwalker Operation Manager Technical Guide</i>- "Systemwalker Operation Manager Configurations on Cluster Systems" in the <i>Systemwalker Operation Manager Cluster Setup Guide for Windows</i>- "Systemwalker Operation Manager Configurations on Cluster System" in the <i>Systemwalker Operation Manager Cluster Setup Guide for UNIX</i>- "Operation Mode of Master Schedule Management Function" in the <i>Systemwalker Operation Manager User's Guide - Master Schedule Management</i>
Network configuration	Determine the network configuration, including the communications environment and firewalls.	<ul style="list-style-type: none">- "Monitoring and Operating Jobs Across Firewalls" in the <i>Systemwalker Operation Manager Technical Guide</i>- "Monitoring and Operating Jobs from Outside NAT Environments" in the <i>Systemwalker Operation Manager Technical Guide</i>- "Operations in the IPv6 Communications Environment" in the <i>Systemwalker Operation Manager Technical Guide</i>

2.2 Operating Environment

Determine the following items in relation to the operating environment.

Check the appropriate references.

Item to be considered	Description	Reference
Operating system	Check the operating systems where servers and clients will run. Also, check functional differences between operating systems.	- "Operating Systems" in the <i>Systemwalker Operation Manager Technical Guide</i> - "List of Functional Differences for Each Operating System" in the <i>Systemwalker Operation Manager Technical Guide</i>
Required software	Check which software is required to use the various functions provided by Systemwalker Operation Manager.	- "Related Software" in the <i>Systemwalker Operation Manager Technical Guide</i> -
Required hardware	Check what hardware is required for various operations, such as performing Power Control, Event Monitoring or Task Link operations.	"Hardware Resources" in the <i>Systemwalker Operation Manager Technical Guide</i>

2.3 Estimates

Estimate the following items by checking the references below.

Item to be considered	Description	Reference
Memory usage	Estimate the memory usage required to operate Systemwalker Operation Manager.	"Hardware Resources" in the <i>Systemwalker Operation Manager Technical Guide</i>
Disk capacity	Estimate the amount of disk space required, according to the operation configuration and the scale of the operation.	"Hardware Resources" in the <i>Systemwalker Operation Manager Technical Guide</i>
Log file size	Estimate and determine the size of log files.	"Hardware Resources" in the <i>Systemwalker Operation Manager Technical Guide</i> Refer to the following for information on estimating the size of the logs: - Jobscheduler log file "Defining Startup Parameters" in the <i>Systemwalker Operation Manager Installation Guide</i> - Job execution history logs "Defining the System Operating Information" in the <i>Systemwalker Operation Manager Installation Guide</i>

Item to be considered	Description	Reference
		<ul style="list-style-type: none"> - Operation record information logs "Defining the System Operating Information" in the <i>Systemwalker Operation Manager Installation Guide</i> - Audit logs "Defining Audit Log Output in the <i>Systemwalker Operation Manager Installation Guide</i>

2.4 Limit Values

There are various limit values that apply when operating Systemwalker Operation Manager. Check these limit values when you design the operation.

Check the appropriate references.

Item to be considered	Description	Reference
Limit values for Systemwalker Operation Manager	Check the limit values in relation to the number of servers that can be monitored, the number of clients that can connect to servers, and scheduling and executing jobs.	"Limit Values" in the <i>Systemwalker Operation Manager Technical Guide</i>
Limit values for the Master Schedule Management function	Check the limit values in relation to Master Schedule Management.	"Limited Values" in the <i>Systemwalker Operation Manager User's Guide - Master Schedule Management</i>

2.5 Environment Definitions

Determine the following items in relation to the operating environment definitions.

Check the appropriate references.

Item to be considered	Description	Reference
Installation tasks	Check the installation tasks for Systemwalker Operation Manager.	"Installation of Systemwalker Operation Manager" in the <i>Systemwalker Operation Manager Installation Guide</i>
Installation tasks (for cluster operations)	Check the installation tasks for cluster systems.	<ul style="list-style-type: none"> - "Installation" in the <i>Cluster Setup Guide for Windows</i> - "Installation" in the <i>Cluster Setup Guide for UNIX</i>
Definitions relating to the system	Determine host definitions, port numbers, and other issues	- "System Definitions" in the <i>Systemwalker Operation Manager Installation Guide</i>

Item to be considered	Description	Reference
	such as tuning system parameters.	- "Listing of Port Numbers" in the <i>Systemwalker Operation Manager Installation Guide</i>
Common definitions for Systemwalker Operation Manager	Check the definition of the monitoring host and user, the definition of audit log output, and the definition of encrypted communication for the Web console.	"Common Definitions for Systemwalker Operation Manager" in the <i>Systemwalker Operation Manager Installation Guide</i>
Definitions for multi-subsystem operations	Check the environment definitions for multi-subsystem operations.	"Definition of Multi-Subsystem Operations" in the <i>Systemwalker Operation Manager Installation Guide</i>
Definitions for Power Control	Check the environment definitions for Power Control.	"Definition of Power Control" in the <i>Systemwalker Operation Manager Installation Guide</i>
Definitions for Jobscheduler	Check the environment definitions for Jobscheduler.	"Definition of Jobscheduler" in the <i>Systemwalker Operation Manager Installation Guide</i>
Definitions for Job Execution Control	Check the environment definitions for Job Execution Control.	"Definition of Job Execution Control" in the <i>Systemwalker Operation Manager Installation Guide</i>
Definitions for Event Monitoring	Check the environment definitions for Event Monitoring.	"Definition of Event Monitoring [Windows]" in the <i>Systemwalker Operation Manager Installation Guide</i>
Definitions for Task Link	Check the environment definitions for Task Link.	"Definition of Task Link" in the <i>Systemwalker Operation Manager Installation Guide</i>

2.6 Measures to Address Problems

Refer to "[Chapter 8 Determining Contingency Procedures](#)" for details on measures to address problems.

2.7 Security Measures

Determine the following items in relation to security.

Check the appropriate references.

Item to be considered	Description	Reference
Security	Determine security measures for operating Systemwalker Operation Manager securely.	- "Security" in the <i>Systemwalker Operation Manager Technical Guide</i> - "Security Definitions" in the <i>Systemwalker Operation Manager Installation Guide</i>

2.8 Maintenance

Determine the following items in relation to maintenance.

Check the appropriate references.

Item to be considered	Description	Reference
Process Monitoring	Determine Process Monitoring.	"Defining the Process Monitoring Function" in the <i>Systemwalker Operation Manager Installation Guide</i>
Collect Maintenance Information Tool	Determine how to use the Collect Maintenance Information Tool, which collects the information required to address problems.	"Defining the Collect Maintenance Information Tool" in the <i>Systemwalker Operation Manager Installation Guide</i>

2.9 Migrating from Earlier Versions

Determine the following items when migrating from an earlier version.

Check the appropriate references.

Item to be considered	Description	Reference
Migration Method	Determine migrating work to update another version level.	<i>Systemwalker Operation Manager Upgrade Guide</i>
Functional Differences between Versions and Levels	Check the functional differences between Versions and Levels.	"Functional Differences between Versions and Levels" in the <i>Systemwalker Operation Manager Release Guide</i>
Compatibilities across versions and levels	In mixed environments that include earlier versions, consider issues such as the range of connection between servers, and between servers and clients.	- "Compatibility with Earlier Versions" in the <i>Systemwalker Operation Manager Technical Guide</i> - "Consistency with the Earlier version" in the <i>Systemwalker Operation Manager User's Guide - Master Schedule Management</i>
Incompatible information	Check the incompatible information.	"Incompatible Notes" in the <i>Systemwalker Operation Manager Release Guide</i>

2.10 Notes on Design

Refer to the following manual for notes on design:

Appendix B.2, "Precautions before Operations" in the *Systemwalker Operation Manager User's Guide*:

- Notes on building environments in different operating systems
- Notes on operations in Windows Server 2008 or later [Windows]

- Notes on running large-scale jobs
- Notes on the desktop heap [Windows]
- Notes on using antivirus software
- Startup modes of job processes [Windows]
- Notes on using JIS X 0213 in a 2004 environment [Windows]
- Notes on using an EUC code environment [UNIX]

Chapter 3 Determining Power Schedules

This chapter describes how to determine when to set server holidays for the schedule server and execution server operations, as well as how to use automated controls for powering on or off, or rebooting.

Note that this chapter covers considerations for using automated controls. It is, therefore, not essential for those who do not configure server holidays or do not use automated controls for powering on or off, or rebooting.

3.1 Using the SYSTEM_CALENDAR

In the same way as a calendar contains holidays such as Sundays and public holidays, calendars can be created in Systemwalker Operation Manager in which working days and holidays can be set.

The default calendar provided when Systemwalker Operation Manager is installed is the SYSTEM_CALENDAR. Configure holidays and day change times (time when the date changes) in the SYSTEM_CALENDAR registered on the schedule server.

Systemwalker Operation Manager has only one SYSTEM_CALENDAR. When installed, its configuration is as shown below.

Day change time	00:00
Working days/ holidays	All working days

"Working days" are days when the server is powered on, while "holidays" are days when the server remains powered off. Review the SYSTEM_CALENDAR settings to change the working days (weekdays) or holidays when the server is operated.

In addition to being used for power schedules as a calendar to configure working days and holidays, the created calendar can also be used for the following:

- Schedule for services or applications that are to be started (Service/Application Execution)
- Job net schedules (Jobscheduler)
- Action execution conditions (Event Monitoring) [Windows]
- Executing calendar-based processing (Calendar API)

Note that the SYSTEM_CALENDAR can also be used to configure the day change time. Refer to ["5.1 Determining the Day Change Time"](#) for details on the day change time.

3.1.1 Effects of Changing the Calendar

If the calendar or day change time is changed after the power schedule or the job net schedule has been configured, there will be various effects, for example, the power schedule or the job net schedule that was set will change. Therefore, as much as possible, avoid changing the settings for the calendar and the day change time after the power schedule or the job net schedule has been configured.

To change calendar or day change time, set as the following procedure:

1. Set up the SYSTEM_CALENDAR (working days, holidays, and the day change time).
2. Set up the power schedule.
3. Set up the job net schedule.

In this section, we have collated the effects of changing the calendar after the power schedule or the job net schedule has been configured. For the effects of the day change time, refer to ["5.1.1 Effects of Altering the Day Change Time"](#).

Effects on Power schedule

If the date on which the power schedule has been set is changed to a holiday, that power schedule will be invalid. However, even if this date has been changed to a holiday after the server was powered on, the server will not be powered off until the next scheduled power off time. In this case, if the next day is a holiday, the day change time of the current day will be the next scheduled power off time.

After the power schedule has been set, avoid changing the day change time.

Effects on Service/Application Execution pattern

If the calendar is referenced using the Service/Application Execution pattern, the Service/Application Execution working day is set by linking with the content in the calendar that was changed.

Effects on Job net schedule

If a holiday calendar that has already been registered is changed, you can create the job net startup days that have been specified in that holiday calendar again by selecting a setting that enables this.

To do this, use the **Use function2** tab of the **Define Jobscheduler Startup Parameters** window. To create the job net startup days again, check/edit the startup days after the holiday calendar is changed.

3.2 Determining the Power Schedule

A power schedule defines the time when the server power is scheduled to be switched on or off, or rebooted.

In Systemwalker Operation Manager, the following operations can be performed based on the configured power schedule by integrating power control software:

- Power on the server
- Power off (shutdown) or reboot the server

When power is controlled, it will switch off at the time specified in the power schedule on days configured as holidays in the SYSTEM_CALENDAR. For days other than holidays (working days), power will switch on at the time specified in the power schedule (or if no time has been defined, power on/off will occur according to the day change time).

Note that the server power can be switched off (shutdown) or rebooted even when power control software is not used.

Power control modes and power schedules at the time of installation

The following operations are defined by default when Systemwalker Operation Manager is installed.

Windows

- Power is controlled for individual servers
- Only shutdown/reboot is performed (power control software is not used)

UNIX

- Power is not controlled

If any operational issues arise with the above-mentioned operations, review the power control mode or the power schedule.

3.2.1 Determining the Power Control Mode

Consider the following items when determining whether power should be controlled for Systemwalker Operation Manager servers.

Items to be considered

Power control mode

The following power control modes can be selected.

Control mode	Description
Disable the power control	Server power is not controlled.
Power is controlled for individual servers	Power is controlled for each server.
Batch Power Control [windows version]	Power for multiple servers is controlled in a single step. All hosts that execute Batch Power Control are handled as a single logical server. You can collectively specify the power control hosts that will execute Batch Power Control as well as the control target hosts for which power will be controlled. The SYSTEM_CALENDAR is then configured for the power control hosts, together with other items, such as the power control pattern and the completion monitoring options.

The following items should also be considered when controlling power:

- Power control software

Consider using power control software for automatically switching the computer power on and off.

Power control software is required to automatically power on a computer.

Note that computer power can be automatically switched off (shutdown) or rebooted even when power control software is not used. If power control software is not used, configure settings so that only shutdown or reboot is performed.

Refer to "Hardware Resources" in the *Systemwalker Operation Manager Technical Guide* for details on power control software that can be used.

- Power control hosts and control target hosts

To execute Batch Power Control, determine the power control hosts and the control target hosts. Manual server operations can only be performed for power control hosts.

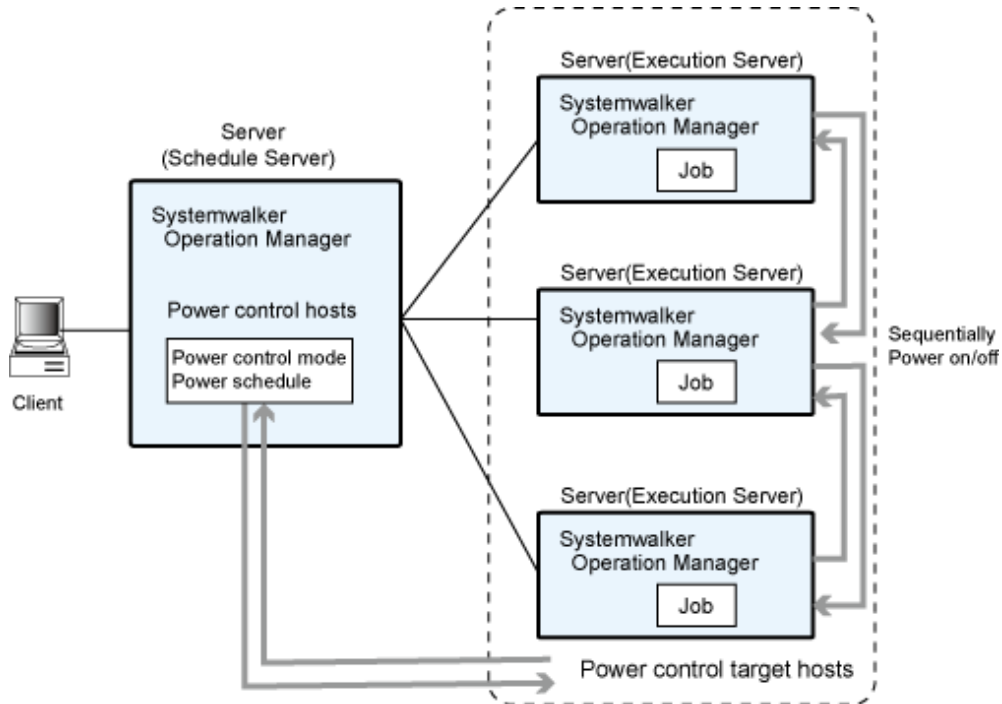
- Powering-on sequence

When executing Batch Power Control, the power control hosts are switched on first, following which the control target hosts are powered on in a set order.

The powering-off sequence is the opposite of the powering-on sequence, and the power control hosts are powered off last.

- Interval between powering on and off

When executing Batch Power Control, determine the time until the next host is started (startup interval) and the time until the next host is stopped (shutdown interval).



3.2.2 Determining the Power Schedule

In Systemwalker Operation Manager, the power schedule is integrated with power control software and the server power is automatically switched on or off, or rebooted, based on the defined schedule.

Determine a server power schedule if power is to be controlled.

Items to be considered

Power control patterns

Determine the power control pattern. There are three types of patterns, as described below.

Create a non-standard pattern or an individual pattern, as required.

Note that standard patterns must be created and applied in advance even when using a non-standard pattern or an individual pattern.

- Standard pattern

This is a general-purpose pattern that defines the power-on/off schedule and the reboot opportunities for a week. If the server operates using a standard pattern alone, the server will be powered on and off, and rebooted using the same repeated pattern every week.

This standard pattern is automatically applied to all days other than those configured as holidays in the SYSTEM_CALENDAR, and the power is switched on or off, or rebooted, accordingly.

- Non-standard patterns (patterns other than standard patterns)

This is a general-purpose pattern that defines the power-on/off schedule and the reboot opportunities for a day. Consider creating a non-standard pattern for days when you need to power on or off, or reboot, according to a pattern that differs from the standard pattern.

This pattern can be applied to multiple days other than those configured as holidays in the SYSTEM_CALENDAR.

- Individual patterns

This pattern defines the power-on/off schedule and the reboot opportunities for a particular day. An individual pattern can also be created for days set as holidays in the SYSTEM_CALENDAR, but for which the user wants to switch power on or off, or reboot.

Determining the completion monitoring option

If you want to perform completion monitoring when shutting down the server, consider the completion conditions, such as those listed below, for shutting down or rebooting the server.

- Monitor the presence of users connected via the network
- Notify users connected via the network that the server will shutdown
- Shut down the server waiting for jobs to be completed

Other completion monitoring options include the execution of commands when the work end time is reached, and issuing operator notifications if the completion conditions have not been satisfied after a certain period of time has elapsed following the work end time.

Note that shutdown will take place immediately if completion monitoring options have not been specified.

If multiple completion conditions have been specified, shutdown will occur after all the specified conditions have been satisfied.

3.2.3 Tips on Power Schedules

This section provides helpful information that you can refer to when determining power schedules.

3.2.3.1 Configuration Details and Operations

This section explains the power schedule settings and operations.

- **If the power schedule for a day is configured so that operations are performed until the day change time is reached, but the power schedule for the next day is configured so that operations commence from the day change time**

Power will not be switched on or off at the day change time. Operation will be continuous.

- **If, as a result of changing the power schedule, the current time (when modified) is changed from being within operating hours to during shutdown hours**

The server will not be disconnected at the time the change is made. The new time will be effective from the next shutdown time.

3.2.3.2 Timing Server Shutdown to Smoothly Coincide with Batch Jobs Completion

The queuing completion notification command can be used to issue a "queuing completion notification" from a fixed job operation application, such as the Jobscheduler. Create the "queuing completion notification" for a job so that it is issued when the job is completed, and define it as a completion condition in completion monitoring.

Using completion monitoring in this way enables server shutdown to smoothly coincide with the completion of batch jobs, as a result, eliminating the need for the system administrator to wait for jobs to finish. Server shutdown can also be delayed automatically if the jobs are delayed.

Refer to the *Systemwalker Operation Manager Reference Guide* for details on the queuing completion notification command.

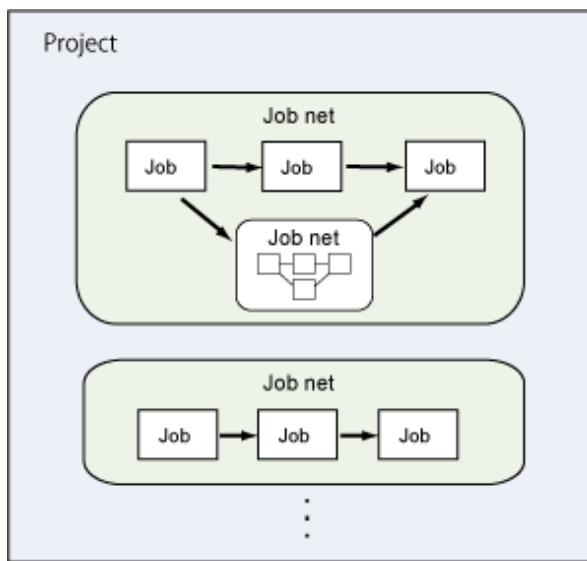
Chapter 4 Determining How to Create Jobs and Job Nets

This chapter explains how to create jobs and job nets.

Note that there is a limit to the number of projects that can be created, and the number of jobs that can be registered in a job net. Refer to "Limit Values" in the *Systemwalker Operation Manager Technical Guide* for details.

4.1 Designing Projects

A "project" is a unit for managing a collection of job nets, classified according to their business operations. A project consists of at least one job net.



Separate access rights can be set up for each project. Projects are also a basic unit for monitoring, so administration is easier if projects are organized by business categories when projects are created.

Consider the following items for projects.

Items to be considered

Project name

Ensure that project names do not conflict with the names of other projects. Administration is easier if projects are given names that represent the group of business operations to be automated or monitored.

Note that "temp" and "core" are reserved by the Jobscheduler and cannot be used as project names.

Refer to "Project Information Control Statement" in the *Systemwalker Operation Manager Reference Guide* for information on the characters that can be used in project names.

Owner

Specify the OS user as the owner of a project. The owner has update rights for projects, and can register, change, monitor, and operate job nets and jobs within projects.

For jobs where an execution user is not specified in the job information, the owner specified here becomes the execution user.

Access rights

Users and their respective access rights must be configured for a project in advance, enabling users to access projects.

The following access rights can be configured for a project.

Access rights	Description
Update Rights	Enables job nets and jobs in approved projects to be updated, registered, operated, or referenced
Registration Rights	Enables job nets and jobs in approved projects to be registered or referenced
Operation Rights	Enables job nets and jobs in approved projects to be operated or referenced
Reference Rights	Enables job nets and jobs in approved projects to be referenced only

Users that can access a project, and their respective access rights, at the time a project is created are given below.

Registered user	Access rights	Remarks
System administrators (Users belonging to the Administrators group [Windows] or superusers [UNIX])	Update Right	
Project owners	Update Right	
Operation Manager users with administrator rights	Update Right	When Extended User Management function is enabled [UNIX]

If the above settings are not appropriate, the users and groups that will be allowed to access a project, together with their respective access rights, must be configured using the **Set Permissions** window.

Specify these settings as one of the following users:

- System administrators (belonging to the Administrators group [Windows]/superusers [UNIX])
- Operation Manager users with administrator rights [Unix]

Refer to "Access Control" in the *Systemwalker Operation Manager Technical Guide* for details on access control for projects. Also, refer to "Setting up Access Permissions for Projects" in the *Systemwalker Operation Manager User Guide* for details on how to set up access control.

4.2 Designing Jobs

A "job" is the smallest processing unit for the business operations to be automated. A job involves processing business operations by using, for example, shell scripts, executable programs, or a Job Control Language (JCL) file.

Identify which business operations will be automated using Systemwalker Operation Manager, and determine, for example, the format of the job executable files that will process business operations or the job information to be defined.

4.2.1 Designing Executable Files

Systemwalker Operation Manager processes business operations using batch files or shell scripts called "jobs". Determine the format of the executable files that can be registered as jobs with Systemwalker Operation Manager.

The executable files that can be registered as jobs are given below.

Windows

- Executable programs (.exe and .com)
- Batch files (.bat and .cmd)
- JCL files (.jcl) (*1)
- Systemwalker scripts (.swt) (*1)

*1: Executable only if the execution attribute is "Job Execution Control".

The commands listed below cannot be registered as jobs because they cannot be executed using Job Execution Control. Also, exercise caution with commands that start memory-resident programs, as these commands do not terminate as jobs even when the processing for commands has terminated.

- Commands with window-based interfaces (such as Notepad, Microsoft Excel, and Microsoft Word. However, these commands can be used if they are used with the Task Link client application startup commands.)
- Interactive commands
- Commands that prompt for input (such as **format** and **backup**)
- Applications that are not processed correctly when an execution request is issued using the **AT** command (only when the **AT** command is executed without specifying the "/interactive" option and the logon accounts for the "Schedule" and "Systemwalker MpMjes" services are the same)

Moreover, internal commands provided by cmd.exe, such as **dir** and **chdir**, cannot be specified as is. To use these internal commands, they must be executed in a new command interpreter environment, such as placing them in a batch program (.bat) or using cmd.exe.

UNIX

Determine the file format for the jobs to be registered. The executable files that can be registered as jobs include:

- Executable programs
- Shell scripts
- Commands
- JCL files (.jcl) (*1)
- Systemwalker scripts (.swt) (*1)

*1: Executable only if the execution attribute is "Job Execution Control".

The commands listed below cannot be registered as jobs because they cannot be executed using Job Execution Control:

- Commands with window-based interfaces
- Interactive commands










- Commands and shell scripts that prompt for input


















4.2.2 Determining Job Types








Determine a job type according to the operation to be processed.

When defining jobs, allocate a job icon to each job. A job icon represents the type of job displayed, for example, in a job flow. Selecting a job icon makes it easier to use the executable files (commands) provided by Systemwalker Operation Manager by default, and also makes it easier to identify each job.

The following types of jobs are handled by Systemwalker Operation Manager.

Job icon	Job type (icon type)	Job description
	Execute command	Runs an executable file
	Send message	Generates a message event
	Determine file existence	Evaluates whether a certain file exists It can check the existence of a file that is dynamically created or deleted by another job or job net. It is then possible to determine which job to branch to, depending on whether the file exists.
	Determine job net variable	Evaluates the value of a job net variable It compares the value of a job net variable (set by the execution of a preceding job) with a numerical value or string, and then evaluates whether it matches a condition. It is then possible to determine which job to branch to, depending on the result of the comparison.
	Start service [Windows]	Starts a service
	Stop service [Windows]	Stops a service
	Check service status [Windows]	Evaluates whether a certain service has started or stopped It can be defined to run after the processing that starts or stops a service, to check the status of the service.
	Wait for file	Waits until a particular file is created, deleted, updated, or resized The job monitors the file at the specified monitoring interval, and terminates when the conditions are met.
	Wait for Message Event	Waits until a particular message event occurs The job monitors the message event (that it is waiting for) at the specified monitoring

Job icon	Job type (icon type)	Job description
		interval, and terminates when the message event occurs.
	Backup cooperation [Windows]	Performs a backup using the ARCserve function
	Check log file	Analyzes a text format file and evaluates whether it contains a string specified by a regular expression It is then possible to determine which job to branch to, depending on whether the file contains the string.
	Wait for time	Waits until the specified time is reached The job terminates when the specified time is reached.
	Wait until Specified Time	Waits for the specified time period, starting from the time when the job started executing The job terminates when the specified time period has elapsed.
	REST execution [Windows] [Linux]	Executes an API of a cloud service that provides a REST interface.
	Job net	Executes a child job net
	Linked job net	Executes a linked job net
	Application execution (client)	Starts an application on the client
	File compression/expansion (client)	Compresses or decompresses a file on the client
	Rename file (client)	Renames a file on the client
	Delete file (client)	Deletes a file on the client
	Create directory (client)	Creates a directory on the client
	Delete directory (client)	Deletes a directory on the client
	File transfer (client)	Transfers files between the server and the client
	Power on (client)	Turns on power to the client
	Shutdown (client)	Turns off power to the client
	Application execution (server)	Starts an application on the server

Job icon	Job type (icon type)	Job description
	File compression/expansion (server)	Compresses or decompresses a file on the server
	Rename file (server)	Renames a file on the server
	Delete file (server)	Deletes a file on the server
	Create directory (server)	Creates a directory on the server
	Delete directory (server)	Deletes a directory on the server
	File transfer (server)	Transfers a file between servers
	Interstage	Executes WorkUnits defined using Interstage Note: This type of job can only be registered in job nets with the "Interstage" attribute.

If the default job icons above are insufficient, consider adding more job icons as required. Refer to "Adding User-Defined Job Icons/Categories" in the *Systemwalker Operation Manager User Guide* for details on how to add job icons.

Job categories

Jobs are classified into categories, depending on their purpose. This section explains three categories that are useful to know when designing business operations (basic jobs, critical jobs, and event wait jobs), and also explains the jobs included in these categories.

Basic jobs

Basic jobs are jobs that are frequently used in Systemwalker Operation Manager. Basic jobs include the following:

- Execute command
- Send message
- Determine File Existence
- Determine Job Net Variable
- Start service [Windows]
- Stop service [Windows]
- Check service status [Windows]
- Wait for file
- Wait for Message Event
- Backup cooperation [Windows]
- Check log file
- Wait for time
- Wait until Specified Time
- REST execution job [Windows] [Linux]

Criteria jobs

Criteria jobs evaluate particular events based on criteria. A criteria job can be used to determine whether to start the succeeding job based on the result of evaluating certain criteria. For jobs that follow criteria jobs, define jobs that should start if the criteria are met. Criteria jobs include the following:

- Determine File Existence
- Determine Job Net Variable
- Check service status [Windows]
- Check log file

Note that you need not use criteria jobs to determine which job to branch to, depending on the completion code of the preceding job. A condition for evaluating the completion code of the preceding job can be specified in the execution condition definitions of the succeeding job.

Event wait jobs

An event wait job waits for particular conditions. Such jobs can be used as a preceding job for jobs that you do not want to start until certain conditions are met. For the succeeding job, define a job that should wait until certain conditions are met before starting. Event wait jobs include the following:

- Wait for file
- Wait for Message Event
- Wait for time
- Wait until Specified Time

4.2.3 Determining Job Information

Determine the job information that is defined for jobs.

The following table provides the items that are defined as the job information.

Item to be considered	Description
Job name	Determine the job name.
Executable file	Determine the batch file (for Windows), shell script (for UNIX), command, executable program, Systemwalker script, JCL, or any other executable file that will be started as a job. It is also possible to transfer the executable file to another server where it is to be executed.
Execution directory	Determine the current directory to be used when the job is executed.
Execution server	Determine which server will execute the job. If an execution server is not specified, the job will be executed on the schedule server.
Queue name	Determine which queue the job will be submitted to.
Upper limit of completion code	Determine the value range of completion codes used for checking whether a job terminated normally or abnormally. Completion codes from 0 to the value specified here will be regarded as "Completed".

Item to be considered	Description
Standard output/ standard error output	Determine the name of the file to be output, and whether to save the file that is output.
Child job net/ linked job net	For child job nets, specify the job net to be registered as the child job net. For linked job nets, specify the job net to be used as the master linked job net.
Job control information	<p>Determine whether to perform the following types of job control:</p> <ul style="list-style-type: none"> - Specify a restart point When the job net is restarted, processing will resume from the specified job. - Register the job as a recovery job The recovery job will be started if the preceding job terminates abnormally. - Use different execution conditions, depending on the completion code of the preceding job The start of succeeding jobs will be allocated depending on the completion code of the preceding job. - Wait using the OR condition of the preceding job Job control will wait using the OR condition of multiple preceding jobs. - Enable the startup days specified for child job nets or linked job nets - Enable the startup days specified for child job nets or linked job nets.
Execution user name	Determine the execution users for the job.
Resources	Determine whether to allocate resources, and if so, how to allocate the resources.
Transmission of execution command	Determine whether to execute the job by sending the executable file to the requested server, or by running an executable file that already exists on the requested server.
Execution priority	Determine if there are jobs that should be given priority over this job, and if so, what the priority level should be.
Estimated processing time	<p>Determine whether to monitor for job end delay. If an estimated processing time is specified, a notification message can be output to the event log or SYSLOG when the value is exceeded.</p> <p>If specifying this item together with the job cut-off execution time, ensure that the estimated processing time is less than the cut-off execution time.</p>
Cutting off jobs	Determine whether to impose a limit on the job execution time. If a limit is imposed, the job is cut off at the specified elapsed time and becomes "Abended".

Item to be considered	Description
Job net variable	Determine whether to use the job output information as a job net variable within the job net. If you decide to use this information, also determine a prefix.
Environment variables	Determine the environment variables for executing the job.

You can specify job information in the **Add/Change - Job** window. Refer to the *Systemwalker Operation Manager Online Help* for details.

4.2.4 Notes on Creating Jobs

This section explains several points to take into account when designing and creating jobs.

Appendix A, "Important Notes" in the *Systemwalker Operation Manager User's Guide* also explains points you should note when creating jobs. Refer to the following:

- "Cautions on Jobs"
- "Notes on Network Jobs"
- "When Registering a Backup Command as a Job [Windows version]"
- "When Registering the lfmt Command as a Job [UNIX version]"

Commands that cannot be executed for jobs

Design a job so that the following commands are not executed as jobs. If these commands are executed, the behavior of Systemwalker Operation Manager and Systemwalker Centric Manager cannot be guaranteed.

- Start commands for Systemwalker Centric Manager and Systemwalker Operation Manager
- Stop commands for Systemwalker Centric Manager and Systemwalker Operation Manager
- Backup commands for Systemwalker Centric Manager and Systemwalker Operation Manager (unless you specify options to prevent the service or daemon from stopping)
- Restore commands for Systemwalker Centric Manager and Systemwalker Operation Manager
- **qsub** command for Job Execution Control
- Commands that cause a cluster system to stop if Systemwalker Operation Manager has been registered on the cluster system
- Other commands that cause Systemwalker Operation Manager to start

Note that if the system includes Systemwalker Centric Manager, commands that cause Systemwalker Centric Manager to start (-a option of the **scentricmgr** command and the **swstart** command of Systemwalker Centric Manager) will also start Systemwalker Operation Manager.

Jobs that cannot be submitted as network jobs

The following jobs and commands cannot be submitted as network jobs:

- If a job is a Job Control Language (JCL) file
- If the following jobs and commands are used:
 - Wait for Message Event job
 - Determine Job Net Variable job
 - for SMS commands
 - REST execution job [Windows] [Linux]

Jobs that access other servers [Windows]

Note the job execution restrictions when registering the following types of job that will access another server:

- Jobs that access the network drive, such as jobs that copy files from another server
- Jobs that specify another server and perform message linkage

By default, the local system account is used to execute jobs from Systemwalker Operation Manager. However, there are restrictions on using the local system account to access other servers, so the jobs do not operate normally.

If a job will access another server, you must change the execution user of the job. Consider the following according to your operation:

When the same execution user executes all jobs

Change the startup account for the Systemwalker Operation Manager services. This will automatically change the account of the execution user who will execute the jobs.

Change the startup account for the following services:

- Systemwalker MpMjes service
- Systemwalker MpJmSrv service

Refer to "Changing the Startup Account" in the *Systemwalker Operation Manager Installation Guide* for details.

When different execution users execute different jobs

You must configure all the following settings in the **Systemwalker Operation Manager Environment Setup** window:

- Click **Define Operating Information >> Options >> Execute jobs under the respective job owner's authority**
- In the **Define Job Owner's Information** window, register the password for the execution user of the job

Register passwords for the Jobscheduler project owner and all users who will be execution users of the job.

You must also configure the following Windows settings:

- Click **Control Panel >> Administrative Tools >> Local Security Policy**, and grant **Log on as a batch job** rights to each job owner.

4.3 Designing Job Nets

A "job net" is a group of jobs for which an execution order is specified. Startup and stop are controlled per job net.

When creating a job net, determine the jobs to be registered in the job net and the execution order of the jobs.

4.3.1 Execution Attributes and Executable Jobs for Job Nets

The execution attributes for job nets are listed below. The executable jobs depend on the execution attribute.

Job Execution Control attribute

For job nets with this attribute, it is possible to execute all the jobs listed in "Determining Job Types" above except for Interstage WorkUnits.

However, the following differences are observed, depending on the operating system of the server where the job is executed:

- **Jobs that can only be executed with Windows versions**

- Start service, Stop service, Check service status
- Backup cooperation

Interstage attribute

Job nets with this execution attribute can only execute Interstage WorkUnits. Only one job with an Interstage icon can be registered.

PowerAIM attribute [UNIX]

Job nets with this execution attribute can only execute PowerAIM WorkUnits.

Normal attributes [UNIX]

Job nets with this execution attribute can only execute Execute command, Start service, Stop service, Send message, Wait for file, and Backup cooperation jobs.

Refer to *Systemwalker Operation Manager Technical Guide* for information on the execution attributes.

4.3.2 Job Flow Operations

This section explains basic job flow operations that you should be familiar with before creating a job flow (job execution order).

Basic job flow operations

The job flow you create is essentially operated as explained below.

- The jobs are started in the order in which they are joined by a connection line, starting with the first job.
- If the preceding job terminates normally, the succeeding job is started. If the preceding job terminates abnormally, processing is canceled.

However, if the recovery job is connected, it is started even if the preceding job terminates abnormally.

Normal and abnormal termination of jobs

The completion code of a job is used to determine whether a job is "Completed" or "Abended".

The following table lists the default completion code settings.

"Completed"	Completion code is 0
" Abended "	Completion code is 1 or higher

The range for which the job will be regarded as "Completed" can be changed.

Refer to "[Changing the "Completed" range](#)" for information on how to change the range.

Changing the "Completed" range

You may want to, for example, use the completion code to create a branching flow. By default, only jobs with a completion code of 0 are regarded as "Completed", so jobs with any other completion code are regarded as "Abended" and processing is canceled. For a branching flow, you must therefore configure the settings so that a job is not regarded as "Abended" and processing is not canceled even when the completion code is not 0.

In this case, change the range for which the job will be regarded as "Completed". You can use either of the following methods to change the "Completed" range.

Increasing the upper limit of the completion codes regarded as "Completed"

Change "Upper limit of completion code" in the job information.

After the change, jobs with completion codes from 0 to the specified upper limit will be regarded as "Completed".

Regarding all completion codes as "Completed"

Set "Ignore completion code" in the job information.

After you set this value, jobs will be regarded as "Completed" regardless of the completion code, and processing will not be canceled.

Information

"Pseudo-normal"

If you increase the upper limit of the completion codes that will be regarded as "Completed", you can treat the completion codes from 1 to the upper limit as "Pseudo-normal" to differentiate these completion codes from the original "Completed" (completion code 0). Setting "Validate pseudo-normal" causes the job status to be differentiated as indicated below.


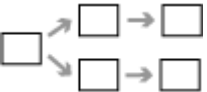
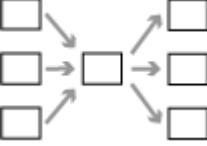
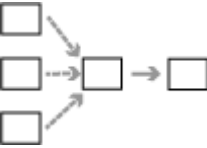
"Completed"	0
"Pseudo-normal"	1 to the set upper limit

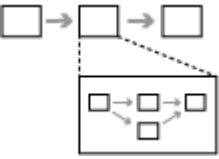
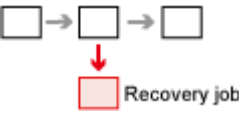
"Pseudo-normal" is differentiated from the "Completed" status, but processing is regarded as having terminated normally.

4.3.3 Basic Patterns for Job Flows

There are several basic job execution patterns that are often used for job flows (order in which jobs are executed). By combining and applying these patterns, advanced processing can be achieved.

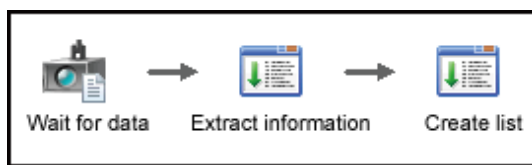
The following table explains some typical patterns. Design job flows by referring to these typical patterns.

job flow pattern	Overview
Flow task 	Is the most basic pattern, where jobs are executed in a sequence
Conditional branching 	Involves the flow to be branched based on the conditions specified beforehand
Consolidate-and-branch 	Requires waiting for multiple preceding jobs to finish before executing (AND condition), and then branching to multiple succeeding jobs
Event wait 	Involves evaluating multiple execution conditions with preceding jobs, and then executing a succeeding job if the condition for one of these jobs (logical OR condition) is met

Job flow pattern	Overview
Layering	 <p>Requires layering the flow by registering another job flow as a single job</p> <p>There are two methods: registering the job flow as a child job net, and using a linked job net.</p>
Recovery job	 <p>Involves registering a job (known as the "recovery job") beforehand that will start if a certain job terminates abnormally</p> <p>If that job terminates abnormally, the recovery job will start automatically.</p>

4.3.3.1 Flow Task

This is the most basic type of job net. Use this job net to execute multiple jobs in a sequence. An example is shown below.



If the preceding job terminates normally, the succeeding jobs are executed in a sequence.

4.3.3.2 Conditional Branching

This type of job net is used to specify a branching condition, causing the job flow to branch across different processing.

There are two types of branching:

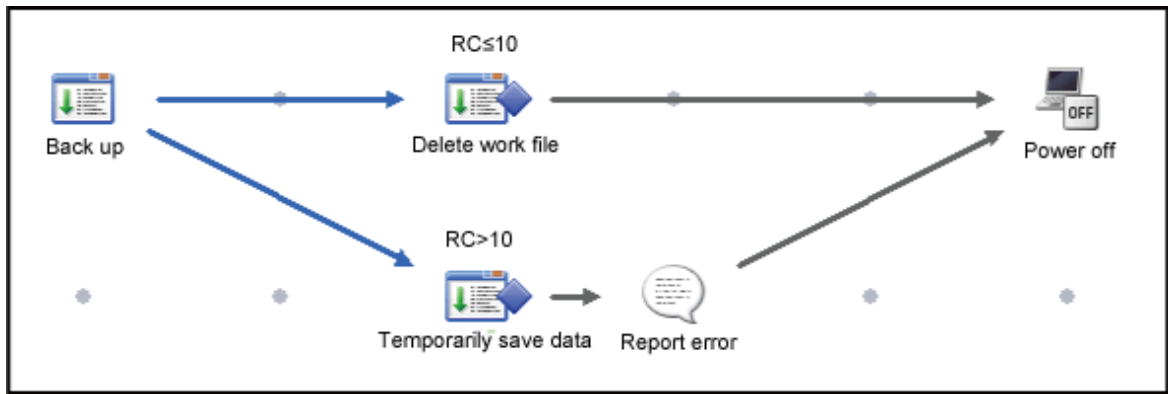
- Branching based on completion codes
- Branching based on criterial jobs

Branching based on completion codes

If the preceding job terminates normally or pseudo-normally, it is possible to determine how to start the succeeding job based on the value of the completion code for the preceding job. To determine which job starts next, specify the range of completion codes, which will be used as the execution condition, on the subsequent jobs.

In the following example, a backup task is performed first, and then if the completion code (RC) is no more than 10, the work file is deleted and power is turned off. If the completion code (RC) is more than 10, the data is saved temporarily, an error notification is sent, and then power is turned off.

This method is useful for performing different tasks that depend on the execution results of the preceding job.



Multiple succeeding jobs can be specified for a single preceding job, but from the perspective of the succeeding jobs, there can only be one preceding job. If multiple succeeding jobs have been specified for a single preceding job, and if the preceding job terminates normally or pseudo-normally, all the succeeding jobs that meet the conditions for the completion code of the preceding job will be started. If there are no such jobs, no jobs will be started.

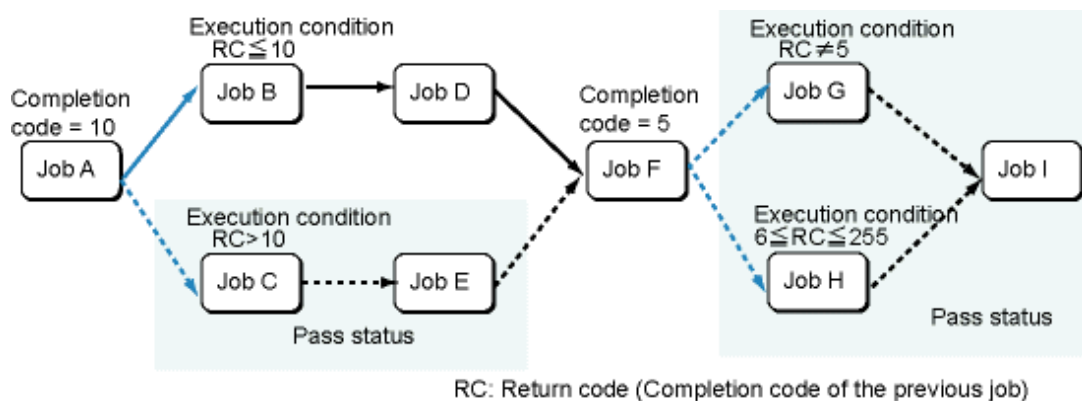
To branch succeeding processing using the completion code of the preceding job, you must configure the preceding job not to become "Abended" even if it has the completion code set as the branch condition. Refer to "4.3.2 Job Flow Operations" for details.

Information

Passed status

The status of a job when it does not start because its conditions are not met is referred to as the "passed status." Even if there are some "passed" jobs that remain without being executed, the job net will still terminate normally if all the other jobs (jobs whose status is not "passed") terminate normally.

The behavior of jobs when the succeeding job is determined based on the completion code of the preceding job is explained below by using the example of a job net with the configuration shown in the following figure.



1. If Job A terminates normally with the completion code "10", Job B will start because a completion code of 10 or less has been set as its execution condition. Job C will enter the passed status because its execution condition is not met. Job E, which waits only for Job C to terminate normally, will also enter the passed status automatically. As a result, Job F will wait only for Job D to terminate normally.
2. Job D will be executed if Job B terminates normally.
3. Job F will be executed if Job D terminates normally.
4. If Job F terminates normally with the completion code "5", both Job G and Job H will enter the passed status as their execution conditions are not met. Similarly, Job I will also enter the passed status

because all of its preceding jobs are in the passed status. The job net will terminate normally if Job F terminates normally, even though there are still some jobs that are in the passed state.

Branching based on criterial jobs

Succeeding jobs can be determined after evaluating the execution results of the preceding job (such as whether an expected file has been created).

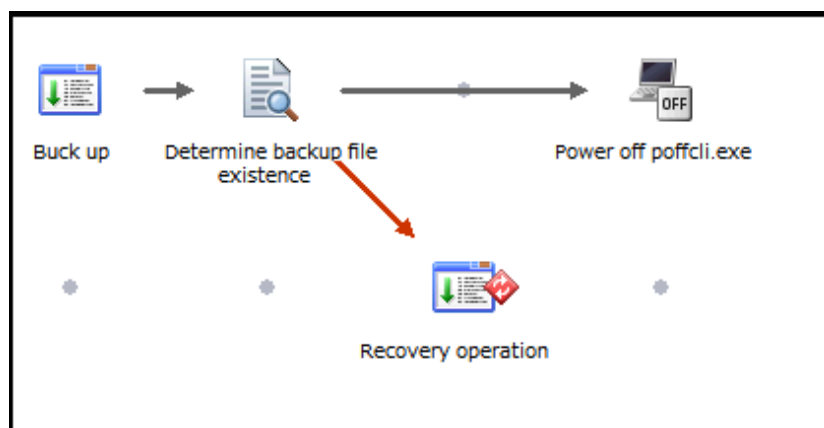
By branching the execution of the succeeding job based on the result of the evaluation, it is possible to perform particular processing only when the evaluation criteria are met. It is also possible to abnormally terminate a job net if the evaluation criteria are not met, or to use a recovery job to recover the situation.

In the following example, a backup task is performed first, and then the job net uses the "Determine backup file existence" job to evaluate whether the backup file exists.

The recovery processing is defined to execute if the completion code of the "Determine backup file existence" job is "1".

If the specified file exists, power will be turned off without executing the recovery processing. If the specified file does not exist, the recovery processing will be executed.

This method is useful for performing different tasks after evaluating the execution results of the preceding job.



The following criterial jobs evaluate the execution results of the preceding job:

- Determine File Existence job
- Determine Job Net Variable job
- Check service status job
- Check log file job

Notes on using "Determine Job Net Variable" jobs

These jobs evaluate job net variables that have been set on the schedule server. These jobs cannot be executed as network jobs or distributed execution jobs. The default host name specified for a job net is ignored by the "Determine Job Net Variable" job, so that the job is executed as a local job.

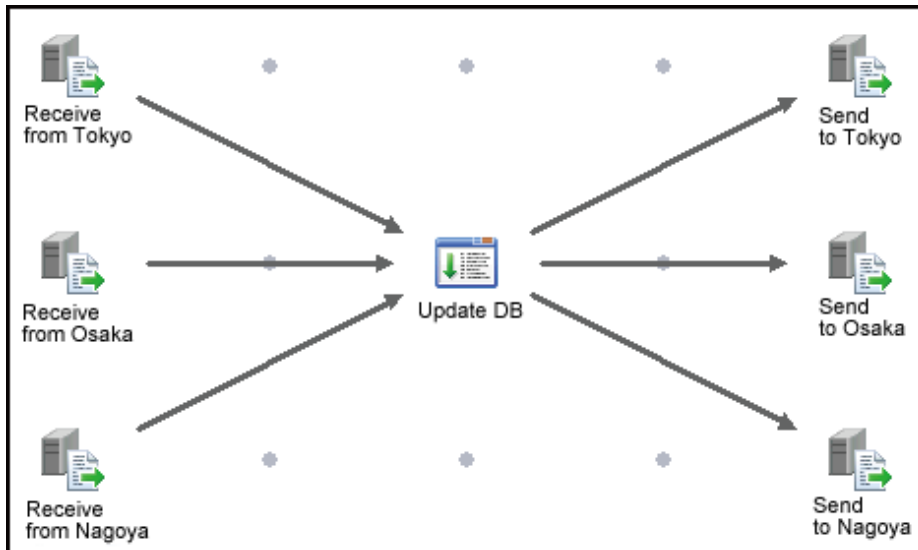
4.3.3.3 Consolidate-and-branch

This pattern involves waiting for multiple preceding jobs (linked using a logical AND condition) to finish, and then branching to multiple succeeding jobs.

This type of job net can be used in situations, such as when data is collected from various different locations and processed in a single location, and then the results are sent back to different locations. If wait with AND

condition is set, the succeeding jobs that wait with AND condition wait for all preceding jobs to become "Completed" before they are executed.

In the following example, the job net receives all the sales data from three servers (Tokyo, Osaka, and Nagoya), and then updates the database. The job net then sends the updated data back to each server.



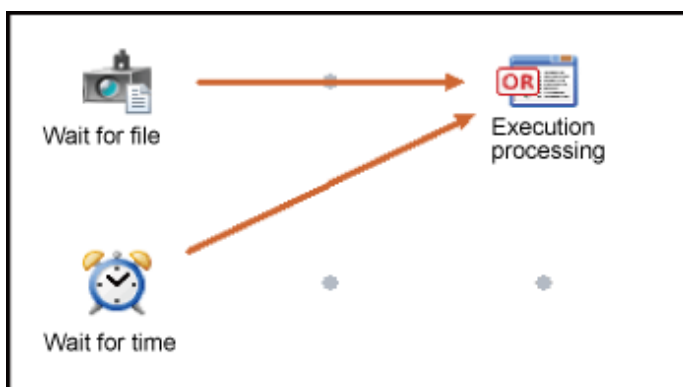
4.3.3.4 Event Wait

This type of job net can wait for any one of the preceding jobs linked with a logical OR condition, and then execute the succeeding job.

If an event wait job flow based on OR conditions is used, multiple execution conditions are evaluated for the preceding job and the succeeding job is executed if any one of these conditions is met.

In the following example, two jobs ("Wait for file" and "Wait for time") have been registered as preceding jobs that the succeeding job waits for.

When either of these conditions is met (when either the file is updated or the specified time is reached), the succeeding job is executed.



The following execution conditions can wait with OR condition:

- File creation, deletion, and update (file size or update time)
- Occurrence of a message event
- Time
- Lapse of a specified time period

Refer to "Wait with OR" Condition" in the *Systemwalker Operation Manager User's Guide* for information on waiting for events using logical OR conditions.

4.3.3.5 Layering

It is possible to register another job net within a job flow. This is called "job net layering."

The job net where another job net is registered is called the "parent job net," while the job net registered with the parent job net is called the "child job net."

For layered job nets, the depth of a layer is expressed as the "N-th layer."

Layering job nets has the following advantages:

- Visibility is improved

Appropriate layering makes the flow easier to understand and easier to monitor.

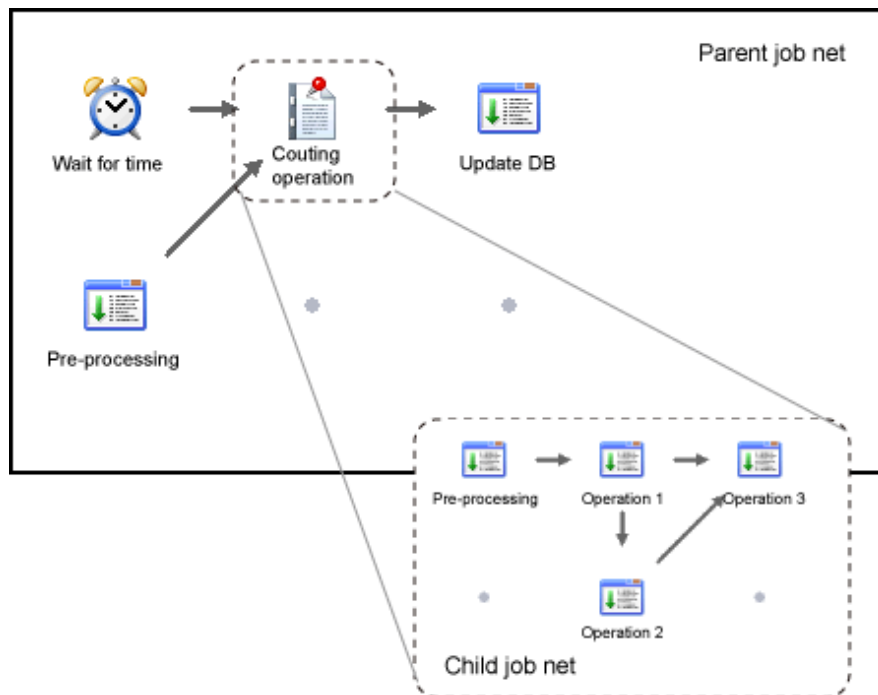
- Job nets can be started in an order

The second job net can be made to wait until the first job net has terminated before the second job net starts, just as jobs can be arranged in job flows.

- Number of jobs is reduced

A child job net is counted as a single job, so the total number of jobs is reduced.

The following example shows a processing sequence that performs aggregation processing (another job net) registered as a child job net.

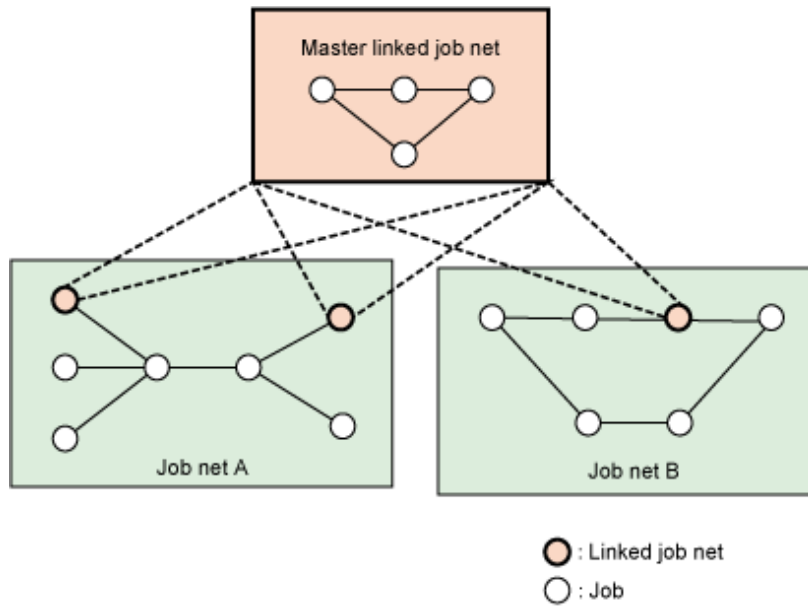


Layering based on linked job nets

Job nets can also be layered using linked job nets.

A "linked job net" is a copy of a master linked job net, and is the definition information for the source master linked job net. A linked job net can be registered as a job in multiple job nets.

In the following case, for example, three linked job nets have been registered as jobs with Job net A and Job net B, and all the three linked job nets refer to the same master linked job net.



Child job nets cannot be shared between different parent job nets. Accordingly, if it is necessary to have a child job net that can be registered multiple times and shared between different parent job nets, consider using a master linked job net or a linked job net.

Refer to ["4.4.1 Creating Job Nets that Can Be Shared as Jobs between Different Job Nets"](#) for details.

Notes on layering job nets

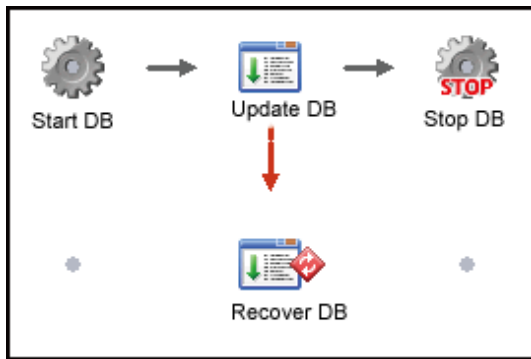
- Job nets can be layered up to five layers.
- A child job net is counted as a single job in the job net where it has been registered. Up to 255 jobs can be registered in a job net, with each child job net being counted as one job.
- There are several conditions that a job net must meet to be registered as a parent job net or a child job net. Refer to "Layering the Job Net" in the *Systemwalker Operation Manager User Guide* for details.
- There are several conditions that a job net must meet to be registered as a master linked job net. Refer to "Registering Linked Job Nets" in the *Systemwalker Operation Manager User Guide* for details.

4.3.3.6 Recovery Jobs

A "recovery job" is a job that is started when another job terminates abnormally, such as when an error occurs midway through an operation. Recovery jobs can be registered in advance as jobs that deal with errors that may occur, for example. If a job terminates abnormally, the recovery job will start automatically.

If the causes of errors can be identified to a certain extent, and the procedure for responding to these errors is also determined, recovery jobs make it possible to respond to errors automatically.

In the following example, if an error occurs with the task of updating the sales database, the "Recover database" recovery job will start a program for recovering the database.



Additionally, if a recovery job terminates normally, the job that originally terminated abnormally can be restarted.

The following explains the behavior of recovery jobs using a job net with the configuration shown in the above figure.

1. If the "Update database" job terminates abnormally, the "Recover database" recovery job will be executed.
2. If the "Recover database" recovery job terminates normally, and there is a specification to restart the original job, then the original job ("Update database") will be restarted only once.
3. If the preceding job "Update database" terminates normally, the succeeding job "Stop database" will be executed.
4. If the succeeding job ("Stop database") terminates normally, the job net will terminate normally.

Note that the job net will terminate abnormally in the following cases:

- If the "Recover database" recovery job terminates abnormally
- If the "Recover database" recovery job does not have the specification to restart the original job
- If the "Recover database" recovery job has the specification to restart the "Update database" source job, and the "Recover database" recovery job terminates normally, but the "Update database" job once again terminates abnormally when it is restarted

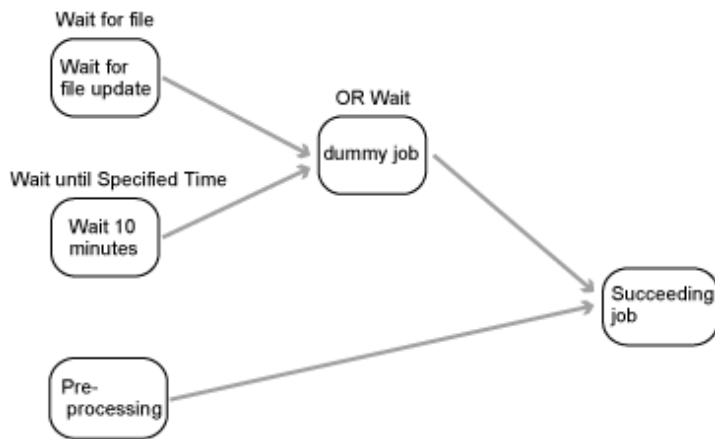
Notes on using recovery jobs

- Only one recovery job can be connected to each job.
- A single recovery job cannot connect to more than one preceding job.
- A regular job cannot connect to a recovery job as the succeeding job.

4.3.3.7 Advanced Usage

Waiting for events using a combination of OR and AND conditions

To create a job net that waits for events with a combination of both OR and AND conditions, use a "Dummy job" that simply terminates normally without actually doing anything but which links two or more event wait jobs with a logical OR condition. Then, create a succeeding job whose preceding jobs are the two jobs that you want to link using an AND condition. (In the following example, these two jobs are the "Dummy job" and the "Pre-processing" job.)



4.3.4 Determining Job Net Properties Information

When configuring the job net information, consider the following items with regards to the job net property information that is configured using the **Job Net Properties** window.

Refer to the "**Job Net Properties** window" in the *Systemwalker Operation Manager Online Help* for details.

Items to be considered

Job net comment

Determine the comments for job nets.

A job net comment is used as the display name for the Gantt chart and client.

Execution condition

Consider the execution condition that will determine when a job net will be started. Refer to "[5.3.1 Determining Execution Conditions](#)" for information on execution conditions.

End delayed monitoring for job nets at relative times

Determine whether to perform delayed monitoring for job nets, and if so, consider the estimated processing time. The job net will be monitored from the start time until the estimated end time, calculated according to the specified estimated processing time.



Instructions for carrying over job nets

Consider whether job nets should be carried over when using the Master Schedule Management function.

Specifying copy and startup

To execute multiple job nets concurrently from a single job net definition, consider concurrent execution using copy and startup.

Execution servers

Determine which server the jobs in the job net will be executed on.

4.4 Tips on How to Create Jobs and Job Nets

This section provides helpful information that you can refer to when creating jobs and job nets.

Refer to the following for useful information:

- "Design" in the *Systemwalker Operation Manager How-To Guide*

- Appendix A "Important Notes" in the *Systemwalker Operation Manager User's Guide*
 - "Notes When Registering a Large Number of Job Nets"
 - "Notes When Registering a Large Number of Jobs"
 - "Using Batch Files as Jobs [Windows version]"
 - "Considering Shell Scripts as Jobs [UNIX version]"

4.4.1 Creating Job Nets that Can Be Shared as Jobs between Different Job Nets

To create job nets that can be registered as shared child job nets in more than one parent job net, use linked job nets.

A linked job net can be registered as a job in multiple job nets. The job net that provides the definition information for linked job nets is the master linked job net. Linked job nets are registered as a copy of their master linked job net, and refer to the definition information in that master linked job net.

Linked job nets can be used in the following ways:

- More than one linked job net can be registered in a single parent job net.
- The startup days can be specified for linked job nets. By setting up the startup days, the configuration of a job net can be changed for different operating days.

Notes on using linked job nets

- Register master linked job nets and their linked job nets in the same project.
- Linked job nets can be combined using layering based on child job nets. Even if linked job nets are combined into layers, there can only be up to five layers.

The master linked job net that a linked job net refers to can also be layered without any issues. However, it is not possible to register a linked job net within another linked job net.
- A linked job net is counted as a single job in the job net where it has been registered. Up to 255 jobs can be registered in a job net, with each linked job net being counted as one job.
- Every time a linked job net is registered, the number of job nets in the project increases (linked job net, in addition to any child job nets that the linked job net may have). This may cause the schedule performance to deteriorate. Design the job net configuration, taking the number of job nets into consideration.
- Several conditions must be met to use a job net as the master linked job net. Refer to "Registering Linked Job Nets and Master Linked Job Nets" in the *Systemwalker Operation Manager User Guide* for details.

Chapter 5 Determining How to Start Jobs and Job Nets

This chapter explains when and how to start jobs and job nets.

It also explains the day change time and holiday calendars for job nets, which should be determined before setting up the startup schedules for job nets.

5.1 Determining the Day Change Time

Before determining the startup schedules for job nets, first start by determining the day change time (operation start time for one day) for the system.

Normally, the date changes from one day to the next at midnight (00:00), but with Systemwalker Operation Manager, the time when the date changes can be set to any desired time.

The day change time can be set per system and per subsystem, and is set to "00:00" when Systemwalker Operation Manager is installed.

Consider altering the day change time in the following situation:

- To adjust the day change time to match the schedule, such as treating job nets that will actually be executed tomorrow as though they are executed today for operational purposes

Suppose, for example, that there is a data center where the processing for batch jobs is often delayed until the following day. In this case, scheduling jobs becomes complicated because jobs relating to the batch job processing need to be scheduled, taking into account the date changeover. In cases like this, by altering the day change time to a later time and making it appear that the batch job processing finishes on the same day that it starts, it is easier to schedule and monitor jobs.

EE

- To set a different day change time per job

Using a different subsystem for each job enables you to set a different day change time for each subsystem. If you do not set a subsystem day change time, the system day change time is used.

Note that the system day change time is set in the SYSTEM_CALENDAR. Refer to "3.1 Using the SYSTEM_CALENDAR" for details on the SYSTEM_CALENDAR.

Note

If the day change time is altered after schedules, such as power schedules or job net schedules, have been set up, this can have serious consequences, such as changes to the power schedules or job net schedules that have been set up. To alter the day change time, consider carefully so as not to make the alteration during operations or after settings have completed. Then, configure settings in the following order:

1. Set up the SYSTEM_CALENDAR (working days, holidays, and the day change time).
2. Set up power schedules.
3. Set the subsystem day change time.
4. Set up schedules for job nets.

Refer to "5.1.1 Effects of Altering the Day Change Time" for details on the effects of altering the day change time.

5.1.1 Effects of Altering the Day Change Time

This section explains the effects of altering the day change time after schedules, such as power schedules or job net schedules, have been set up.

Effects on power schedules (if the system day change time is altered)

- If the day change time is changed to within a defined operation range, the operation range is split into two operation times, and the new ranges will be registered. Accordingly, the power on date/time and shutdown date/time may change. After the day change time is changed, check the work start and end times for each pattern.
- If the day change time was set in a range between the time in **Work end time if next day is a holiday** of the **Power Control Pattern** dialog box and 30 minutes prior to that time, the work end time will be invalid. Additionally, if the day change time was set in a range between the time in **Work start time if previous day was a holiday** of the **Power Control Pattern** dialog box and 30 minutes after that time, the work start time will be invalid.
- If the day change time is changed to a time between the times set for **Work start time if previous day was a holiday** and **Work end time if next day is a holiday**, note that, compared with the pre-change operation, the work start time for when the previous day is a holiday and the work end time when the next day is a holiday will be delayed by 24 hours. If this causes an operational problem, either restore the original day change time, or change **Work start time if previous day was a holiday** or **Work end time if next day is a holiday** according to the operation.

Effects on schedules for job nets (if the system and subsystem day change time are altered)

- If the day change time is altered after job nets have been registered, the job nets for all users will be rescheduled. If a subsystem day change time is being used, the job nets for all users in that subsystem will be rescheduled.
- The relationships relative to the execution time will change, which may cause the scheduled startup day for the job net to change. After altering the day change time, check the scheduled startup days.
- If **Execution interval** is specified as the job net execution condition, the day change time must not be set between the start and end times. Otherwise, the job net may no longer start at regular intervals. After the day change time is changed, check the start and end times.

5.2 Determining Holiday Calendars for Job Nets

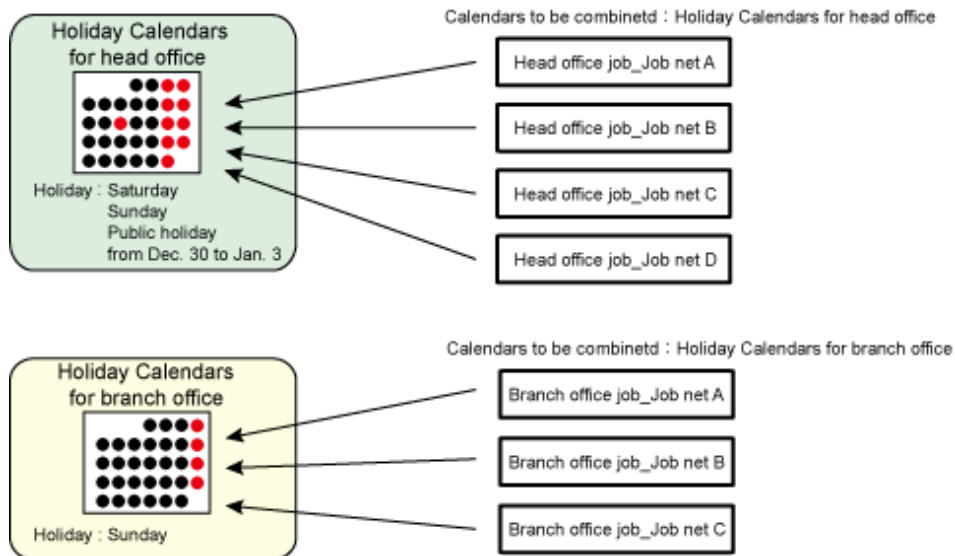
Job nets can be started automatically based on schedules that are set up by specifying the execution conditions and the startup days for each job net.

However, if multiple job nets are run with the same holidays, the startup days will need to be set up, taking into account the holidays for each job net, and so shift processing will become complicated.

In such cases, consider creating a calendar that specifies job net holidays. The startup schedule for job nets can then be set up by combining the job net holiday calendar with the definitions for job net startup days.

For example, suppose there is a head office and a branch office with different holidays, and the business operations of both the head office and the branch office are conducted using multiple respective job nets. In this case, create different holiday calendars for the head office and the branch office. Use the head office holiday calendar for the head office, and the branch office holiday calendar for the branch office. Then, for each office, set up actions (such as shift processing) beforehand for situations where a startup day coincides with a holiday.

This way there is no longer any need to be aware of the respective holidays when setting up the startup days for each job net. Situations where the startup days coincide with holidays will also be dealt with automatically, making it much easier to set up the startup days.



Note that when Systemwalker Operation Manager is installed, the SYSTEM_CALENDAR is provided as the standard calendar. The SYSTEM_CALENDAR is used for schedules for turning the power on and off, but it can also be used as a holiday calendar for job nets. If job net holidays are the same as the holidays in the SYSTEM_CALENDAR, there is no need to create additional calendars.

Consider creating additional job net holiday calendars (other than the SYSTEM_CALENDAR) as required if job net holidays differ from the SYSTEM_CALENDAR, or if there are different holidays for different business operations and multiple calendars need to be used for each business operation.

Refer to "3.1 Using the SYSTEM_CALENDAR" for details on the SYSTEM_CALENDAR.

Note the following points when creating calendars:

- Only the SYSTEM_CALENDAR can be used to schedule when the power is turned on and off.
- There is no upper limit for the number of calendars that can be registered, but it is recommended that you register a maximum of 100.

If large numbers of calendars are registered or when the calendar is edited, it may delay the processing of the arrival of the day change time.

Changing calendars after job net schedules have been set up will affect the schedules that have already been set up, just as altering the day change time affects existing schedules. Refer to "3.1.1 Effects of Changing the Calendar" for details.

5.3 Determining the Startup Method for Job Nets

Job nets are started based on the execution condition and startup day information for job nets.

This section explains how to determine the startup method including the job net execution condition and startup day information defined in the **Job Net Properties** window.

When determining the execution condition and startup days, it is necessary to first understand the timing of schedule creation.

Refer to "Triggers for Job Net Schedule Processing" in the *Systemwalker Operation Manager User Guide* for details on the timing of schedule execution.

5.3.1 Determining Execution Conditions

This section explains how to determine the execution conditions for job nets.

One of the execution conditions given below must be specified for job nets.

Execution condition	Description	Relationship with startup day settings
None	<p>Does not specify an execution condition.</p> <ul style="list-style-type: none"> - "None" is used for child job nets and master linked job nets. 	<p>If you specify "Valid only on startup days" (*1), the job net is started on the startup day specified for child job nets or master linked job nets.</p> <p>If you do not specify "Valid only on startup days" (*1), the job net is started on the startup day for the parent job net.</p>
Execute at	<p>Starts the job net at the specified time(s)</p> <ul style="list-style-type: none"> - Up to six time values can be specified. - If you specify "Estimated end time", a job net that does not end in accordance with the schedule changes to "End delayed" status, and you can check it on the Gantt chart or some other method (provided that only one time has been specified). - Even if a job net is not started because, for example, the power is not turned on at the specified time, you can ensure that the job net is started the next time the power is turned on (provided that only one time has been specified). 	Valid only on days set as the startup days.
Execution interval	<p>Starts the job net at regular intervals</p> <ul style="list-style-type: none"> - Any interval from 1 through 240 minutes can be specified. - You can also specify a period (start time and end time) during which to start the job net at intervals. 	Valid only on days set as the startup days.
Start at the time of server startup	<p>Starts the job net when the schedule server starts</p> <ul style="list-style-type: none"> - You can set the job net to start once a day (the period 	Valid only on days set as the startup days.

Execution condition	Description	Relationship with startup day settings
	from the day change time to the next day change time).	
Start only when message event has occurred	<p>Starts the job net only if a message event occurs.</p> <ul style="list-style-type: none"> - You can set the job net to start once a day (the period from the day change time to the next day change time) when the first message event occurs. - If you specify "Scheduled start time", a job net that does not start even when the scheduled time has passed changes to the "Start delayed" status and you can check it on the Gantt chart etc. (provided that you specified Once in a day). 	<p>Waits every day for a message, regardless of the startup day setting (default).</p> <p>To make the job net wait for a message only on days set as startup days, you must specify "Valid only on startup days" (*2).</p>

*1: **Valid only on startup days** in the **Control information** tab of the **Add/Change - Job** window

*2: **Valid only on startup days** in the **Message** tab of the **Job Net Properties** window

You can specify the execution condition in the **Standard information** tab of the **Job Net Properties** window.

Combining message event occurrence and other execution conditions

You can register message events even when you have specified the following execution conditions:

- **Execute at**
- **Execution interval**
- **Start at the time of server startup**

Essentially, if you combine conditions, the job net starts each time an execution condition is met.

Execution condition	Timing of job net startup
Combination of message event and Execute at	<p>The job net starts when the message event that is waited for occurs and the specified time is reached.</p> <ul style="list-style-type: none"> - To start the job net only if both the message event occurrence and Execute at conditions are met (AND condition) <p>Specify only one time (once a day) and specify Wait for startup time</p>
Combination of message event and Execution interval	The job net starts when the message event occurs and the fixed interval has elapsed.
Combination of message event and	The job net starts when the message event occurs and the server starts.

Execution condition	Timing of job net startup
Start at the time of server startup	

If combining execution conditions, specify **Execute at**, **Execution interval**, or **Start at the time of server startup** as the execution condition in the **Standard information** tab of the **Job Net Properties** window, and register the message event to be waited for in the **Message** tab.

5.3.1.1 Waiting for Message Event Occurrence as the Execution Condition

This section explains points to consider before you use message event occurrence as an execution condition.

Queuing message event name

Consider the name of the message event to be waited for. The job net starts when the message registered here occurs.

Up to 70 message events can be registered for a single job net.

Wait conditions

If you have registered multiple queuing message events, consider whether the job net will wait for all the message events to occur before starting (AND condition) or whether to start the job net when any one event has occurred (OR condition).

Behavior when a message event occurs on an unscheduled day

By default, a job net starts when a message event occurs, regardless of the startup day setting.

To start a job net on startup days only, specify **Valid only on startup days**.

Handling of message events that occur while a job net is executing

You can specify whether to start a job net each time the same message is generated while the job net is running.

If you specify **Valid during job net execution**, message events are accepted even while the job net is running. Therefore, after the job net ends, it is successively started and stopped once for each time the message was received while the job net was running.

You can define queuing message events in the **Message** tab of the **Job Net Properties** window.

To use message receipt as the trigger for starting a job net, specify the queuing message event name explained above in the job net that will receive the message. Then, in the job that will generate the message event, register the jobschmsgevent/jobschmsgeventex command and issue a message event.

Information

Counting message events

Before defining message event occurrence as the execution condition of job nets, you must understand how message event occurrences are counted and cleared. The method used for counting message events depends on, for example, whether the count is started by schedule or manually, and whether only message event occurrence is set as the execution condition or message event occurrence is combined with the startup time as the execution condition.

Refer to "Message Events and Job Net Actions" in the *Systemwalker Operation Manager User's Guide* for details.

5.3.2 Determining Startup Days

This section explains how to determine the startup days for job nets.

You can define startup day information in the **Startup days** tab of the **Job Net Properties** window.

Determine the following three items in relation to the startup days:

1. Determine the basic settings for the startup days.
2. Determine the detailed settings for the startup days, such as the operation period and shift settings for the startup days.
3. To set up individual startup days that do not fall into patterns (for example, starting job nets on a particular date, such as "December 20"), determine the settings for unscheduled days after determining the basic settings.

Note that job nets will not run on a day that has been set up as a startup day if the power to the server is not turned on. Ensure that the days specified as the startup days for job nets are days when the power will be turned on.

5.3.2.1 Basic Startup Day Settings

There are three methods of setting up the basic startup days, as described below. Determine which method to use to set up the basic startup days.

Selecting a schedule pattern

This method involves setting up the basic startup days by selecting a schedule pattern (a startup day pattern for one year).

Registering a schedule pattern beforehand makes it easy to register the startup days for job nets, and so, this method is suitable for situations where there are multiple job nets with similar startup day patterns.

For this method, a schedule pattern must be created beforehand. Refer to "Registering the Schedule Pattern" in the *Systemwalker Operation Manager User Guide* for details on schedule patterns.

Using schedule of other job nets

This method uses the schedule of another job net for which the startup days have already been set up.

Use this method to start a job net on the same days as schedule of another job net.

In additions, to use schedule of another job net, it must be registered as a base job net.

Specifying startup days directly

This method involves specifying the startup days directly.

Use this method to specify the basic startup days when it is necessary to set up specific startup days for individual job nets.

For example, the following specifications are possible:

- Particular date for every year

For example, "April 4, every year", the particular pattern for every year can be specified.

- Particular date or day of the week for every month

For example, "the 20th of every month", or "the Monday of the week", the particular pattern for every month can be specified. It is also possible to specify "the last day of the month."

- Business days

It is possible to specify which day is the first day of the business, and then specify a certain number of business days, either counting forwards from the first day of the business month or counting backwards from the last day of the business month (the day before the day specified as the first day of the business month).

5.3.2.2 Detailed Settings for Startup Days

If necessary, determine the following detailed settings:

Startup day shift processing

To shift the startup days relative to the startup days specified using the **Schedule pattern**, **Another job net**, or **Other** radio buttons, determine the shift direction, the number of days to shift, and how to handle holidays.

How to handle a startup day that coincides with a calendar holiday

Determine how to handle a startup day that coincides with a holiday in the calendar. The following items can be specified:

- Whether to start the job net
- Whether to shift the startup day
- Shift direction
- Number of days to shift

Using calendars and the conditions for combining calendars

To use calendar holiday settings, specify which calendar to use. To use multiple calendars, determine the conditions (**AND** or **OR** conditions) for combining the calendars as well.

Note that the holidays that have been specified in the SYSTEM_CALENDAR are used as the schedule for turning the power on and off. This means that if Power Control is performed, neither servers nor job nets will start on days that have been specified as holidays in the SYSTEM_CALENDAR even if the SYSTEM_CALENDAR is not selected as the job net holiday calendar.

Job net operation period

To restrict the operation period of a job net, determine the operation period.

5.3.2.3 Exception Days

To set up individual startup days that do not fall into patterns (for example, starting job nets on a particular date, such as "December 20"), design and adjust the settings for unscheduled days at the end after the basic startup days settings are complete.

You can fine tune startup days in the **Startup Days** window, **Startup Days (Preview)** window, or **Application Plan** window. However, note that the startup days specified in these windows are known as exception days and may be cleared when the calendar or the startup days are changed.

Refer to "Startup Day Information" in the *Systemwalker Operation Manager User Guide* for details on exception day.

5.3.2.4 Startup Days of Layered Job Nets

Essentially, child job nets in a layered job net start when their parent job net starts. However, you can start child job nets using a different schedule from the parent job net by setting a different startup day for each child job net.

Specifying startup days for the child job nets enables you to, for example, change the job net flow configuration by day.

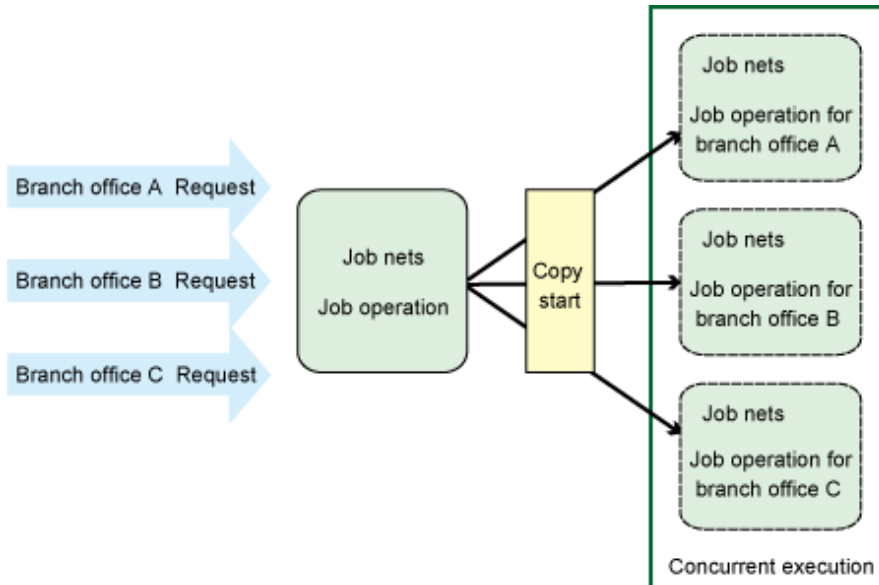
Refer to "I Want to Change the Flow Configuration per Day" in the *Systemwalker Operation Manager How-To Guide* for details.

If you specify startup days for a child job net or master linked job net, you must validate the specified startup days by specifying **Validate only on startup days** in the **Add/Change - Job** window for the child job net or master linked job net.

5.3.3 Determining Concurrent Execution

You may want to execute the same job net concurrently, such as for running aggregation processing for individual stores in parallel, as shown in the example below. In this case, consider concurrent execution using copy and startup.

You can execute multiple instances of the same job net at one time without increasing the number of job net definitions in the system as a whole.



Before starting concurrent execution using copy and startup, you must specify copy and startup in the property information of the copy source job net. Job nets for which you specify copy and startup are always copied and started.

The execution result for each job net that was copied and started is preserved; it is not overwritten. If the number of saved execution results becomes too high, performance may deteriorate. Therefore, if running concurrent execution that uses copy and startup, consider the following when configuring the number of job nets to save so that it is limited to the requisite minimum.

- Number required for job net monitoring
- It should not affect the job net execution performance

Refer to "Notes When Registering a Large Number of Jobs" in the *Systemwalker Operation Manager User's Guide* when determining the number of job nets and carefully verify operations when designing them.

5.4 Tips on How to Start Job Nets

This section introduces some useful know-how on how to start job nets.

The manuals listed below also contain useful know-how. Refer to the following:

- "Design" in the *Systemwalker Operation Manager How-To Guide*
- Chapter 4, "Scheduling Jobs" in the *Systemwalker Operation Manager User's Guide*
-
- "Using the Creation, Update, and Deletion of Files as Startup Triggers"
- "Using Mail Receipt as a Startup Trigger"
- "Executing a Job Net Concurrently Using Copy and Startup"

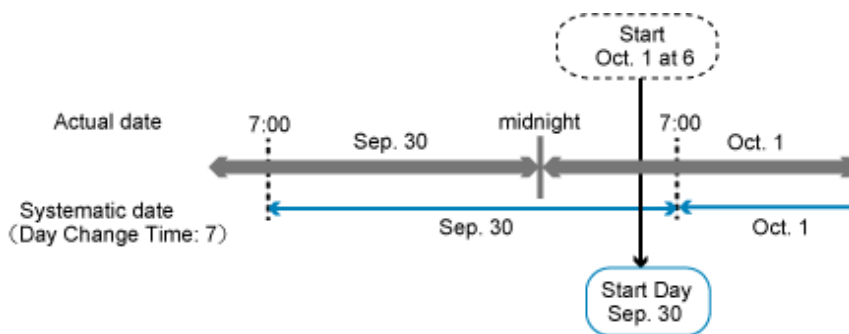
- "Using Variable Parameters to Link to a Job Net when a Message Event Occurs"
- Appendix A, "Important Notes" in the *Systemwalker Operation Manager User's Guide*
- "Shutting Down the System at Optional Times [Windows version]"
- "Considering the Startup Time"

5.4.1 Relationship between Startup Days and the Day Change Time

The day change time is the time when a new day starts, and times prior to the day change time are counted as part of the previous day. Accordingly, when designing schedules, it is necessary to pay attention to the relationship between the execution times and the day change time.

To start a job net earlier than the day change time, set the startup day to the previous day.

For example, suppose there is a system where the day change time has been set to 7:00 A.M. To start a job net at 6:00 A.M. on October 1, specify September 30 (the previous day) as the startup day.



5.4.2 Executing Job Nets in Order

Job nets are started individually based on their execution conditions and startup days, but it is also possible to execute job nets in a fixed order, as is the case with jobs.

However, if you simply set staggered execution times for two job nets so that the second one can carry on after the processing for the first one has finished, there is a chance that the second job net will start before the first job net has finished.

To ensure that the second job net waits for the first job net to finish processing before it starts, use either of the following methods to specify the job net execution order:

- Job net layering

A job net can be registered as a job within another job net. (This is called "job net layering.") By using job net layering, job nets can be scheduled by connecting job net icons on the screen to create a flow diagram.

- Linking with message events

This method involves scheduling the job nets so that the second job net is triggered to start when it receives a message from the final job in the first job net.

To generate a message event, register the **jobschmsgevent** command in the final job of the first job net. For the job net that is triggered to start by receiving the message, specify the settings in the **Standard information** and **Message** tabs of the **Job Net Properties** window.

- Grouping

Grouping multiple job nets enables you to execute the job nets in order. You can schedule the job nets by connecting them using the flow for creating a group and then visually creating a flow diagram.

These methods have their advantages and disadvantages, so determine which method to use according to the actual operation.

Scheduling method	Advantage	Disadvantage
Job net layering	The relationship between the first (preceding) and second (succeeding) job nets can be edited and checked visually using a flow diagram. Job nets are executed as specified by the flow, making it easy to keep track of the operational status.	Only job nets within the same project can be linked.
Linking with message events	Job nets can be linked to job nets in other projects or on other systems.	It is difficult to check the operational status of job nets, and the operation can easily become complicated as a result of message events being cleared.
Grouping	The sequential relationship can be visually edited and confirmed using the flow. A message or time can be set for starting job nets in a group.	This function groups job nets to be started at a certain time once a day, and was designed for changing the configuration of job nets executed on certain days, so numerous restrictions apply.

If you are not certain which method to use, it is recommended to use job net layering, which is easy to understand, has fewer restrictions, and where the relationship between the first and second job nets can be edited and checked visually using a flow diagram.

5.4.3 Using the Master Schedule Management Function

The Master Schedule Management function manages and operates the schedule information of job nets for each working day.

Consider using the Master Schedule Management function in the following situations:

- When there are a large number of jobs and job nets, which interfere with the execution of other business operations during the day
- When there are job nets whose execution conditions are not met during the working day, or which do not complete within the working day because they execute for a long time
- When there are job nets that do not complete within the working day and you want to control the schedule for the next day

Normal operations are recommended if most jobs and job nets complete within the working day when they are executed, and there are few jobs or job nets whose execution extends into the following day.

Refer to the *Systemwalker Operation Manager User's Guide - Master Schedule Management* for details on how to use the Master Schedule Management function.

Chapter 6 Designing the Job Execution Environment

The job execution environment includes queues for accepting jobs that have been submitted.

This chapter explains how to design queues, as well as the considerations for using the execution servers for jobs.

6.1 Queue Environment

The six queues shown in the table below are provided by default when Systemwalker Operation Manager is installed.

Standard settings are configured when Systemwalker Operation Manager is installed, so queues can be operated without any particular changes being made to the settings. The table below lists the settings that are defined in the **Operating control** tab of the **Define Operating Information** window and configured at installation.

Queue settings at installation

Registered queues	<ul style="list-style-type: none"> - queue1 (*1) - queue2 - queue3 - queue4 - queue5 - online1
Maximum number of jobs that can be executed simultaneously (Max Execution Jobs for all servers on which Systemwalker Operation Manager is installed)	30 (Windows) 256 (UNIX)

*1:

"queue1" is set as the default queue.

It is possible to specify the destination queue for each job or job net, but if a queue is not specified, jobs or job nets will be submitted to the default queue.

The table below lists the main settings at installation for each queue. These settings are defined in the **Create/Edit Queue** window.

Queue status at installation

Maximum number of jobs that can be executed simultaneously (Max Execution Jobs for a queue)	[Windows] queue1/ online1 : 30 queue2/ queue3/ queue4/ queue5 : 10 [UNIX] queue1/ online1 : 256 queue2/ queue3/ queue4/ queue5 : 99
Limit for job elapsed time	No limit

Number of jobs that can be submitted (Total number of jobs that can exist in a queue)	No limit
Job priority	Set to 31 by default
Job execution priority	Not specified

Note that when jobs are executed, they reside in memory as processes until the **Max Execution Jobs** setting is reached. By adjusting queue settings, such as **Max Execution Jobs**, it is possible to create a safe, efficient job execution environment.

For this reason, take into account factors, such as the number of jobs that will be executed by business applications, the execution time for each job, the hardware performance, and the load on servers and, if the settings need to be reviewed, consider changing the queue settings.

Designing queues

Design the queue configuration and settings in situations, such as the following:

- When simultaneously executing a large number of jobs
- When terminating jobs that have been running for a certain amount of time
- When minimizing the impact on online job responses
- When executing online jobs

Limiting the number of jobs that can be executed simultaneously

Limitations can be imposed on the number of jobs that can be executed simultaneously, both for the entire system and for jobs in queues.

The relationship between the limit defined for each differs as follows:

- a. If the sum total of jobs executed simultaneously in queues is equal to the total number of jobs executed simultaneously on the entire system:

The actual number of jobs that are executed simultaneously on a queue is guaranteed to increase until **Max Execution Jobs** is reached, and this number will not be affected by the number of jobs being executed simultaneously in other queues.

If the number of jobs being executed simultaneously on another queue is less, in other words, if the number of jobs being currently executed on the entire system is less than **Max Execution Jobs** configured for the entire system, the queue cannot simultaneously execute more jobs than **Max Execution Jobs** configured for the queue.

- b. If the sum total of jobs executed simultaneously in queues is greater than the total number of jobs executed simultaneously on the entire system:

The actual number of jobs that are executed simultaneously is not guaranteed to increase until **Max Execution Jobs** is reached in each queue, and this number will be affected by the number of jobs being executed simultaneously in other queues.

If the number of jobs being executed simultaneously in another queue is less, the queue can execute more jobs simultaneously compared to the case described in "a."

6.2 Queue Design Examples

This section provides queue design examples.

Refer to "Defining the System Operating Information" in the *Systemwalker Operation Manager Installation Guide* for details on how to define queues.

6.2.1 When Simultaneously Executing a Large Number of Jobs

For schedules where a large number of jobs are executed simultaneously, adjust the **Max Execution Jobs** setting in queues to reduce the load on the system, and prevent memory and other resources from running out.

Another example is when a large number of files or events are received from numerous branch offices or departments simultaneously, and business operations are triggered by the reception of these files or events. In this case, adjusting the execution time does not help at all, because the business operations do not start at particular times. There is a possibility that a large number of files or events will be received simultaneously, so if **Max Execution Jobs** is not specified, a large number of jobs will be executed at the same time. In such cases, adjusting **Max Execution Jobs** can suppress operational load to a certain level.

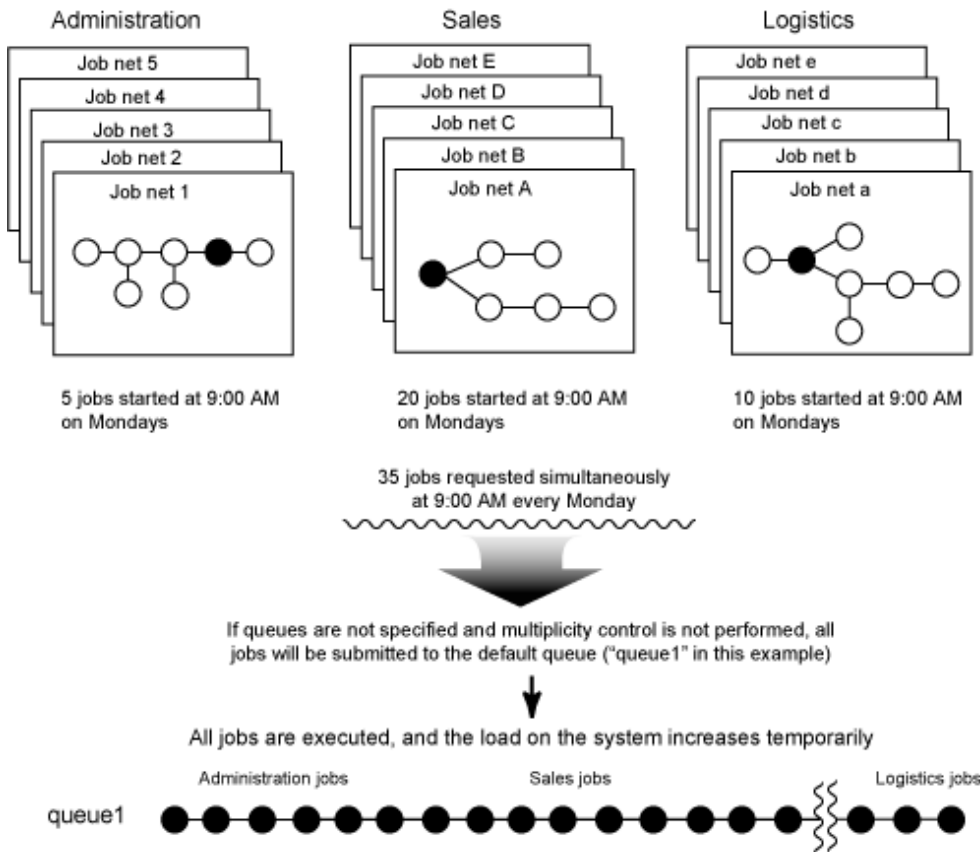
The following example shows how to adjust **Max Execution Jobs**.

Note that the number of jobs to be executed simultaneously can be changed dynamically during operations.

Overview

There are three types of business schedules, depending on the type of business operation.

For all business operations, all jobs are scheduled to start at 9:00 A.M. on Monday mornings.



For the operations described above, simply scheduling the execution start times of jobs will cause system load to temporarily increase considerably every Monday morning.

In such cases, design queues by using **Max Execution Jobs** to adjust the load.

Adjustment method

Create a separate queue for each different type of business operation, and configure the **Max Execution Jobs** setting for each queue.

Specifying the **Max Execution Jobs** setting has the following benefits:

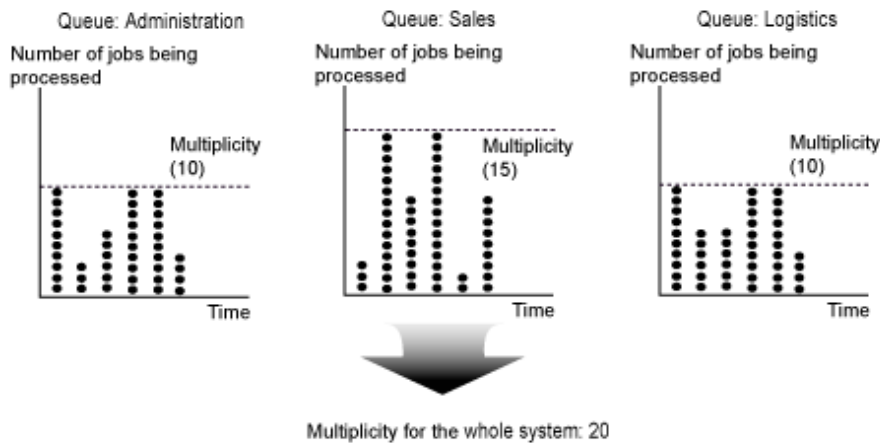
- The number of jobs that can be executed simultaneously per queue is limited, and so load is restrained to a certain level.
- If different business operations are allocated to different queues and different **Max Execution Jobs** settings are specified for each queue, even if a large number of jobs are executed for one business operation, the impact on other business operations (in relation to not being able to execute jobs) can be limited.
- Business applications can be created more easily, because it is no longer necessary to be aware of the number of jobs that may be executed simultaneously when scheduling jobs.

The adjustment method is explained below.

Set **Max Execution Jobs** for the system to "20", and set **Max Execution Jobs** for each queue as given below.

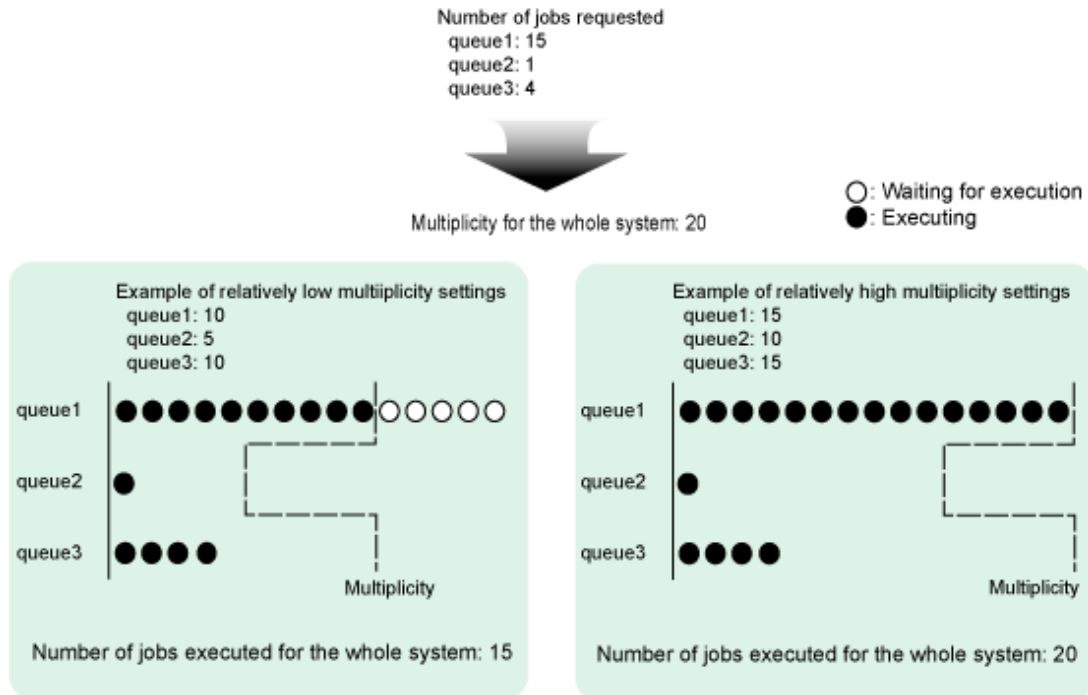
Business name	Number of jobs in a queue that can be executed simultaneously
Administration	10 jobs
Sales	15 jobs
Logistics	10 jobs

This enables the number of jobs that are executed simultaneously per business operation to be restrained to a certain level, making it possible to avoid situations where jobs for some business operations cannot be executed because a large number of jobs are being executed for one particular business operation.



Note that setting relatively high **Max Execution Jobs** for each queue, and then restricting **Max Execution Jobs** for the entire system to a certain level allows system resources (resources that are used when jobs are executed) to be used more efficiently.

An example is shown below.



In the above example, the following number of jobs are simultaneously requested to each queue when **Max Execution Jobs** (number of jobs that can be executed simultaneously) for the entire system is set to 20.

- Queue 1: 15 jobs
- Queue 2: 1 job
- Queue 3: 4 jobs

In the example to the left, **Max Execution Jobs** has been set relatively low. In this case, the system as a whole can execute up to 20 jobs, but the actual number of jobs that can be executed simultaneously is controlled by the **Max Execution Jobs** setting for each queue, so that the free space in Queue 2 and Queue 3 cannot be used efficiently, and in practice a total of only 15 jobs can actually be executed.

In the example to the right, **Max Execution Jobs** has been set relatively high. In this case, the **Max Execution Jobs** setting for each queue include a certain degree of margin, so that the free space in Queue 2 and Queue 3 is used, and all 15 jobs can be executed by Queue 1. As a whole, the system can execute 20 jobs.

In this way, careful consideration is required to ensure that **Max Execution Jobs** is not set too low, so that the jobs for busy operations can be executed efficiently when some margin becomes available in one of the queues.

Note that **Max Execution Jobs** can be configured as follows:

- Number of jobs that can be executed simultaneously across an entire system
Select the **Limit the number of jobs to execute simultaneously** check box in the **Operating control** tab of the **Define Operating Information** window, and specify the value in **Max Execution Jobs**.
- Number of jobs that can be executed simultaneously in a queue
Select the **Limit the number of jobs to execute simultaneously** check box in the **Edit Queue** window, and specify the value in **Max Execution Jobs**.

Refer to "Defining the System Operating Information" in the *Systemwalker Operation Manager Installation Guide* for details on the configuration procedures.

6.2.2 When Executing Online Jobs

If an online job is executed in a queue that also contains regular batch jobs, the online job may not be able to start at the scheduled time due to the **Max Execution Jobs** control within the queue.

For online jobs, use the "online1" queue that has been defined exclusively for online jobs.

Initially, only the following items are set up for the dedicated queue for online jobs ("online1").

Queue Name	online1
Limit the number of jobs to execute simultaneously (Max Execution Jobs)	30 [Windows], 256 [UNIX]
Default Job Priority	31

Note that if a limit is defined for the job elapsed time, the online jobs will be canceled in accordance with the limit that is defined, so limit for the job elapsed time should not be defined for the dedicated queue for online jobs.

Also, if **Max Execution Jobs** control is being performed within the system, it will be necessary to review the definition values. Design carefully so that **Max Execution Jobs** control does not prevent online jobs from starting at their execution time.

6.2.3 When Terminating Jobs that Have Been Running for a Certain Amount of Time

Jobs can be terminated after a certain amount of time has elapsed using one of the following two methods:

- a. Configure **Cut-off execution time** when registering a job

Use this method to monitor the elapsed time of specific jobs, and then terminate those jobs, or to configure detailed specifications for the maximum elapsed time for a particular job and its subsequent termination

- b. Configure limit for job elapsed time for a queue

Use this method to collectively terminate all jobs, rather than a specific job, according to the job elapsed time

Use both "a." and "b." above to monitor all jobs for a uniform elapsed time, and then terminate, as well as to make detailed specifications for the maximum planned execution time of individual jobs and their subsequent terminations.

If both methods are used, the jobs will terminate at the time specified in **Cut-off execution time** when the jobs were registered. If this setting has been omitted, the jobs will terminate at the time specified in limit for job elapsed time for the queue.

If specifying **Estimated processing time** when registering a job, note that the value entered must be less than the value of **limit for job elapsed time** for the queue.

If the value specified for **Estimated processing time** when registering a job is greater than the value of **limit for job elapsed time** for the queue, the **Estimated processing time** specification will be ignored.

6.2.4 When Setting Up an Operation where Jobs Do not Have to Wait before Executing

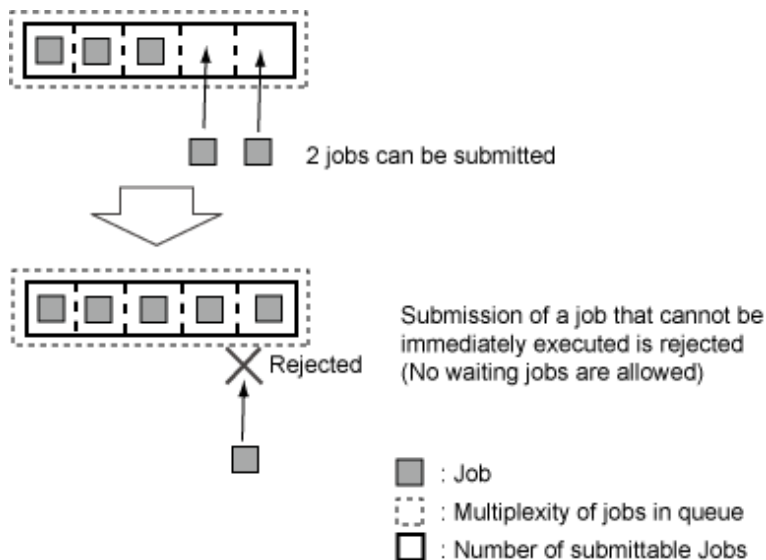
If the same limit value is specified for both the total number of jobs that can be executed simultaneously and the maximum number of jobs that can be submitted in a queue, jobs cannot be submitted beyond the **Max Execution Jobs** limit, which means that jobs that cannot be executed immediately will be rejected. In this way, it is possible to set up an operation where there are no jobs waiting to be executed.

The total number of jobs that can be executed simultaneously in a queue is defined using the **Limit the number of jobs to execute simultaneously** and **Max Execution Jobs** settings in the **Create/Edit Queue**

window. The maximum number of jobs that can be submitted is defined in the same window using the **Limit the number of jobs to submit** and **Number of jobs** settings.

When setting the limit values, consider the load placed on the system by the application.

The following figure shows what happens when the total number of jobs that can be executed simultaneously in a queue and the maximum number of jobs that can be submitted to the queue are both set to "5."



6.3 When the Execution Server is Different than the Schedule Server

Jobs that are executed on servers other than the schedule server are referred to as network jobs. ("Servers other than the schedule server" can be any servers that have Systemwalker Operation Manager installed and are connected via the network.)

This section explains the items to be considered when executing job nets or jobs as network jobs.

Items to be considered

Scope of a network job

Network jobs can be specified as individual jobs or as individual job nets (jobs within a job net).

To specify individual jobs, select the **Submit the job as a network job** check box in the **Add/Change - Job** window.

To specify individual job nets, select the **Submit the jobs of the job net as a network job** check box in the **Standard information** tab of the **Job Net Properties** window.

Besides the server name, you can also use a predefined execution subsystem name as the target of the specified network jobs. Refer to "Defining an Execution Subsystem Name" in the *Systemwalker Operation Manager Installation Guide* for details.

Execution servers

Determine which servers will execute network jobs.

For multi-subsystem operations, network job requests can be sent to a subsystem of the local host.

Trust host definitions

It is possible to reject network jobs coming from unexpected servers.

For execution servers, determine which servers they will accept job execution requests from. Immediately after Systemwalker Operation Manager is installed, the default settings are to accept requests from all servers.

Refer to "Defining a Trust Host" in the *Systemwalker Operation Manager Installation Guide* for details.

Code conversion

Consider code conversion when executing network jobs between servers with different character encodings.

To perform code conversion, create definitions in the **Define Operating Information** window on the server from which jobs are submitted (server where Systemwalker Operation Manager has been installed).

Refer to "Definition of Job Execution Control" in the *Systemwalker Operation Manager Installation Guide* for details.

Execution priority

When network jobs are executed between Windows and UNIX platforms, the execution priority will be converted. Ensure that there are no problems with priorities after conversion.

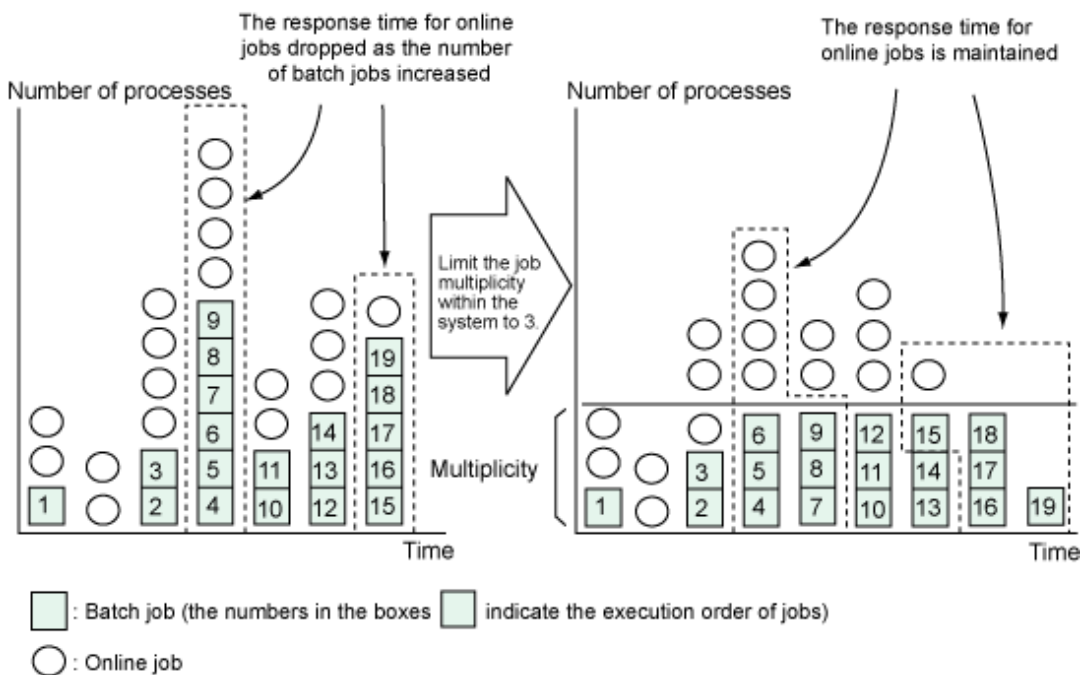
Refer to "Controlling Job Execution Environments" in the *Systemwalker Operation Manager Technical Guide* for details.

6.4 Notes for Adjusting the Job Execution Environment

When automating the business, it may not be possible to perform the action just by configuring the job schedule. In this case, adjust the job execution environment as necessary. In this section, the method used to adjust the job execution environment is explained with examples.

6.4.1 Using "Max Execution Jobs" to Improve Response Times

The response time for online jobs can be improved by limiting the total number of jobs that can be executed simultaneously to restrain the impact of fluctuations in the number of batch jobs submitted.



6.4.2 Executing Jobs with Automatic Load Distribution

The Distributed Execution function can be used to distribute load by automatically allocating job execution requests to the server with the lowest load. With the Distributed Execution function, job execution requests are automatically allocated to the server with the lowest load, out of a list of multiple predefined execution servers.

Chapter 7 Determining the Monitoring Method

Systemwalker Operation Manager can monitor multiple servers as a group, as well as monitor individual connected servers. It is possible to monitor from the Web console or from a Systemwalker Operation Manager client.

Note that Web servers used with the Web console cannot be built in an IPv6 single-stack environment. Therefore, monitor using a Systemwalker Operation Manager client when operating in an IPv6 single-stack environment.

This chapter explains job monitoring.

7.1 Using Systemwalker Operation Manager Clients for Monitoring

Systemwalker Operation Manager clients can monitor information, such as the execution status of job nets being executed on connected servers.

Refer to "Monitoring Jobs" in the *Systemwalker Operation Manager User Guide* for details on monitoring.

Number of clients that can connect to the Systemwalker Operation Manager server

Up to 62 Operation Manager clients can connect to the Operation Manager server (schedule server), but as the number of client connections increases, the response time for each individual client deteriorates. As a best practice, do not connect more than eight Systemwalker Operation Manager clients to a particular Operation Manager server at the same time.

7.1.1 Execution Status Monitoring

The execution status of business operations can be monitored by monitoring job nets, jobs, and queues.

Determine which monitoring method to use, depending on your objective.

Monitoring jobs/job nets

The following table describes the methods for monitoring the execution status of jobs and job nets.

Objective	Monitoring content	Window used
Overall monitoring using the time chart format	Check the status and the execution schedule of all job nets to be started today. Use the time chart format with the current time in the center.	Gantt Chart window
Check details of a particular job net	Check the details of the execution status of a specific job net while it is being monitored using a Gantt chart. For example, check the following information or the precise time in hours, minutes, and seconds: <ul style="list-style-type: none">- List of jobs that constitute the job net- Previous job net execution results (start time and end time)- Next scheduled start time- Message event occurrence etc.	Job List window

Objective	Monitoring content	Window used
Check job nets with specific status values displayed in a list	<p>Check various details, such as the scheduled start time and end time displayed in a list, to monitor those job nets. This method can also be used to:</p> <ul style="list-style-type: none"> - Batch-operate multiple job nets. - Filter the job nets to be displayed. <p>For example, when executing a large number of job nets, you can specify the "Waiting" status to filter the job nets to be displayed and check if all job nets for today have been completed.</p> <ul style="list-style-type: none"> - When using memos for notes on operations or handover information from the previous operator, you can monitor job nets while checking the memo information. 	Job Net Management window
Monitor the job net status using a flow diagram	<p>Check the status of jobs that constitute a job net by using a flow diagram.</p> <p>For example, if a job net terminates abnormally or is delayed, use the flow diagram to check the point (job) where the job net terminated abnormally or which job the job net has progressed to.</p>	Monitor Job Net window

Refer to "Monitoring Jobs/Job Nets" and "Monitoring Scheduled Jobs" in the *Systemwalker Operation Manager User Guide* for details.

Monitoring queues

The following table describes the methods for monitoring queues.

Objective	Monitoring content	Window used
Overall monitoring of queues	Monitor the operational status of all queues on the server, including information, such as the number of jobs executing and the number of jobs waiting in each queue	View Queue Status/Operate window
Check the detailed information about a queue	Check information, such as the detailed information about a queue and the statuses of the jobs in a queue	View Queue Details dialog box

Refer to "Monitoring Queues" in the *Systemwalker Operation Manager User Guide* for details.

7.1.2 Delayed Monitoring

Determine how to conduct delayed monitoring for jobs and job nets.

In either case, in the default specification of the startup parameters for the jobscheduler, messages are output to the event log or the syslog when jobs or job nets are delayed.

Items to be considered

Start delayed monitoring for job nets

If a job net does not start even though its scheduled start time has been reached, a notification message can be output to the event log or the syslog. Additionally, the status of the job net can be verified as "Start delayed" in the Gantt chart. Job nets are subject to scheduled start time monitoring in the following situations:

- If **Scheduled start time** has been defined for a job net under **Start only when message event has occurred** in the **Standard information** tab of the **Job Net Properties** window but the scheduled start time has passed before the message event occurs
- When a job net is waiting for an execution time and message event occurrence linked with a logical AND condition (**Wait for startup time** has been specified in the message conditions), but the message event that the job net is waiting for does not occur by the scheduled start time

Refer to "Monitoring the Start Delayed of the Job Net that is Started at a Message Event Occurrence" in the *Systemwalker Operation Manager User's Guide* for details.

End delayed monitoring for job nets

If a job net does not end even though its scheduled end time has been reached, a notification message can be output to the event log or the syslog. If a job net is still executing even though the estimated end time has elapsed, the Gantt chart can be used to verify its status as "End delayed".

There are two ways of specifying the estimated end time:

- Absolute time
- Relative time (relative to the time when a job net starts)

The relative time is calculated from the time specified for the estimated processing time for a job net.

Specify either of these settings in the **Standard information** tab in the **Job Net Properties** window.

Refer to "Monitoring the End Delayed of the Job Net" in the *Systemwalker Operation Manager User's Guide* for details.

End delayed monitoring for jobs

If the estimated processing time for a job is being monitored (time from when the job starts until it ends) and the job has still not ended even though the estimated processing time has elapsed, a notification message can be output to the event log or the syslog.

Specify this setting in the **Detail information** sheet of the **Add/Change - Job** window.

7.2 Using the Web Console for Monitoring

Monitoring can be performed from the Web console in the same way as a Systemwalker Operation Manager client.

Refer to the *Systemwalker Operation Manager Web Console Operation Guide* for details.

If there are multiple subsystems, the browser response time will slow down as the number of subsystems and job nets being monitored increases.

An estimate for the number of job nets and subsystems with multi-subsystem operations that can be monitored from the Web console is given below.

Items to be considered

Number of job nets and subsystems that can be monitored

- Number of subsystems: As a rough estimate, up to 30 subsystems can be monitored from the Web console.

- Number of job nets: As a rough estimate, up to 1,000 job nets can be monitored from the Web console.

Depending on the server and the client being used, it may not be possible to get full performance from the system. In this case, further limit the number of subsystems and job nets.

To monitor even more systems from the Web console, prepare multiple Web servers and assign each Web server to monitor a different group of subsystems and job nets.

7.3 Using Multi-Server Monitoring

Job nets on multiple Systemwalker Operation Manager servers can be monitored as a batch. Multi-server monitoring can be performed from either a multi-server monitoring client or the Web console.

Multi-server monitoring clients can perform batch operations, and also use criteria to filter job nets on servers subject to monitoring.

To perform multi-server monitoring, determine the monitored objects server and other issues in advance.

Refer to "Monitoring Multiple Servers" in the *Systemwalker Operation Manager User Guide* for details on multi-server monitoring clients.

Items to be considered

Monitored servers

Determine which servers are to be monitored.

Up to 62 servers can be monitored.

Number of connected clients

If the startup parameters for the Jobscheduler limit the number of clients that can be connected, this limit also applies to multi-server monitoring. Determine whether the limit is valid.

When this limit is reached, monitoring cannot be performed by additional clients.

Filtering monitored job nets

Monitoring can also be performed by filtering only related job nets from all the job nets on multiple servers.

If necessary, determine the criteria for filtering which job nets are to be monitored.

Filtering messages in the status message list

The messages output to the status message list in the monitoring window can be filtered.

If necessary, determine the criteria for filtering which displayed messages are to be monitored.

Chapter 8 Determining Contingency Procedures

This chapter explains various measures for minimizing the impact of any faults that may occur.

8.1 Addressing Problems

Systemwalker Operation Manager provides the FJQSS data collection tool, which collects the information required to investigate problems that may occur during operations.

Check the data collection tool in case of any problems that may occur.

FJQSS data collection tool

The FJQSS data collection tool is included in Systemwalker Operation Manager to collect investigation data when problems occur. This tool makes it easy to collect the data required to investigate problems.

Windows

For details on the FJQSS data collection tool, refer to the manuals displayed by selecting **Start >> FJQSS(Information Collection Tool) >> FJQSS User's Guide**.

UNIX

For details on the FJQSS data collection tool, refer to the manuals stored in the Fjqss_Manual folder in Systemwalker Operation Manager product media as shown below:

```
SWOPEREE (SWOPERSE)
|
| ~
|
| ----Fjqss_Manual
|      |---eng
|      |---jpn
```

8.2 Determining Backup and Recovery Methods

Backups prevent situations that may lead to serious problems, such as data losses. For example, the following kinds of problems can occur with any system:

- Damage to hardware due to earthquakes or other natural disasters
- Data loss due to operational errors
- Data loss due to hardware faults

By backing up the necessary data in advance, the impact of these problems can be minimized.

It is necessary to carefully determine what needs to be backed up, and when it should be backed up.

In addition to regular backups, it is recommended to take backups at the following times:

- When definitions or settings are changed
- When operational data is saved

Besides taking backups, ensure that the system can always be recovered quickly and certainly.

The backup and recovery methods available with Systemwalker Operation Manager are explained below.

Backing up the operating environment

Systemwalker Operation Manager provides tools and commands for backing up and restoring the user registration information and the operation management information, in case the operating environment is damaged or accidentally deleted.

Refer to "Backing Up or Restoring Operation Environment" in the *Systemwalker Operation Manager Installation Guide* for details on how to back up and recover the operating environment. Also, refer to

"Storage Location of Definition Information Files" in the *Systemwalker Operation Manager Installation Guide* for details on the information that is backed up.

8.2.1 Notes on Using Backup Products [Windows]

If you are considering using a commercial backup product to back up the following directories and their subdirectories, note that the Systemwalker Operation Manager service must be stopped before backing up these directories.

If the following resources are backed up without stopping the Systemwalker service, Systemwalker Operation Manager may stop running correctly.

- Job Execution Control directory

mpwalker.jm\mpmjessv in the Systemwalker Operation Manager installation directory

- Job Execution Control spool directory

Default: mpwalker.jm\mpmjessv\mjespool in the Systemwalker Operation Manager installation directory

- Database directory for the Jobscheduler

Default: mpwalker.jm\mpjobsch\jobdb in the Systemwalker Operation Manager installation directory

- Calendar information directory

mpwalker.jm\mpjmcac\caldb in the Systemwalker Operation Manager installation directory

EE

- Database directory for the Master Schedule Management function

mpwalker.jm\mpstemsv in the Systemwalker Operation Manager installation directory

If the system also includes Systemwalker Centric Manager, it is necessary to stop the Systemwalker Centric Manager service as well.

8.3 Determining Redundancy

One way of addressing faults is to use redundancy for the system, the definition information, and other resources.

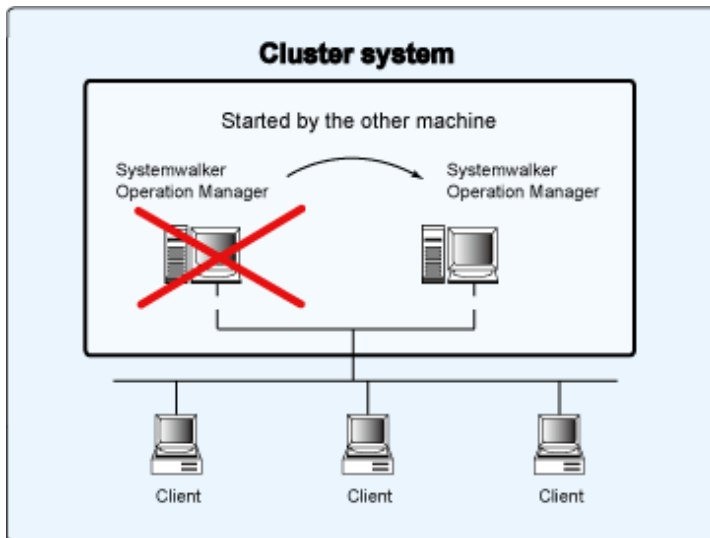
By making the system, the definition information, and other resources redundant in case of unexpected problems, the impact of problems can be minimized, in the same way as with backups.

The redundancy methods supported by Systemwalker Operation Manager are explained below.

Cluster system

Systemwalker Operation Manager supports cluster systems.

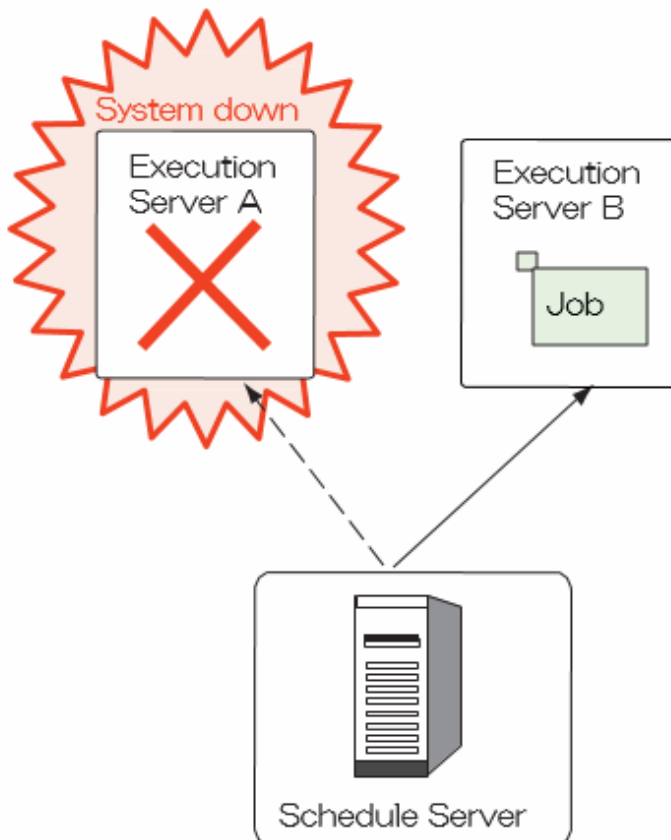
Cluster systems involve operating multiple linked servers as though they are a single server. In this way, even if a fault occurs on one server, processing can be taken over by another server, making it possible to achieve higher availability than a system that is operated using stand-alone servers.



Refer to the *Systemwalker Operation Manager Cluster Setup Guide for Windows* and the *Systemwalker Operation Manager Cluster Setup Guide for UNIX* for details on cluster systems.

Execution server redundancy

Redundancy can be applied to execution servers in case an execution server for a network job fails.



By specifying two candidate execution servers (Execution server A as the first candidate and Execution server B as the second candidate), job execution requests will be automatically sent to the second candidate (Execution server B) if the first candidate (Execution server A) fails. Refer to "[8.4.2 How to Continue Business Operations when Problems Occur with Execution Servers \(Execution Server Redundancy for Network Jobs\)](#)" for details.

Another method for automatically executing jobs on another server when a server fails involves using the Distributed Execution function. The Distributed Execution function is originally used as a function that groups multiple execution servers and automatically allocates jobs to the execution server with the lowest load. However, this function can also be used for execution server redundancy, because if one of the execution servers in a group fails, then jobs will be automatically allocated to the execution server that has the lowest load in the group.

Each method has its advantages and disadvantages, so refer to the following table when considering these methods.

	Advantage	Disadvantage
Execution server redundancy	Jobs are normally executed on the first candidate server. Jobs are executed on the second candidate server only if the first candidate server fails. In other words, this method enables jobs to be executed on substitute servers only when a fault occurs.	If the first candidate server fails, it can take some time for jobs to be submitted, because execution does not switch to the second candidate server until job submission to the first candidate server has been attempted.
Distributed Execution	Up to 100 servers can be set up as distribution destinations. If a particular server fails, it is possible to recognize that the server has failed for up to 10 minutes, during which jobs will not be submitted to that server (no attempt is made to submit jobs to the server that has failed).	It is impossible to predict on which server a job will be executed.

Schedule information file redundancy

By using the `jmmode` command to enable data to be written to backup files, it is possible to duplicate the files containing schedule information (known as "schedule information files").

If redundancy has been specified, synchronous writes will be performed on the schedule information files and the backup files. Synchronous writes make it less likely that inconsistencies will occur with the schedule information files. However, the schedule performance and the startup performance for groups, job nets, and jobs will decline. Before adopting file redundancy, conduct thorough performance testing to ensure that job nets start on schedule without any problems. Even if redundancy has been specified, inconsistencies may occur with the schedule information files, for example, if power shutdown occurs while the operating system is writing data to the disk. Consider taking regular backups in anticipation of such circumstances.

If redundancy is not specified, the schedule performance and the startup performance will not decline. However, there is a high possibility that inconsistencies will occur with the schedule information files due to unscheduled power shutdown or some other reason, so consider backing up the schedule information files regularly and restoring them when faults occur.

Refer to "jmmode Continuous Execution Mode Switching Command" in the *Systemwalker Operation Manager Reference Guide* for details on the `jmmode` command.

8.4 Determining Business Continuity when Faults Occur

From the perspective of system operations, it is important to minimize the impact of business interruptions and stoppages even when faults occur.

It is possible to minimize the impact of faults if you adopt measures to enable business operations to continue as much as possible even when faults occur.

The following sections describe effective measures for addressing business continuity.

8.4.1 How to Continue Business Operations when the Schedule Server Fails (Enabling Continuous Execution Mode)

To continue network jobs that are executing even when the schedule server fails, consider enabling the continuous execution mode. (A "network job" is a job that is executing on a server other than the schedule server.)

The continuous execution mode is disabled when Systemwalker Operation Manager is first installed, so any network jobs that are executing will be canceled if the schedule server fails.

If the continuous execution mode is enabled, network jobs that are executing will continue to execute on the execution server rather than being canceled, even if the schedule server fails.

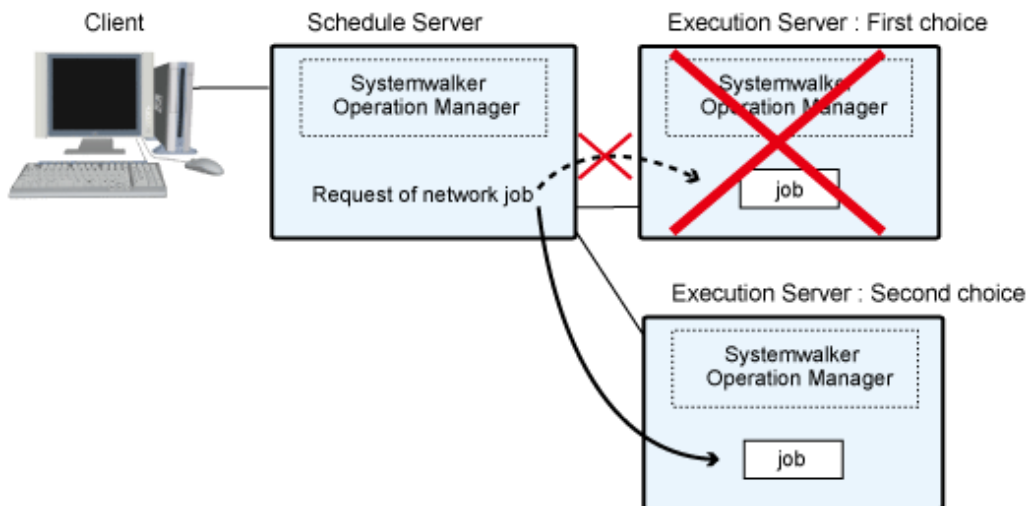
After the schedule server has been restarted, the status of the jobs that have continued will be automatically reflected to the schedule information file. Also, if the network job has any succeeding jobs, the job net will be restarted automatically so the business operation continues.

Refer to "Continuing Job Operations at Schedule Server System Down" in the *Systemwalker Operation Manager User Guide* for details on the definition procedure.

8.4.2 How to Continue Business Operations when Problems Occur with Execution Servers (Execution Server Redundancy for Network Jobs)

To continue business operations even if the execution server for network jobs becomes unusable due to problems, such as system failure or communication disruptions, consider adopting redundancy for execution servers.

Redundancy for execution servers is a mechanism where the first and second candidate execution servers are defined in advance, enabling job execution requests to be automatically sent to the second candidate execution server when problems occur.



The execution server definitions can be created for each job net or each job.

Refer to "Duplicating the Network Job Execution Server" in the *Systemwalker Operation Manager User Guide* for details on the definition procedure.

Chapter 9 Items for Consideration

This chapter explains the items to be considered, as required.

9.1 Startup Schedules for Services and Applications

To automatically start different services and applications on different days when the operating system starts, determine the startup schedules for services and applications.

Services and applications will be started according to the startup schedules that are set up. (For UNIX, only applications will be started.)

Items to be considered

Startup patterns for services and applications

The following definition patterns can be created:

- Standard schedules
- Holiday schedules
- User-specific schedules (up to seven schedules)

Determine which services and applications to start, the order in which they should start, and which calendar to apply.

When the server stops, services and applications must be stopped in the reverse order compared to the order in which they were started.

Refer to "Scheduling Services and Applications" in the *Systemwalker Operation Manager User Guide* for details on the startup schedules for services and applications.

9.2 Event Monitoring and Action Control [Windows]

To automatically perform actions (such as "sending a short mail to the system administrator" or "starting an application to deal with an event") without human intervention when events occur (such as a message notifying a system error), consider using Event Monitoring and Action Control.

The actions that are automatically executed can be changed, depending on the days and times, such as weekdays, weekends, holidays, and timeslots.

If the system also includes Systemwalker Centric Manager, use the Event Monitoring and Action Control functions provided by Systemwalker Centric Manager.

The following items must be considered when Event Monitoring and Action Control are performed.

Items to be considered

Monitored events

Consider which events to monitor. Up to 1,024 events can be defined as monitored events.

The following table shows the types of events that can be monitored.

Event type	Description
Event log	The messages output to the event log files.
Monitored log file	The added parts of text output to log files. Systemwalker Operation Manager can monitor products that log error events to custom text log files and other files. When text is added to the log files being monitored, this is treated as if an event has occurred.

Event type	Description
Monitored message	The messages handled by Systemwalker Centric Manager (System Monitor). These are the relayed messages transferred from the system monitoring agent located at a lower level of the logical hierarchy than the definition destination system. These messages can be monitored if Systemwalker Centric Manager has been installed.

Also, determine the conditions for identifying particular events (such as the log, source name, event ID, type, and category).

Automatic actions

Determine which actions to execute automatically when events occur.

The following table shows the automatic actions provided by Event Monitoring.

Automatic action	Description
Audio notification	Notifies the administrator of an event by reading the message aloud, playing a WAV file or playing a beep sound
E-mail notification	Notifies the administrator of an event by sending any specified document via email
Popup message notification	Notifies the administrator of an event by displaying any specified message as the popup message
Short mail notification	Notifies the administrator of an event by sending any specified message as a short mail
SNMP trap notification	Issues SNMP traps
Application start	Starts any specified application
Event log output	Outputs any specified message to the event log

Also, determine the execution conditions for actions (dates and times when they will be executed).

Refer to "Monitoring Events and Managing Actions [Windows version]" in the *Systemwalker Operation Manager User Guide* for details on Event Monitoring and Action Control.

9.3 Audit Logs

The records of operations performed by Systemwalker Operation Manager can be saved as audit logs. This makes it possible to check who performed what kind of operation, as well as where and when the operation was performed.

By default, Systemwalker Operation Manager is set up to output audit logs when Systemwalker Operation Manager is installed. Audit logs will be stored on the specified output destination on a Systemwalker Operation Manager server for only the specified number of days. Operations on clients are also output to the audit log files on the server.

To change the default settings (such as canceling audit log output, or changing the output directory and the retention period for audit logs), refer to "Defining Audit Log Output" in the *Systemwalker Operation Manager Installation Guide*.

Appendix A Conventions Used in This Document and Trademarks

A.1 Abbreviations and Generic Terms Used

- The term "Windows Server 2022" is used to refer to all of the following products:
 - Microsoft(R) Windows Server(R) 2022 Standard (x64)
 - Microsoft(R) Windows Server(R) 2022 Datacenter (x64)
- The term "Windows Server 2019" is used to refer to all of the following products:
 - Microsoft(R) Windows Server(R) 2019 Standard (x64)
 - Microsoft(R) Windows Server(R) 2019 Datacenter (x64)
- The term "Windows Server 2016" is used to refer to all of the following products:
 - Microsoft(R) Windows Server(R) 2016 Standard (x64)
 - Microsoft(R) Windows Server(R) 2016 Datacenter (x64)
- The term "Server Core" is used to refer to all of the following products:
 - Microsoft(R) Windows Server(R) 2022 Standard Server Core
 - Microsoft(R) Windows Server(R) 2022 Datacenter Server Core
 - Microsoft(R) Windows Server(R) 2019 Standard Server Core
 - Microsoft(R) Windows Server(R) 2019 Datacenter Server Core
 - Microsoft(R) Windows Server(R) 2016 Standard Server Core
 - Microsoft(R) Windows Server(R) 2016 Datacenter Server Core
- The term "Windows(R) 10" is used to refer to all of the following products:
 - Windows(R) 10 Home (x64)
 - Windows(R) 10 Pro (x64)
 - Windows(R) 10 Enterprise (x64)
- Windows Internet Explorer(R) is abbreviated as "Internet Explorer".
- Versions of Systemwalker Operation Manager that run on all of the following operating systems are referred to as "Windows versions of Systemwalker Operation Manager" or simply "Windows versions":
 - Windows
 - 64-bit versions of Windows, except Itanium
- Articles specific to the version of Systemwalker Operation Manager that runs on 32-bit versions of Windows are referred to as "Windows x86 version".
- Articles specific to the version of Systemwalker Operation Manager that runs on Itanium-compatible versions of Windows are referred to as "Windows for Itanium version".
- Articles specific to the version of Systemwalker Operation Manager that runs on 64-bit versions of Windows, except Itanium, are referred to as "Windows x64 version".
- Server Core, Windows Server 2022, Windows Server 2019, and Windows Server 2016 may be abbreviated as "Windows servers".
- Oracle Solaris may be referred to as Solaris, Solaris Operating System or Solaris OS.

- Versions of Systemwalker Operation Manager that run on Solaris are referred to as "Solaris versions of Systemwalker Operation Manager" or simply "Solaris versions".
- Articles specific to the version of Systemwalker Operation Manager that runs on 32-bit versions of Solaris are referred to as "Solaris 32-bit version".
- Articles specific to the version of Systemwalker Operation Manager that runs on 64-bit versions of Solaris are referred to as "Solaris 64-bit version".
- Versions of Systemwalker Operation Manager that run on HP-UX are referred to as "HP-UX versions of Systemwalker Operation Manager" or simply "HP-UX versions".
- Versions of Systemwalker Operation Manager that run on AIX are referred to as "AIX versions of Systemwalker Operation Manager" or simply "AIX versions".
- Articles specific to the version of Systemwalker Operation Manager that runs on 64-bit versions of Linux, except Itanium, are referred to as "Linux x64 version" or simply "Linux versions".
- Articles specific to the version of Systemwalker Operation Manager that runs on 32-bit versions of Linux are referred to as "Linux x86 version".
- Articles specific to the version of Systemwalker Operation Manager that runs on Itanium-compatible version of Linux are referred to as "Linux for Itanium version".
- Solaris, HP-UX, AIX, Linux and Linux for Itanium versions of Systemwalker Operation Manager are referred to collectively as "UNIX versions of Systemwalker Operation Manager" or simply "UNIX versions".
- Solaris, HP-UX, AIX and Linux may be referred to as "UNIX servers".
- Systemwalker Operation Manager Standard Edition may be abbreviated as "SE".
- Systemwalker Operation Manager Enterprise Edition may be abbreviated as "EE".
- Standard Edition may be abbreviated as "SE" and Enterprise Edition may be abbreviated as "EE".
- Arcserve(R) Backup for Windows is abbreviated as "Arcserve".
- Microsoft(R)-Mail that is provided as a standard feature with Windows NT(R) is abbreviated as "MS-Mail".

A.2 Trademarks

Amazon Web Services, Amazon Elastic Compute Cloud, Amazon CloudWatch and AWS Lambda are trademarks of Amazon.com, Inc. or its affiliated company in the United States and/or other countries.

APC and PowerChute are trademarks or registered trademarks of Schneider Electric IT Corporation.

All products, service name, company name and logo of Arcserve are registered trademark or trademark of Arcserve (USA), LLC.

HP-UX is a registered trademark of Hewlett-Packard Development Company in the United States and/or other countries.

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

Intel and Itanium are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

MC/ServiceGuard is a registered trademark of Hewlett Packard Enterprise Development LP in the United States and other countries.

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

Red Hat are registered trademarks of Red Hat, Inc. in the U.S. and other countries.

Linux(R) is a registered trademark of Linus Torvalds in the United States and/or other countries.

Oracle, Java, and GlassFish are registered trademarks of Oracle Corporation or its subsidiaries in the U.S. and/or other countries.

R/3, SAP and all SAP trade names mentioned are trademarks or registered trademarks of SAP AG in Germany and other countries.

UNIX is a registered trademark of The Open Group.

VMware, the VMware logo are trademarks or registered trademarks of VMware, Inc. in the United States and/or other countries.

Short Mail is a registered trademark of NTT DoCoMo, Inc.

In addition, company name and product name may appear in this document are trademarks or registered trademarks of their respective owners.

The use of screenshots follows the guidelines of Microsoft Corporation.