# Hitachi Application Server V10 User's Guide (For UNIX® Systems)

## Overview, User's Guide and Operator's Guide

3021-3-415-10(E)

# Notices

## ■ Relevant program products

Applicable OSs: Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64), Red Hat Enterprise Linux 5 (AMD/Intel 64), Red Hat Enterprise Linux Server 6 (64-bit x86_64)

P-9W43-5KA2 Hitachi Application Server 10-10

Note that OSs other than those listed above in "Applicable OSs" might become usable. For details, see the *Release Notes*.

## ■ Export restrictions

If you export this product, please check all restrictions (for example, Japan's Foreign Exchange and Foreign Trade Law, and USA export control laws and regulations), and carry out all required procedures.

If you require more information or clarification, please contact your Hitachi sales representative.

## ■ Trademarks

Adobe and Flash Player are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

AMD, AMD Opteron, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

Apache is either a registered trademark or a trademark of the Apache Software Foundation in the United States and/or other countries.

CORBA is a registered trademark of Object Management Group, Inc. in the United States.

Firefox is a registered trademark of the Mozilla Foundation.

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

IIOP is a trademark of Object Management Group, Inc. in the United States.

Intel is a trademark of Intel Corporation in the U.S. and/or other countries.

Kerberos is a name of network authentication protocol created by Massachusetts Institute of Technology.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft, Excel, Hyper-V, Internet Explorer, SQL Server, Visual C++, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Mozilla is a trademark of the Mozilla Foundation in the U.S and other countries.

OMG, CORBA, IIOP, UML, Unified Modeling Language, MDA and Model Driven Architecture are either registered trademarks or trademarks of Object Management Group, Inc. in the United States and/or other countries.

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

RC4 is either a registered trademark or a trademark of EMC Corporation in the United States and/or other countries.

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

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

RSA SECURED®

SOAP is an XML-based protocol for sending messages and making remote procedure calls in a distributed environment.

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

VMware, VMware vSphere ESX and VMware vSphere ESXi are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

W3C is a trademark (registered in numerous countries) of the World Wide Web Consortium.

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

This product includes RSA BSAFE® Cryptographic software of EMC Corporation.

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

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

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

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

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

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

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

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

This product includes software developed by Andy Clark.

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

HITACHI
Inspire the Next
Hitachi, Ltd.

Java™ COMPATIBLE ENTERPRISE EDITION

Java™ COMPATIBLE

## ■ Issued

Mar. 2015: 3021-3-415-10(E)

## ■ Copyright

# Preface

## ■ Purpose of this manual

This manual describes the procedures mapped to each process based on the use cases in accordance with the operation process, system development, and functional overview of Application Server. By using this manual, users will be able to understand the functional overview and product architecture, and use the system, from the installation to system operations.

## ■ Intended readers

The intended readers of this manual are as follows:

- For systems engineer

The prerequisite is as follows:

- For systems engineer
  - Information related to the operations of Windows or UNIX.
  - Information related to the Application Server development.
  - Information related to the Java EE Standard specifications.
  - Information related to the peripheral environment (such as database, network, and job management, and so on) used in the system development.

## ■ List of related manuals

The following table shows the titles of related manuals, and related manuals used in this manual.

Hitachi Application Server manuals

- UNIX manuals

| Abbreviation | Formal name | Reference number |
|---|---|---|
| *User's Guide* | *Hitachi Application Server V10 User's Guide* (For UNIX® Systems) | 3021-3-415(E) |
| *GUI Reference* | *Hitachi Application Server V10 GUI Reference Guide* (For UNIX® Systems) | 3021-3-417(E) |
| *Command Reference* | *Hitachi Application Server V10 Command Reference Guide* (For UNIX® Systems) | 3021-3-419(E) |
| *Definition Reference* | *Hitachi Application Server V10 Definition Reference Guide* (For UNIX® Systems) | 3021-3-421(E) |
| *Messages* | *Hitachi Application Server V10 Messages* | 3021-3-422(E) |
| *API Reference* | *Hitachi Application Server V10 API Reference Guide* | 3021-3-423(E) |

## ■ Abbreviations for products and functions

This manual uses the following abbreviations for product names and function name:

| Abbreviations | | Product name and function name |
|---|---|---|
| Application Server | | Hitachi Application Server |
| Application Server - Base | | Hitachi Application Server - Base |
| Application Server - Optional License for Java | | Hitachi Application Server - Optional License for Java |
| Application Server for Developers | | Hitachi Application Server for Developers |
| APV | | IBM Advanced POWER Virtualization |
| DAS | | Domain Administration Server |
| domain administration server | | |
| Developer's Kit for Java | | Hitachi Developer's Kit for Java |
| Excel | | Microsoft® Excel |
| Firefox | | Firefox® |
| HiRDB | HiRDB Version 9 | HiRDB Server Version 9 |
| | HiRDB/Single Server | HiRDB/Single Server Version 9 |
| HWS | | Hitachi Web Server |
| Web Server | | |
| Internet Explorer | | Windows® Internet Explorer® |
| Java EE Server | | Hitachi Java EE Server |
| JP1/AJS3 | | Job Management Partner 1 Version 10 Job Management Partner 1/Automatic Job Management System 3 - Agent |
| | | Job Management Partner 1 Version 10 Job Management Partner 1/Automatic Job Management System 3 - Manager |
| | | Job Management Partner 1 Version 10 Job Management Partner 1/Automatic Job Management System 3 - View |
| JP1/IM | | Job Management Partner 1 Version 10 Job Management Partner 1/Integrated Management - Manager |
| | | Job Management Partner 1 Version 10 Job Management Partner 1/Integrated Management - View |
| Microsoft IIS | Microsoft IIS 7.5 | Microsoft® Internet Information Services 7.5 |
| | Microsoft IIS 8.0 | Microsoft® Internet Information Services 8.0 |
| | Microsoft IIS 8.5 | Microsoft® Internet Information Services 8.5 |
| Microsoft Visual C++ | | Microsoft® Visual C++® |
| Oracle | Oracle 11g | Oracle Database 11g |
| | | Oracle Database 11g R2 |
| | Oracle 12c | Oracle Database 12c |
| performance tracer | | Performance Tracer |
| SQL Server | | Microsoft® SQL Server |
| UNIX | AIX | AIX V6.1 |

| Abbreviations | | | Product name and function name |
|---|---|---|---|
| | | | AIX V7.1 |
| | Linux | Linux (x86/AMD64 & Intel EM64T) | Red Hat Enterprise Linux® 5 Advanced Platform (AMD/Intel 64) |
| | | | Red Hat Enterprise Linux® 5 (AMD/Intel 64) |
| | | | Red Hat Enterprise Linux® Server 6 (64-bit x86_64) |
| Virtage | | | Hitachi Virtage |
| VMware ESX | | | VMware vSphere ESX |
| VMware vSphere ESXi | | | VMware vSphere ESXi |
| Windows | Windows Server 2008 R2 | | Microsoft® Windows Server® 2008 R2 Standard |
| | | | Microsoft® Windows Server® 2008 R2 Enterprise |
| | | | Microsoft® Windows Server® 2008 R2 Datacenter |
| | Windows Server 2012 | | Microsoft® Windows Server® 2012 Standard |
| | | | Microsoft® Windows Server® 2012 Datacenter |
| | Windows Server 2012 R2 | | Microsoft® Windows Server® 2012 R2 Standard |
| | | | Microsoft® Windows Server® 2012 R2 Datacenter |
| | Windows 7 | Windows 7 x86 | Microsoft® Windows® 7 Professional (32 bit) |
| | | | Microsoft® Windows® 7 Enterprise (32 bit) |
| | | | Microsoft® Windows® 7 Ultimate (32 bit) |
| | | Windows 7 x64 | Microsoft® Windows® 7 Professional (64 bit) |
| | | | Microsoft® Windows® 7 Enterprise (64 bit) |
| | | | Microsoft® Windows® 7 Ultimate (64 bit) |
| | Windows 8 | Windows 8 x86 | Windows® 8 Pro (32 bit) |
| | | | Windows® 8 Enterprise (32 bit) |
| | | Windows 8 x64 | Windows® 8 Pro (64 bit) |
| | | | Windows® 8 Enterprise (64 bit) |
| | | Windows 8.1 x86 | Windows® 8.1 Pro (32 bit) |
| | | | Windows® 8.1 Enterprise (32 bit) |
| | | Windows 8.1 x64 | Windows® 8.1 Pro (64 bit) |
| | | | Windows® 8.1 Enterprise (64 bit) |
| Windows Server Failover Cluster | | | Windows Server® Failover Cluster |
| Class-wise statistics | | | Hitachi Class-wise statistics |

For each version of Linux, the following abbreviations might be used.

| Abbreviation | | OS |
|---|---|---|
| Red Hat Enterprise Linux 5 | Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) | Red Hat Enterprise Linux® 5 Advanced Platform (AMD/Intel 64) |
| | Red Hat Enterprise Linux 5 (AMD/Intel 64) | Red Hat Enterprise Linux® 5 (AMD/Intel 64) |
| Red Hat Enterprise Linux Server 6 | Red Hat Enterprise Linux Server 6 (64-bit x86_64) | Red Hat Enterprise Linux® Server 6 (64-bit x86_64) |

## ■ Conventions: Acronyms

This manual also uses the following acronyms:

| Acronym | Full name or meaning |
|---|---|
| ACC | Application Client Cotainer |
| ACL | Access Control List |
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| ASCII | American Standard Code for Information Interchange |
| BLOB | Binary Large Object |
| CA | Certificate Authority |
| CDI | Contexts and Dependency Injection |
| CGI | Common Gateway Interface |
| CMP | Container-Managed Persistence |
| CMT | Container-Managed Transaction |
| CopyGC | Copy Garbage Collection |
| CORBA | Common Object Request Broker Architecture |
| | CORBA® |
| CPU | Central Processing Unit |
| CRL | Certificate Revocation List |
| CSR | Certificate Signing Request |
| CSV | Comma Separated Value |
| CVS | Concurrent Versions System |
| DBMS | Database Management System |
| DCOM | Distributed Component Object Model |
| DD | Deployment Descriptor |
| DDE | Dynamic Data Exchange |
| DER | Distinguished Encoding Rules |
| DES | Data Encryption Standard |

| Acronym | Full name or meaning |
|---|---|
| DI | Dependency Injection |
| DLL | Dynamic Link Library |
| DMZ | Demilitarized Zone |
| DN | Distinguished Name |
| DNS | Domain Name System |
| DoS | Denial of Service |
| DSO | Dynamic Shared Object |
| DTD | Document Type Definition |
| EAR | Enterprise Archive |
| ear | |
| EIS | Enterprise Information System |
| EJB | Enterprise JavaBeans™ |
| EJB QL | EJB™ Query Language |
| EL | Expression Language |
| EUC | Extended UNIX Code |
| FQDN | Fully Qualified Domain Name |
| FullGC | Full Garbage Collection |
| G1GC | Garbage First Garbage Collection |
| GC | Garbage Collection |
| GMS | Group Management Service |
| GMT | Greenwich Mean Time |
| GUI | Graphical User Interface |
| HA | High Availability |
| HTML | Hyper Text Markup Language |
| HTTP | Hyper Text Transfer Protocol |
| HTTPS | Hyper Text Transfer Protocol Security |
| I/O | Input/Output |
| ID | Identifier |
| IDE | Integrated Development Environment |
| IEC | International Electrotechnical Commission |
| IIOP | IIOP™ |
| | Internet Inter-Orb Protocol |
| IIS | Internet Information Services |

| Acronym | Full name or meaning |
|---|---|
| IMAP | Internet Message Access Protocol |
| IP | Internet Protocol |
| IPv6 | Internet Protocol Version 6 |
| ISO | International Organization for Standardization |
| J2EE | J2EE™ |
|  | Java™ 2 Platform, Enterprise Edition |
| JAAS | Java™ Authentication and Authorization Service |
| JACC | Java™ Authorization Service Provider Contract for Containers |
| JAF | JavaBeans™ Activation Framework Specification |
| JAR | Java™ Archive |
| jar |  |
| JASPIC | Java™ Authentication Service Provider Interface for Containers |
| Java | Java™ |
| Java EE | Java™ Platform, Enterprise Edition |
| Java EE RI | Java EE Reference Implementation |
| Java HotSpot Client VM | Java HotSpot™ Client Virtual Machine |
| Java Platform Debugger Architecture | Java™ Platform Debugger Architecture |
| JPDA |  |
| Java SE | Java™ Platform, Standard Edition |
| Java VM | Java™ Virtual Machine |
| JVM |  |
| JavaMail | JavaMail™ |
| JAX-RPC | Java™ API for XML-based RPC |
| JAX-RS | Java™ API for RESTful Web Services |
| JAX-WS | Java™ API for XML-based Web Services |
| JAXB | Java™ Architecture for XML Binding |
| JAXP | Java™ API for XML Processing |
| JAXR | Java™ API for XML Registries |
| JCA | J2EE™ Connector Architecture |
| JDBC | Java™ Database Connectivity |
|  | JDBC™ |
| JDK | Java™ Development Kit |
|  | JDK™ |

| Acronym | Full name or meaning |
|---|---|
| JIS | Japanese Industrial Standards |
| JMS | Java™ Message Service |
| JMX | Java™ Management Extensions |
| JNDI | Java Naming and Directory Interface™ |
| JNI | Java™ Native Interface |
| JPA | Java™ Persistence API |
| JSF | JavaServer™ Faces |
| | JavaServer™ Faces Reference Implementation (RI) Version: 1.1_01 FCS |
| JSON-P | Java™ API for JSON Processing |
| JSP | JavaServer Pages™ |
| | JSP™ |
| JST | Japan Standard Time |
| JSTL | JavaServer Pages™ Standard Tag Library |
| JTA | Java™ Transaction API |
| JVMPI | Java™ Virtual Machine Profiler Interface |
| JVMTI | Java™ Virtual Machine Tool Interface |
| KVM | Kernel-based Virtual Machine |
| LAN | Local Area Network |
| LDAP | Lightweight Directory Access Protocol |
| MAC | Message Authentication Code |
| MIME | Multipurpose Internet Mail Extensions |
| OASIS | Organization for the Advancement of Structured Information Standards |
| OMG | Object Management Group |
| ORB | Object Request Broker |
| OS | Operating System |
| OTS | Object Transaction Service |
| QNAME | Qualified Name |
| REST | Representational State Transfer |
| RMI | Remote Method Invocation |
| RPC | Remote Procedure Call |
| RSA | Rivest, Shamir and Adleman |
| SAAJ | SOAP with Attachments API for Java™ |
| SAX | Simple API for XML |

| Acronym | Full name or meaning |
|---|---|
| SEI | Service Endpoint Interface |
| Servlet | Java™ Servlet |
| SHA | Secure Hash Algorithm |
| SMAP | Source Map |
| SMTP | Simple Mail Transfer Protocol |
| SOAP | Simple Object Access Protocol |
| SSH | Secure Shell |
| ssh | |
| SSL | Secure Sockets Layer |
| StAX | Streaming API for XML |
| TCP | Transmission Control Protocol |
| TLD | Tag Library Descriptor |
| TLS | Transport Layer Security |
| UCS | Universal multi-octet coded Character Set |
| UDP | User Datagram Protocol |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| URN | Uniform Resource Name |
| UTC | Coordinated Universal Time |
| UTF | UCS Transformation Format |
| UTF-8 | 8-bit UCS Transformation Format |
| VM | Virtual Machine |
| VTL | Velocity Template Language |
| W3C | World Wide Web Consortium |
| WAR | Web Archive |
| war | |
| WBEM | Web-Based Enterprise Management |
| WSDL | Web Services Description Language |
| XML | Extensible Markup Language |

## ■ Conventions: KB, MB, GB, TB, and PB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.

- 1 MB (megabyte) is $1{,}024^2$ bytes.
- 1 GB (gigabyte) is $1{,}024^3$ bytes.
- 1 TB (terabyte) is $1{,}024^4$ bytes.
- 1 PB (petabyte) is $1{,}024^5$ bytes.

# Contents

## 4 Configuring an application execution environment 97

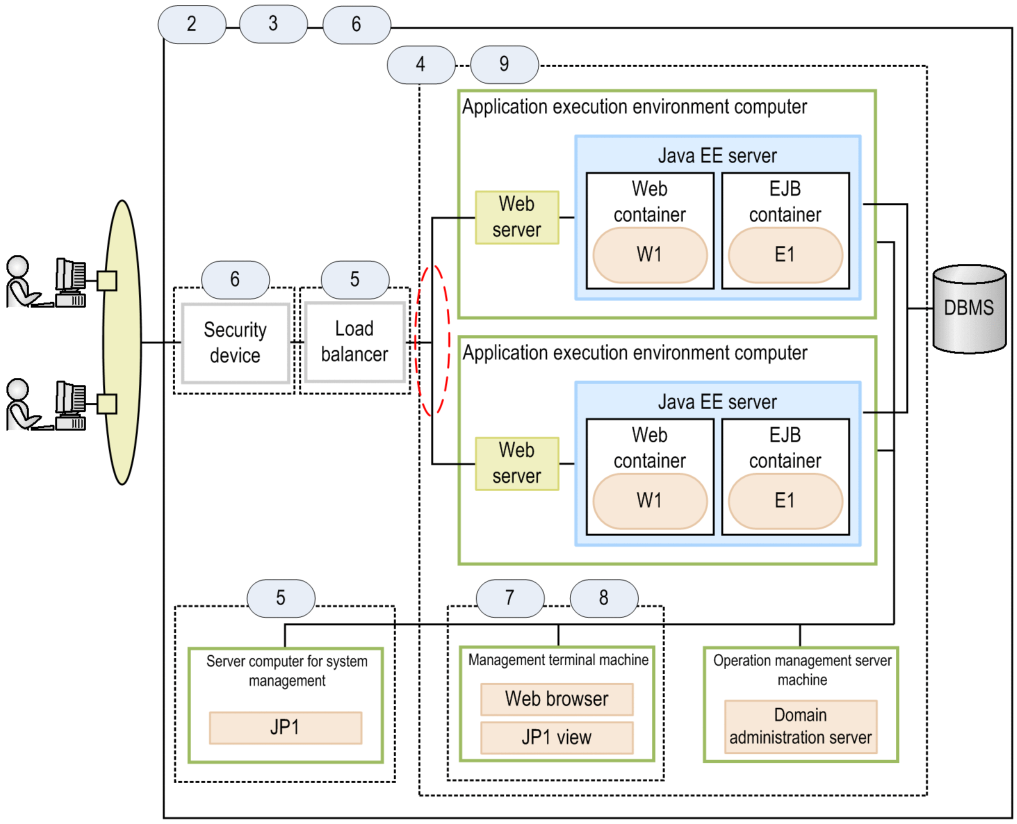
# 1 How to read this manual

This chapter describes the location of the information related to the process that is required to develop and operate a system using Application Server.

# 1.1 How to read the User's Guide

This section provides information that you should refer to while developing web front-end systems that use Application Server. References to every process from the development of a web front-end system to its operation are provided. This section also provides an outline of each chapter. Note that the environment where the application is executed is called the application execution environment, and the environment where the application is developed is called the application development environment.

The following figure shows the configuration of the web front system that uses Application Server, which is explained in this manual:


2
3
6
4
9
Application execution environment computer
Java EE server
Web server
Web container
W1
EJB container
E1
6
5
Security device
Load balancer
DBMS
Application execution environment computer
Java EE server
Web server
Web container
W1
EJB container
E1
5
7
8
Server computer for system management
JP1
Management terminal machine
Web browser
JP1 view
Operation management server machine
Domain administration server

Legend:

◌ : Request distribution

▭ : Application

▨ : Process or program

n : Chapter number and scope of description in the manual

Chapters to be referenced during the design, creation, and operation of the application execution environment

The following table shows the operations to be performed during the design, creation, and operation of the application execution environment, and provides a summary of the relevant chapter to be referenced. Refer to the relevant chapter based on your requirements. Note that the operations to be performed during the creation and operation of the system can be performed by using commands or by using Administration Console.

| Process | Content to be implemented | Number in the diagram | Chapter in the manual | Summary of the corresponding chapter |
|---|---|---|---|---|
| Designing the system | Configuring a system and implementing the system design | 2 | Overview of Application Server | This chapter provides the information that is required to examine the system configuration.<br>Examine the system configuration based on information such as Application Server product information, system configuration examples for each purpose, and process configuration information. |
| | | 3 | Application Server design items | This chapter provides the information that is required to design a system.<br>To ensure system reliability and performance, a system is designed by selecting the design items if required. |
| Configuring a system | Creating the application execution environment, and configuring settings for the functions to be used. | 4 | Configuring an application execution environment | This chapter provides the procedure for creating the application execution environment.<br>There are two types of application execution environments: a system where a single Java EE server is allocated, and a cluster configuration where multiple Java EE servers are allocated to distribute the load. Note that a cluster configuration is a configuration for grouping and managing multiple Java EE servers. |
| | | 5 | Specifying system environment settings | This chapter provides information about the items to be set up in the surrounding system environment. The items to be set up are the network, load-balancing, and automation operation. |
| | | 6 | Settings for achieving high reliability | This chapter provides information on how to achieve high reliability by setting up functionality related to error detection (process monitoring and message monitoring) and by specifying settings to strengthen security (reverse proxy and SSL). |
| Operating the system | Performing tasks while maintenance is being performed or while operating normally | 7 | Tasks involved in normal operation | This chapter provides information about the daily tasks that begin after the operations start, and during normal operations.<br>For system operation, the following tasks can be performed for Application Server:<br>• Starting or stopping the system<br>• Starting or stopping of the system at the same time as the starting or stopping of the computer<br>• Checking the execution status of Application Server<br>• Checking the execution status of the application<br>• Checking the connection status of the database |
| | | 8 | About the maintenance operations | This chapter provides information about the tasks completed during maintenance in order to handle changes in the activity status or system configuration.<br>For system maintenance, the following tasks can be performed for Application Server: |

| Process | Content to be implemented | Number in the diagram | Chapter in the manual | Summary of the corresponding chapter |
|---|---|---|---|---|
| | | | | • Changing the environment definition based on the execution status and the system configuration<br>• Changing the IP address and the host name<br>• Replacing the application<br>• Backing up and restoring environment information<br>• Applying revision patches and other revisions<br>• Checking the usage status and operating status of the system<br>• Scaling out the system<br>• Upgrading |
| | Understanding the content of information that is output for troubleshooting | 9 | Using troubleshooting data | This chapter provides information about troubleshooting information, such as the logs and trace logs that are output to the application execution environment and to the application development environment. Examine troubleshooting information based on the explanation of the output troubleshooting information and the notes on obtaining the trace logs of the performance tracer (trace collection point). |

### Important note

This manual explains how to design, create, and operate web front systems for systems that have a single main path. This manual does not describe systems other than web front systems. Furthermore, this manual does not describe all of the design items for system design, all of the steps and values to be set for system creation, or all of the operational procedures to be performed for system operation.

# 2 Overview of Application Server

This chapter presents information required to consider the configuration of a system that uses Application Server. This chapter examines the system configuration based on information about the Application Server products, the system configuration examples for each purpose, the process configurations, and other items.

# 2.1 Standard specifications supported by Application Server

Application Server complies with standard specifications related to Java EE 7, Java SE, Web Service, CORBA, and the Internet. This section explains the standard specifications supported by Application Server.

## Java EE 7 specifications

| No. | Standard specifications |
|---|---|
| 1 | Java EE 7 |
| 2 | WebSocket 1.0[#1] |
| 3 | JSON-P 1.0 |
| 4 | Servlet 3.1 |
| 5 | JSF 2.2 |
| 6 | EL 3.0 |
| 7 | JSP 2.3 |
| 8 | JSTL 1.2 |
| 9 | Batch 1.0 |
| 10 | Concurrency Utilities 1.0 |
| 11 | CDI 1.1 |
| 12 | DI 1.0 |
| 13 | Bean Validation 1.1 |
| 14 | Interceptors 1.2 |
| 15 | JCA 1.7 |
| 16 | JPA 2.1 |
| 17 | Common Annotations for the Java Platform 1.2 |
| 18 | JMS 2.0 |
| 19 | JTA 1.2 |
| 20 | JavaMail 1.5 |
| 21 | JAX-RS 2.0 |
| 22 | Web Services for Java EE 1.4 |
| 23 | JAX-WS 2.2 |
| 24 | Web Services Metadata for the Java Platform 2.1 |
| 25 | JASPIC 1.1 |
| 26 | JACC 1.5 |
| 27 | Java EE Management 1.1 |
| 28 | Debugging Support for Other Languages 1.0 |
| 29 | Managed Beans 1.0 |
| 30 | EJB 3.2[#2] |

| No. | Standard specifications |
|---|---|
| 31 | JAX-RPC 1.1 |
| 32 | JAXR 1.0 |
| 33 | Java EE Deployment 1.2 |

#1:

This specification is supported when Microsoft IIS 8.0 or later is used as an HTTP server.

#2:

CMP and EJB QL are not supported.

## Java SE specifications

- Java SE 8

## Java SE specifications related to Java EE7 specifications

| No. | Standard specifications |
|---|---|
| 1 | JAXB 2.2 |
| 2 | JAXP 1.3 |
| 3 | JMX 1.2 |
| 4 | JAF 1.1 |
| 5 | StAX 1.0 |
| 6 | SAAJ 1.3 |
| 7 | JDBC 4.1 |

## Web Service specifications

| No. | Standard specifications | |
|---|---|---|
| 1 | SOAP (W3C standards) | SOAP 1.1 |
| 2 | | SOAP 1.2 |
| 3 | WSDL (W3C standards) | WSDL 1.1 |
| 4 | Bootstrapping | WS-MetadataExchange v1.1 |
| 5 | Policy (W3C standards) | WS-Policy v1.2 |
| 6 | | WS-PolicyAttachment v1.2 |
| 7 | | WS-Policy v1.5 |
| 8 | | WS-PolicyAttachment v1.5 |
| 9 | Reliable Messaging (OASIS standards) | WS-ReliableMessaging v1.0 |
| 10 | | WS-ReliableMessaging Policy v1.0 |
| 11 | | WS-ReliableMessaging v1.1 |
| 12 | | WS-ReliableMessaging Policy v1.1 |
| 13 | | WS-ReliableMessaging v1.2 |

| No. | Standard specifications | |
|---|---|---|
| 14 | | WS-ReliableMessaging Policy v1.2 |
| 15 | | WS-MakeConnection v1.1 |
| 16 | Atomic Transactions (OASIS standards)# | WS-AtomicTransaction v1.0 |
| 17 | | WS-Coordination v1.0 |
| 18 | Security (OASIS standards) | WS-Security v1.0 |
| 19 | | WS-Security v1.1 |
| 20 | | WS-SecurityPolicy v1.1 |
| 21 | | WS-SecurityPolicy v1.2 |
| 22 | | WS-Trust v1.2 |
| 23 | | WS-Trust v1.3 |
| 24 | | WS-Trust v1.4 |
| 25 | | WS-SecureConversation v1.2 |
| 26 | | WS-SecureConversation v1.3 |
| 27 | | WS-SecureConversation v1.4 |
| 28 | Security Profiles (OASIS standard) | Web Services Security: SOAP Message Security V1.0 |
| 29 | | WS-Security Core Specification 1.1 |
| 30 | | Username Token Profile V1.0 |
| 31 | | Username Token Profile 1.1 |
| 32 | | X.509 Token Profile V1.0 |
| 33 | | X.509 Token Profile 1.1 |
| 34 | | SAML Token Profile V1.0 |
| 35 | | SAML Token profile 1.1 |
| 36 | | Kerberos Token Profile 1.1 |
| 37 | | Rights Expression Language (REL) Token Profile 1.1 |
| 38 | | SOAP with Attachments (SWA) Profile 1.1 |
| 39 | Addressing (W3C standard) | Web Services Addressing 1.0 - Core |
| 40 | | Web Services Addressing 1.0 - SOAP Binding |
| 41 | | Web Services Addressing 1.0 - WSDL Binding |
| 42 | WS-I Profile | WS-I Basic Profile 1.2 |
| 43 | | WS-I Basic Profile 2.0 |
| 44 | | WS-I Attachments Profile 1.0 |
| 45 | | WS-I Simple SOAP Binding Profile 1.0 |

#:

Metro supports the pre-standardization version of items 16 and 17.

## CORBA specifications

- Common Object Request Broker Architecture (CORBA), Version 3.0

## Standard specifications related to the Internet

| No. | Standard specifications |
|---|---|
| 1 | IPv6 |
| 2 | HTTP 1.0 |
| 3 | HTTP 1.1 |
| 4 | SSL v3 |
| 5 | TLS 1.0 |
| 6 | TLS 1.1 |
| 7 | TLS 1.2 |

## 2.2 Product configuration, prerequisite software programs, and related products for Application Server

This section explains the product configuration and prerequisite OSs for Application Server. This section also presents information about the products required to use Application Server, such as supported virtualization platforms, DBMSs, web browsers, HTTP servers, mail servers, and related products.

### Product configuration

There are the following Application Server products:

Application Server

This product creates an environment in which applications are executed (application execution environment).

Application Server for Developers

This product creates an environment in which applications to be executed on Application Server are developed (application development environment). This product is only provided for Windows.

### Prerequisite OSs

Application Server can only be used under the following OSs:

Prerequisite OSs for Application Server

| OS versions |
|---|
| Red Hat Enterprise Linux 5<br>Red Hat Enterprise Linux Server 6 |

### Supported character encodings

Application Server supports the following character encodings:

| OS | Character encoding |
|---|---|
| Linux | UTF-8 |

### Virtualization platforms

Application Server supports the following hypervisors for virtualization platforms:

| Hypervisor | Supported guest OS |
|---|---|
| Virtage | Linux |
| KVM | Linux |
| VMware ESX<br>• VMware ESX 4.0<br>• VMware ESX 4.1<br>• VMware vSphere ESXi 5 | Linux |

### DBMSs

Application Server can connect to the following DBMSs:

| DBMS | JDBC driver |
|---|---|
| Oracle 11g | Oracle JDBC Drivers 12.1.0.1.0 or later |
| Oracle 12c | Oracle JDBC Drivers 12.1.0.1.0 or later, or Oracle JDBC Drivers 12.1.0.1.0 or later |

## Web browsers

Application Server can be set up and operated from the web browsers shown in the following table.

Note that the Administration Console cannot be used with UNIX.

| Web browser | Required software program |
|---|---|
| Internet Explorer 7 or later | Adobe Flash Player 11.8 or later |
| Firefox 34.0 or later<br>Firefox ESR 31 or later | |

## HTTP servers

Application Server can connect to the following HTTP servers:

| HTTP server | Version |
|---|---|
| Web Server | HTTP server bundled with Application Server |

## Mail servers

Application Server supports the following mail servers:

| Mail server | Purpose |
|---|---|
| SMTP server, IMAP server | To use JavaMail |

## Related products

Application Server can operate the system in conjunction with the following products:

| Product name | Purpose |
|---|---|
| JP1/IM | To monitor the entire system for faults |
| JP1/AJS3 | To automate system operations |
| HA monitor | To operate cluster systems |

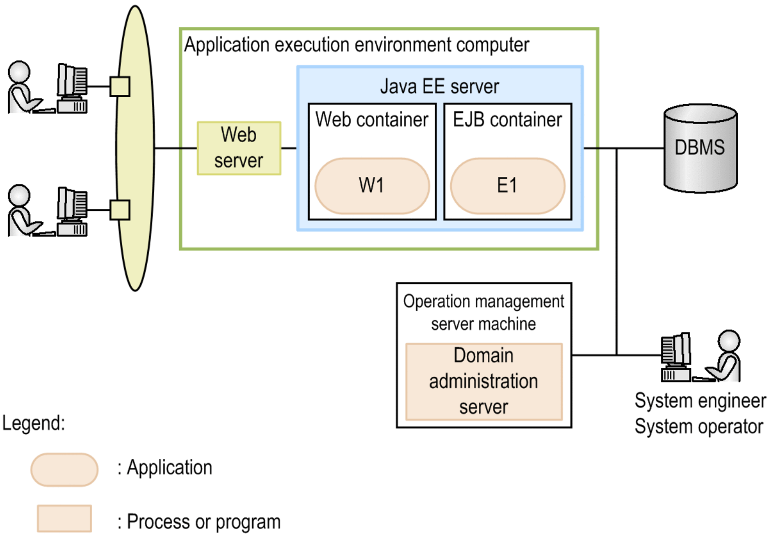
## 2.3 System configurations for Application Server

Using Application Server enables you to construct a system with a configuration that meets the requirements for intended jobs and applications to be executed. You can create a variety of system configurations, such as minimum configurations where Application Server and a web server are located on the same machine, and configurations that ensure availability and performance. You can also create configurations that ensure security, configurations that ensure safe operations, configurations that allow linkage to other systems, and multi-tenant configurations.

### 2.3.1 Minimum system configurations

This subsection explains a minimum configuration where a Java EE server and a web server are located on the same machine.

In this configuration, a Java EE server and a web server are located on the same machine. This configuration is suitable for systems with relatively few transactions.

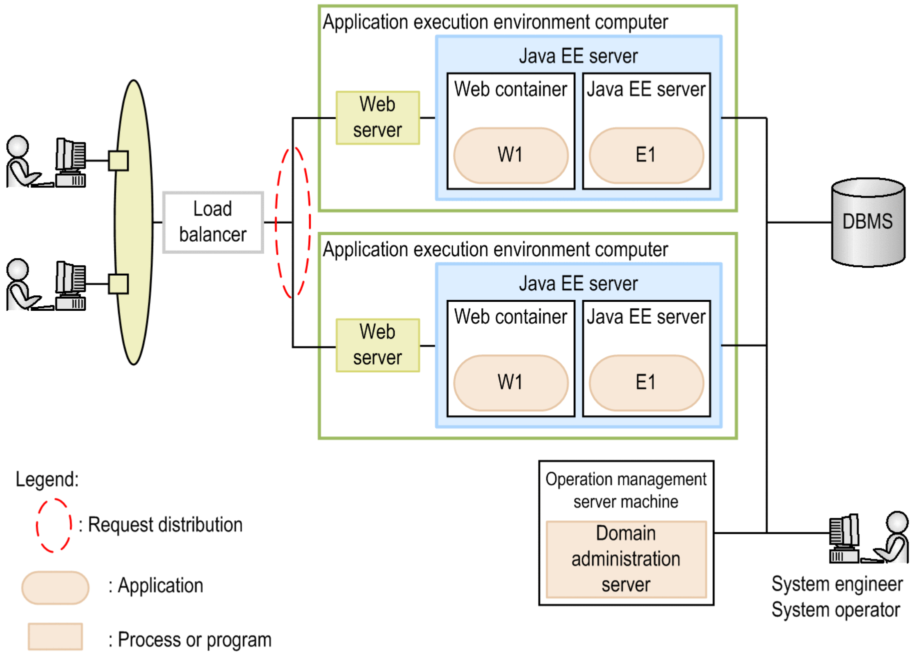


### 2.3.2 System configurations that ensure availability and performance

This subsection explains system configurations that use Application Server to ensure availability and performance. As typical examples of such system configurations, this subsection describes the characteristics of the following configurations: 1) configuration that uses a hardware load balancer, 2) configuration that uses a generic software load balancer, 3) configuration that uses the Web Server load balancing function, and 4) configuration that uses cluster software.

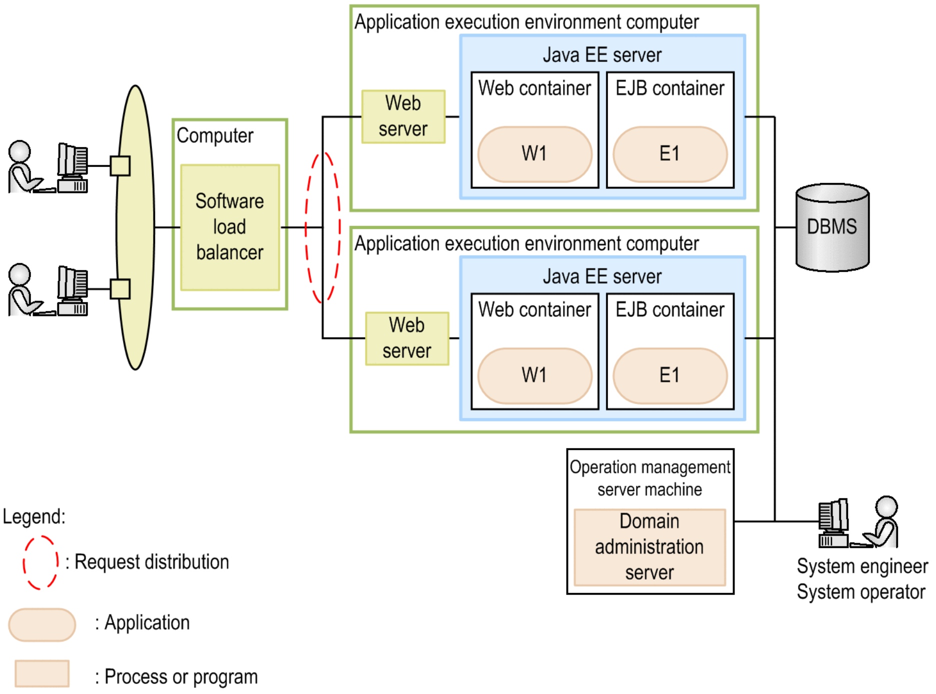
Configuration that use a hardware load balancer

> This type of configuration uses a hardware load balancer to distribute requests evenly to multiple Java EE servers. This type is suitable for mission-critical systems with a large number of transactions. This type can ensure high reliability and availability.



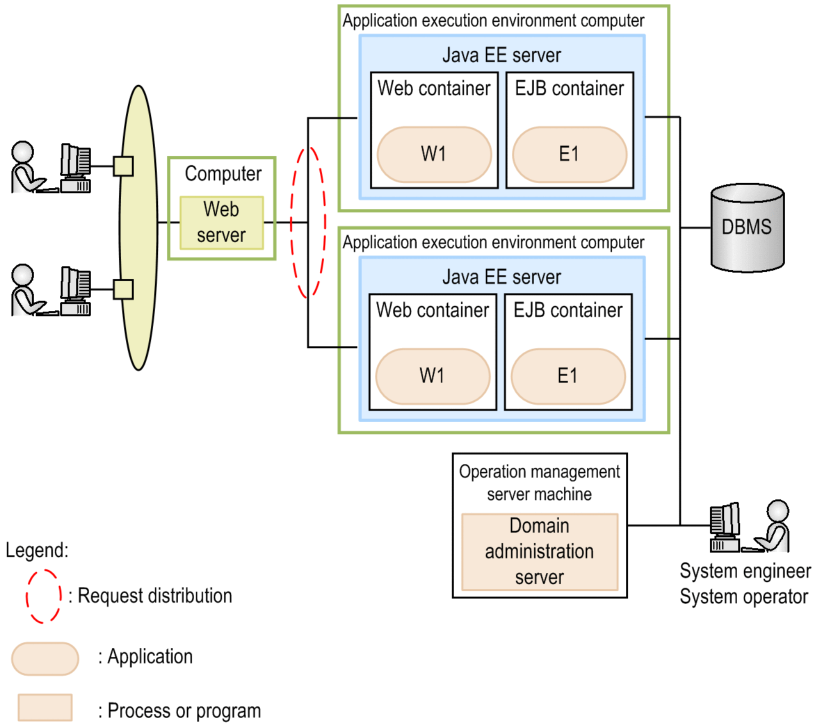
Configuration that uses a generic software load balancer

This type of configuration uses a generic software load balancer to distribute requests evenly to multiple Java EE servers. This type is suitable for systems with a large number of transactions. This type ensures reliability and availability without using expensive load balancers.



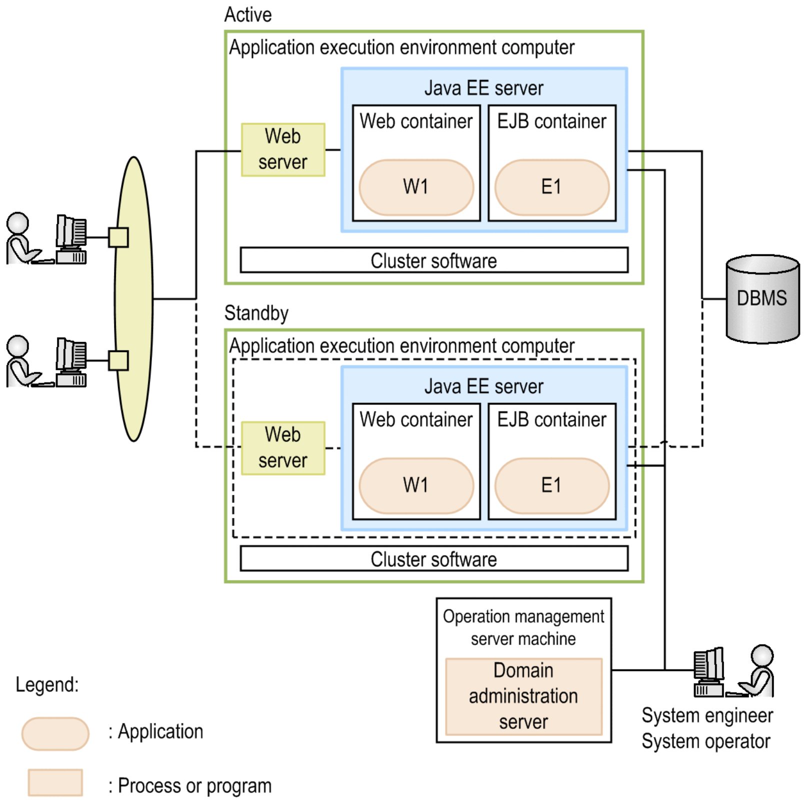
Configuration that uses the Web Server load balancing function

This type of configuration uses the Web Server load balancing function to distribute requests evenly to multiple Java EE servers. This type is suitable for systems with a large number of transactions. Using the load balancing function of Web Server can ensure reliability and availability at low cost.



Configuration that uses cluster software

This type of configuration uses cluster software to create a cluster configuration where multiple Java EE servers are installed on active and standby nodes. This type is suitable for systems with few transactions where only a single machine can process requests. This type of configuration prepares a standby node in case of a fault occurrence, and this node can also be used effectively as a development or test environment during normal operations.

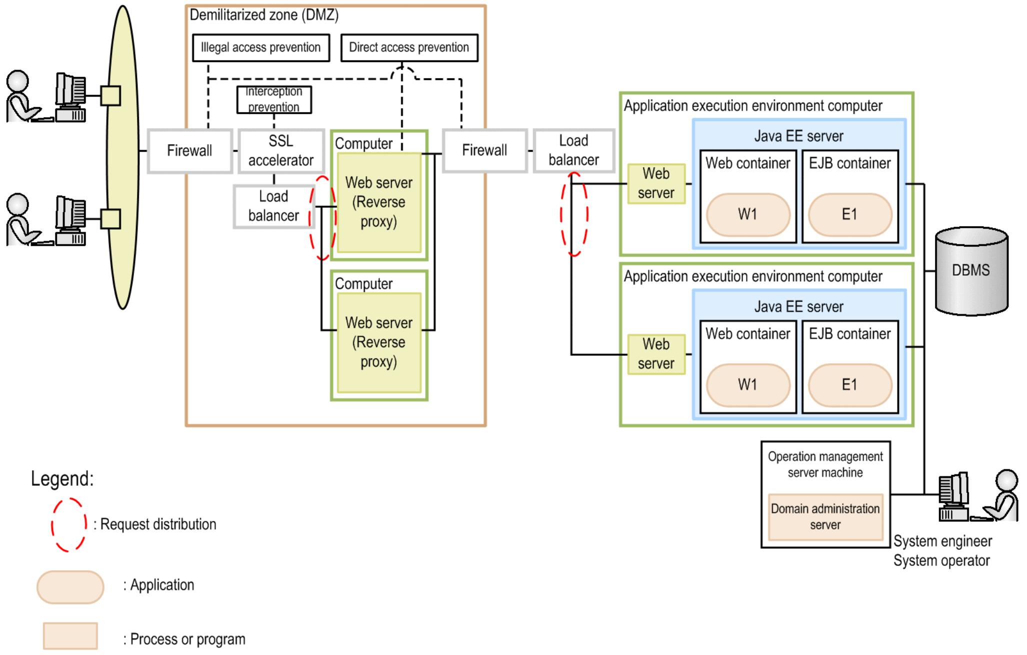


## 2.3.3 Configurations that ensure security

This subsection explains system configurations that use Application Server to ensure security. As typical examples of such system configurations, this subsection describes the characteristics of a configuration that uses reverse proxies and a configuration that does not use reverse proxies.

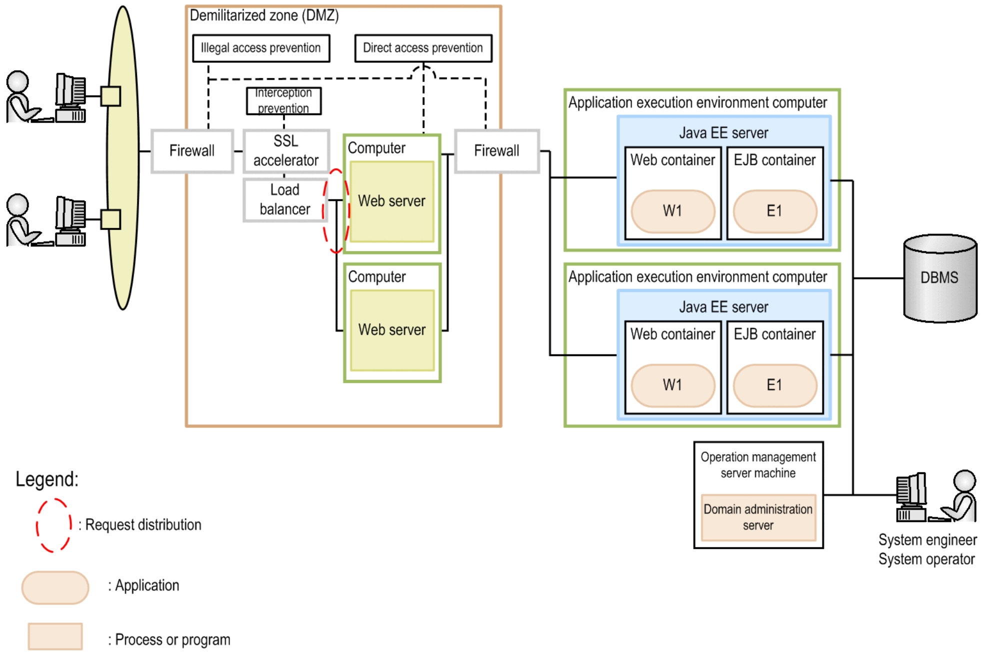
Configuration that uses reverse proxies

This type of configuration places reverse proxy servers inside the DMZ, and web servers and Java EE servers in the internal network in order to ensure security. This type prevents illegal accesses, eavesdropping, and DoS attacks. Multiple Java EE servers ensure availability.



Configuration that does not use reverse proxies

This type of configuration places web servers inside the DMZ, and Java EE servers in the internal network in order to ensure security. This type uses no reverse proxies in order to reduce communication overhead. This type prevents illegal accesses, eavesdropping, and DoS attacks. Multiple Java EE servers ensure availability.

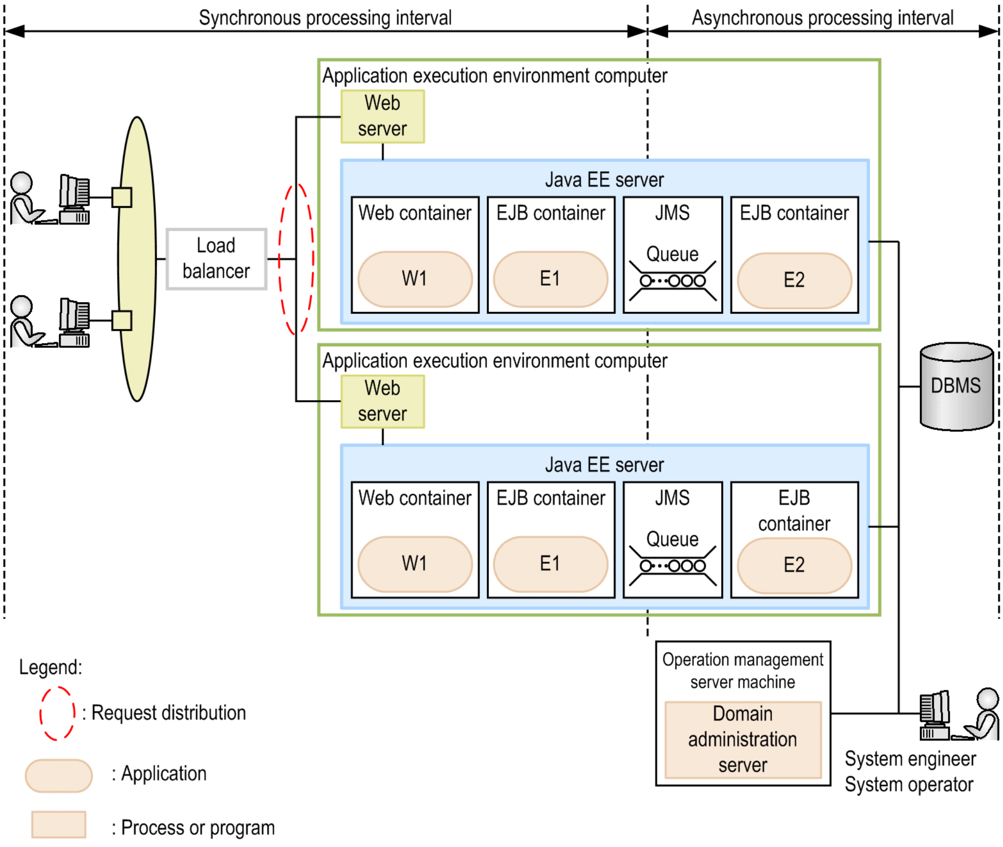


## 2.3.4 Configurations that ensure stable operations

This subsection explains system configurations in which Application Server can operate stably. As typical examples of such system configurations, this subsection describes the characteristics of a configuration that asynchronously performs online batch processing.

Configuration for asynchronously performing online batch processing

This type of configuration temporarily places high-load processes in a JMS queue, and asynchronously executes them by controlling the number of concurrently executed processes. This type can ensure stable operations because increase of concurrently executed processes can be prevented in such cases where many requests are sent in a short time, and execution of a certain process takes a long time.

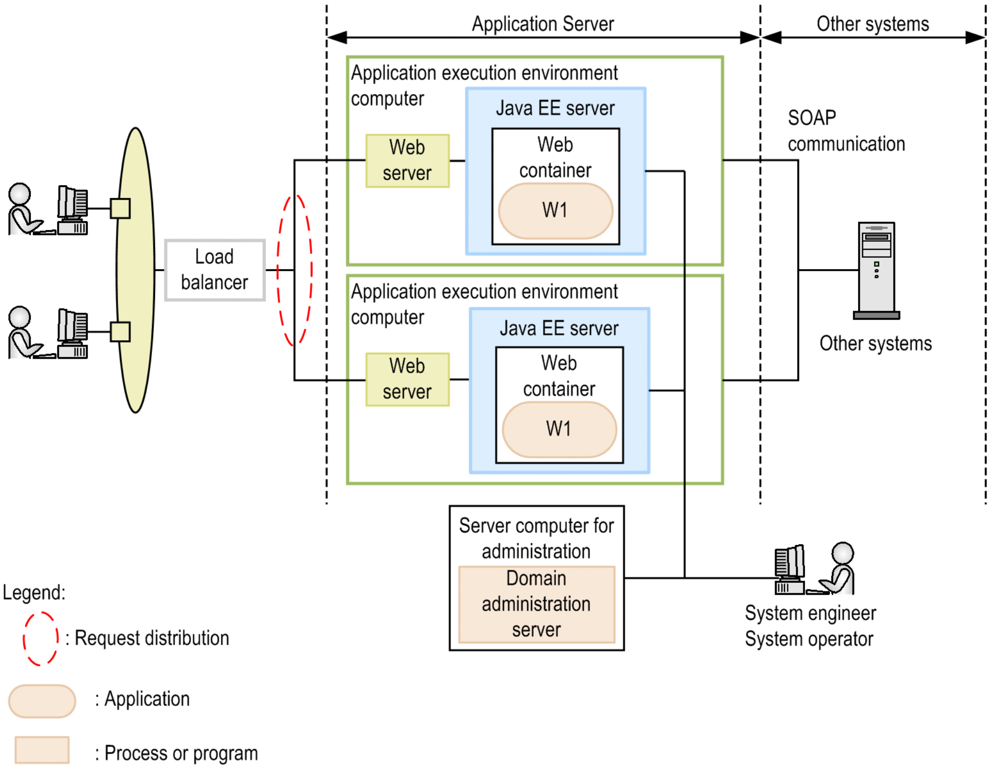


## 2.3.5 Configurations that allow linkage to other systems

This subsection explains system configurations in which Application Server links to other systems. As typical examples of such system configurations, this subsection describes the characteristics of the following configurations: 1) configuration where the back end connects to other systems, 2) configuration where the front end connects to other systems, and 3) configuration that uses WebSphere MQ to connect to other systems.

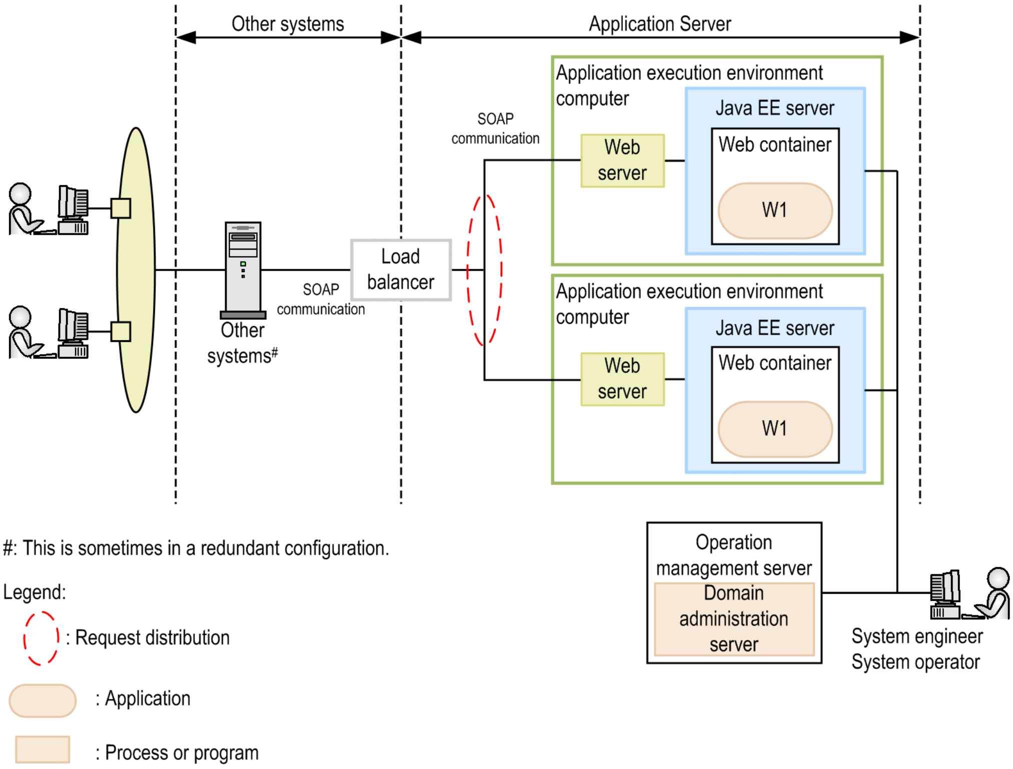
Configuration where the back end connects to other systems

In this configuration, Application Server at the front end receives requests, and accesses business systems at the back end. Application Server can be loosely coupled with existing systems by using interfaces such as SOAP, CORBA, and REST.



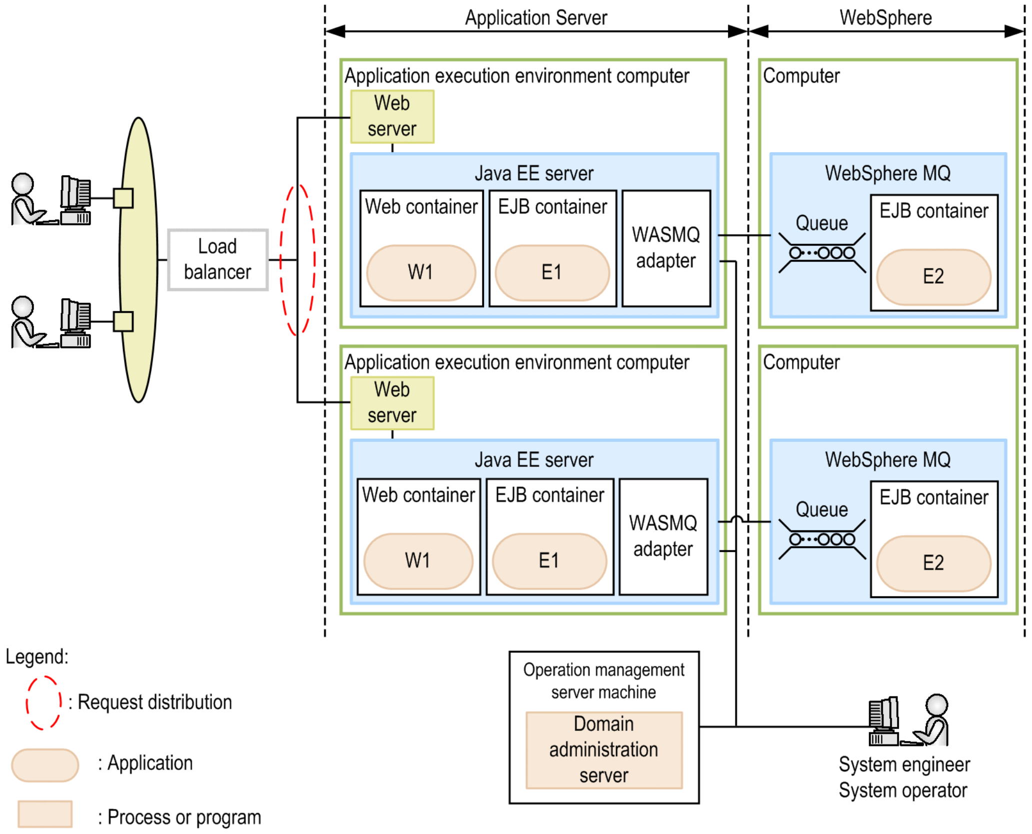
Configuration where the front end connects to other systems

In this configuration, upon receiving requests from existing systems, Application Server at the back end executes jobs. Application Server can be loosely coupled with existing systems by using interfaces such as SOAP and REST.



Configuration that uses WebSphere MQ to connect to other systems

In this configuration, WebSphere MQ resource adapters are deployed on Application Server to connect to WebSphere MQ. Application Server can link to existing systems that have the WebSphere MQ interface.

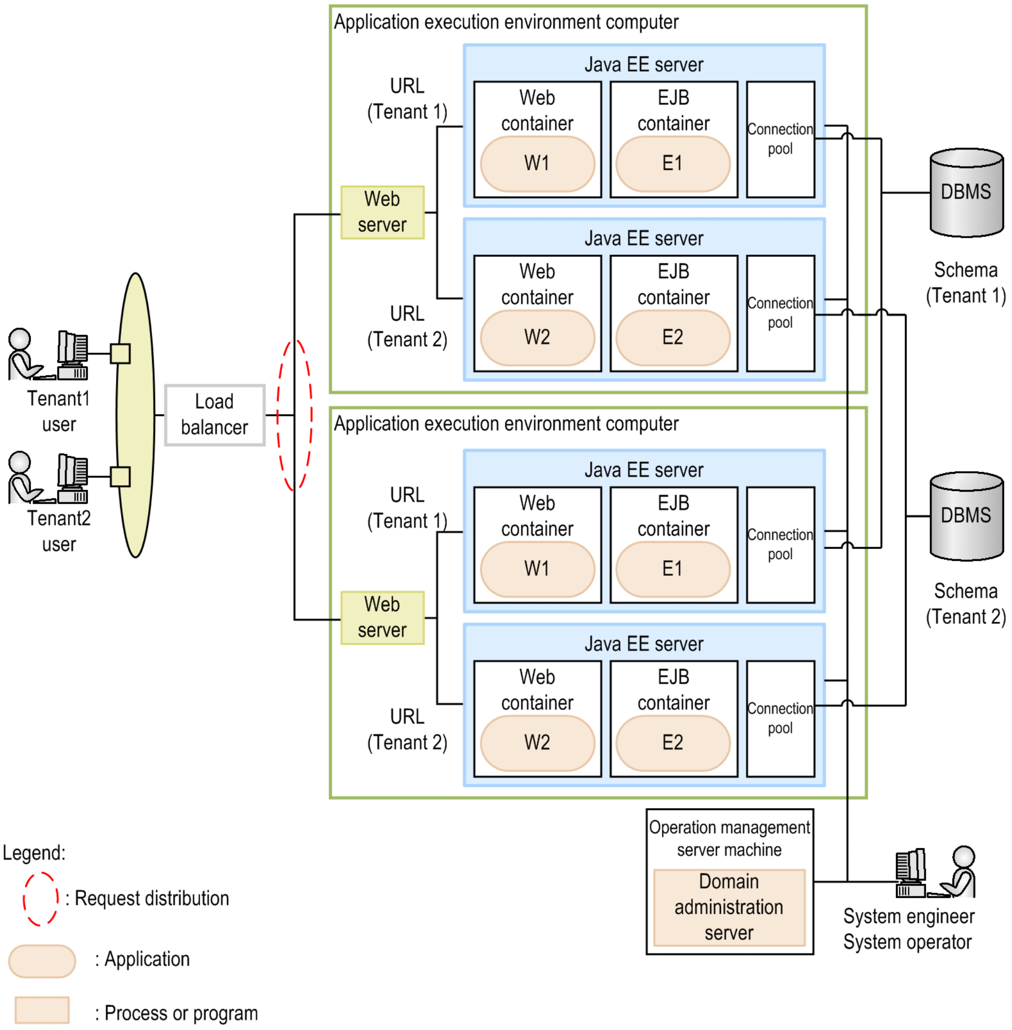


## 2.3.6 Multi-tenant configurations

This subsection explains multi-tenant configurations that use Application Server. As typical examples of such system configurations, this subsection describes the characteristics of a configuration in which a Java EE server is deployed for each tenant.

Configuration in which a Java EE server is deployed for each tenant

In this configuration, a Java EE server (at the back of a web server) is deployed for each tenant. The users of each tenant access the system through a different URL. The web server distributes requests to Java EE processes according to the URL. Because individual tenants use separate Java EE servers, faults such as process failures occurring in a tenant do not affect other tenants.

Application execution environment computer
Java EE server
URL
(Tenant 1)
Web container
W1
EJB container
E1
Connection pool
Web server
URL
(Tenant 2)
W2
E2
DBMS
Schema
(Tenant 1)
Schema
(Tenant 2)
Tenant1 user
Tenant2 user
Load balancer
Operation management server machine
Domain administration server
System engineer
System operator
Legend:
: Request distribution
: Application
: Process or program

# 2.4 Management elements and process configurations for Application Server

Systems that use Application Server are managed by using elements such as domains, nodes, servers, clusters, configurations, and dependency relations between servers. Domains are managed by using a domain administration server. Process configurations required for Application Server to operate include web servers, Java EE servers, performance tracers, and domain administration servers.

## Management elements for Application Server

For systems that use Application Server, system components are managed by units called *domains*. A single domain administration server resides on each domain to manage the domain. A domain (managed by a domain administration server) consists of management elements called nodes and servers. Servers are managed by elements called clusters, dependency relations, and configurations.

Domain

A domain is an element that manages multiple servers making up a business system. A domain can also manage servers (Java EE servers, web servers, and performance tracers) on different machines. Domains are managed by using domain administration servers.

Domain administration server

A domain administration server is a server instance that is prepared specifically for managing a domain. A domain administration server exists for the domain and is created with the name `server`. We recommend that you create a domain administration server on an operation management server machine that is not the application execution environment machine.

Node

A node is an element that defines a destination host (machine) within a domain. A server is created on a host defined by a node. Multiple nodes can be defined within a domain. Multiple domains cannot share a single node.

Server

An instantiated element of various servers managed by a domain. This element represents a server entity and contains server configuration information. Server elements include Java EE servers, web servers, and performance tracers. Every server belongs to a single domain.

| **Server type** | **Overview** |
|---|---|
| Java EE server | A container server that runs Java EE applications. |
| Web server | A service program that provides web content. Use Web Server. |
| Performance tracer | A software program that monitors the operating performance of servers such as Java EE servers and web servers. This program is located on the same node as the server to be monitored. |

Cluster

A cluster is a group element of Java EE servers that share the same applications, resources, and configuration information. A cluster can group and manage Java EE servers that are located on different machines. A system configuration that uses a cluster to group multiple Java EE servers is called a cluster configuration. A cluster is used to ensure scalability, load balancing, and failover.

Dependency relation

An instantiated element that holds the configuration information required to correlate two servers. For servers to work with each other, this element automatically sets information about the depended-on destination server on the dependent source server. For example, this element automatically sets the listening port number of the depended-on destination server as a connection port parameter on the dependent source server. Dependency relations are classified into redirection dependency relations and performance tracer dependency relations.

| Relation type | Overview |
|---|---|
| Redirection dependency relation | This relation sets up the redirection destination Java EE servers for the requests received by web servers. It enables you to set up relations between servers on different nodes.<br>Source server: Web server<br>Destination server: Java EE server or cluster |
| Performance tracer dependency relation | This relation specifies settings that enable performance tracers to collect performance analysis traces in order to check server operating performance during operations. You cannot set up a dependency relation on different nodes.<br>Source server: Web server or Java EE server<br>Destination server: Performance tracer |

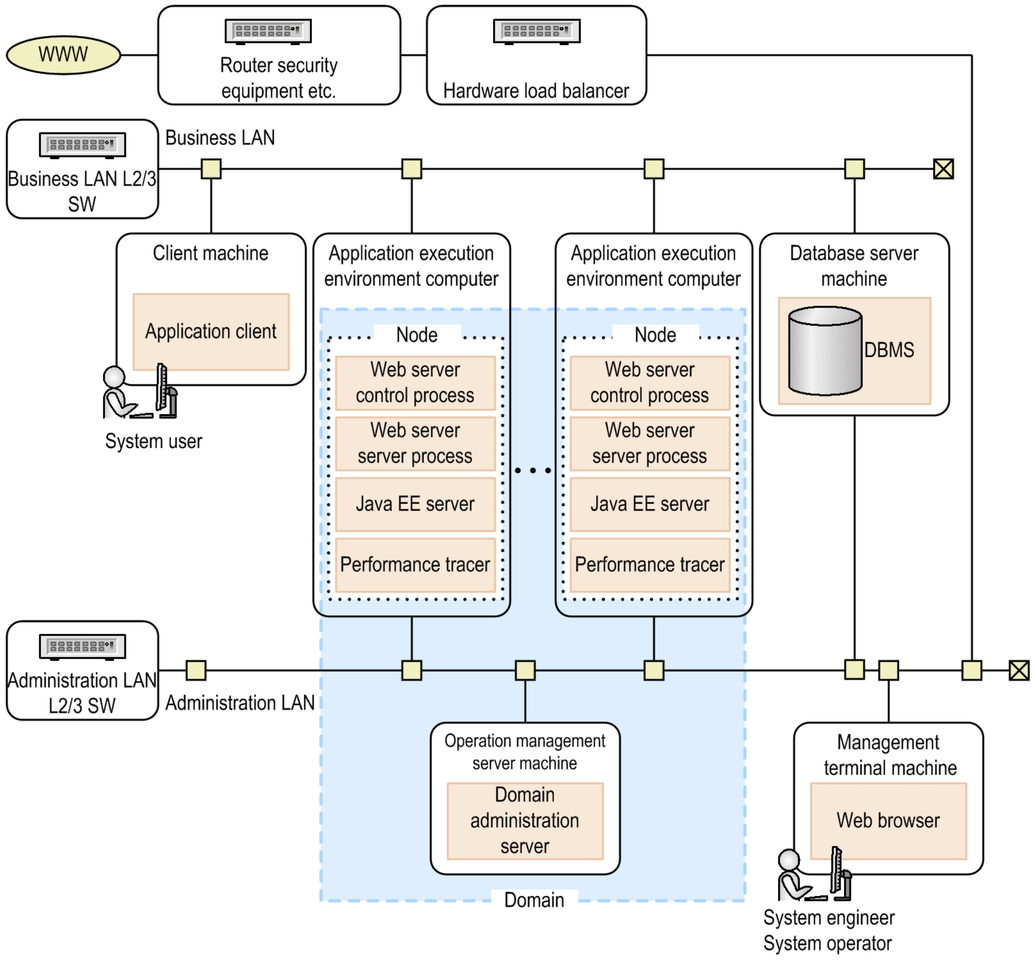
Configuration

A configuration is an instantiated element that holds configuration information for various servers. Servers and clusters obtain configuration information from this element. However, if a server is associated with a certain configuration and the value of a parameter differs between the server and the configuration, the parameter set on the server takes precedence. There are three types of configurations: Java EE server, web server, and performance tracer configurations.

| Configuration type | Overview |
|---|---|
| Java EE server configuration | This configuration is associated with Java EE servers and clusters, and contains configuration information. Creating a Java EE server or cluster automatically creates a configuration for the server or cluster.<br>If a Java EE server is registered for a cluster, the Java EE server inherits the configuration associated with the cluster. |
| Web server configuration | This configuration is associated with web servers and contains configuration information. Creating a web server automatically creates a configuration for the server. |
| Performance tracer configuration | This configuration is associated with performance tracers and contains configuration information. Creating a performance tracer automatically creates a configuration for the performance tracer. |

## Process configurations for Application Server

The following figure illustrates an example of process configurations for Application Server.


WWW
Router security equipment etc.
Hardware load balancer
Business LAN L2/3 SW
Business LAN
Client machine
Application client
System user
Application execution environment computer
Node
Web server control process
Web server server process
Java EE server
Performance tracer
Application execution environment computer
Node
Web server control process
Web server server process
Java EE server
Performance tracer
Database server machine
DBMS
Administration LAN L2/3 SW
Administration LAN
Operation management server machine
Domain administration server
Domain
Management terminal machine
Web browser
System engineer
System operator

Legend:

▭ : Hardware

▨ : Process
The configuration might contain multiple processes that include child processes.

The following table shows a list of processes on Application Server that are operated with root privileges.

| Component | Process name | | | Process privileges | Role |
|---|---|---|---|---|---|
| Application client | `java` | | | General user privileges | Executes an application client. |
| Web server control process | `httpsd` | | | Administrator privileges | Starts a server process that handles requests, and monitors its operations. |
| Web-server server process | --- | `httpsd` | | User privileges specified in the `User` directive | Sends responses to static content requests from web browsers, or transfers data to Java EE servers in response to requests from applications.<br>For `httpsd`, the number of processes varies according to the settings and the number of requests.<br>For `rotatelogs` and `rotatelogs2`, the number of processes varies according to the settings. For |
| | --- | `rotatelogs` | | Administrator privileges | |
| | --- | `rotatelogs2` | | Administrator privileges | |
| | --- | --- | `gcache` | User privileges specified in the `User` directive | |

| Component | Process name | | | Process privileges | Role |
|---|---|---|---|---|---|
| | | | | | rotatelogs, the default number of processes is two. For rotatelogs2, the default number of processes is one. |
| Java EE server | java | --- | --- | Administrator privileges | Executes an application. |
| Performance tracer | cprfd | --- | --- | Administrator privileges | Collects a series of execution logs for web servers and Java EE servers. |
| Domain administration server | java | --- | --- | Administrator privileges | Manages operations for web servers, Java EE servers, and performance tracers. |

The following table shows the current directories for the Java EE server processes.

| Java EE server process | Current directory at startup |
|---|---|
| Server instance# | *installation_directory_for_Java_EE_Server*/glassfish/nodes/*node_name*/*server_instance_name*/config |
| Domain administration server# | *installation_directory_for_Java_EE_Server*/glassfish/domains/*domain_name*/config |
| MQ Broker | *installation_directory_for_Java_EE_Server*/glassfish/nodes/*node_name*/*server_instance_name*/config |

#:

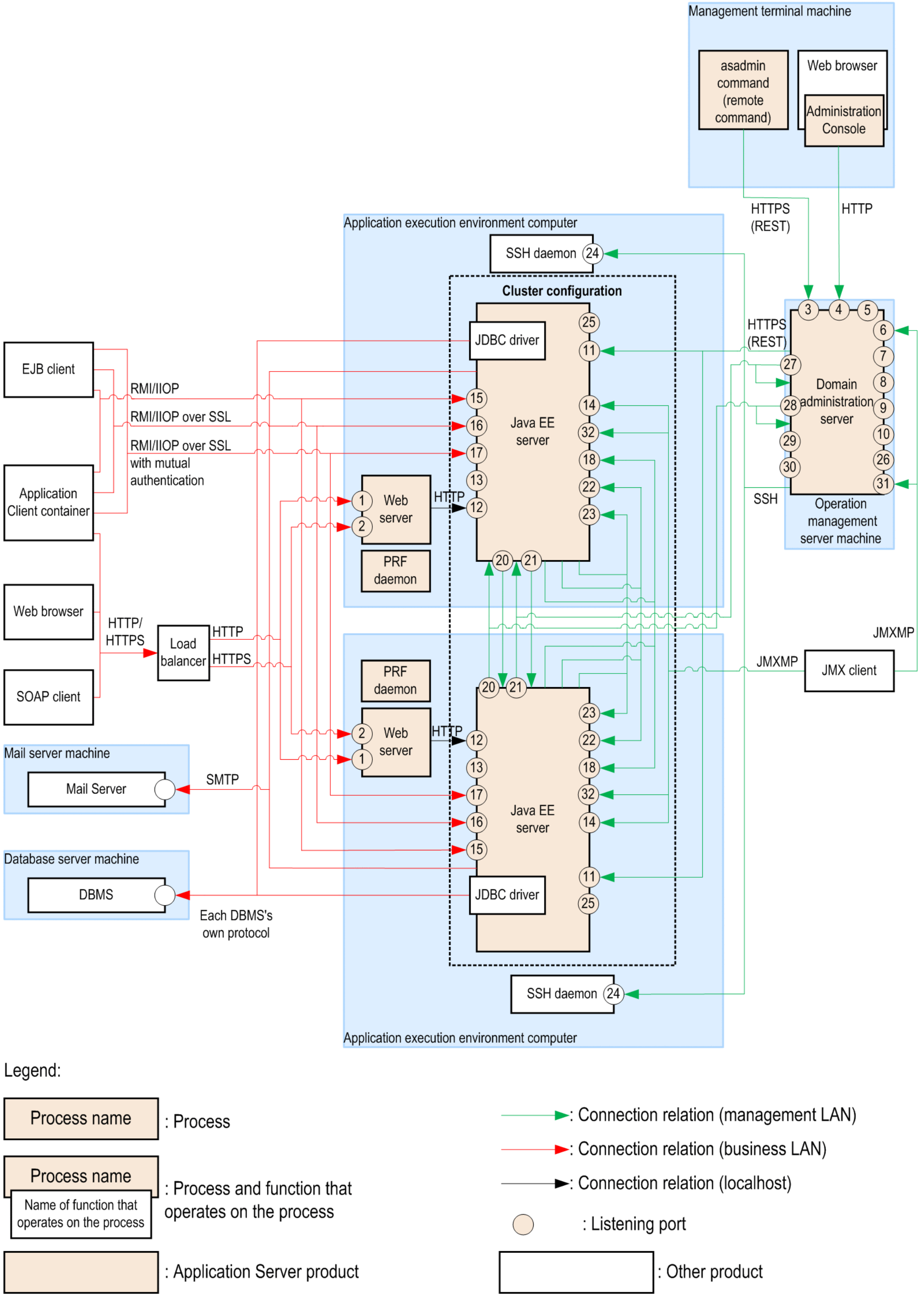
When a process starts or restarts, all files that were output to the current directory (excluding the log files below) are deleted. Note that, if you change the names of the log files below, the log files will also be deleted. In addition, if you execute the `start-instance` subcommand of the `asadmin` utility command with the `--sync=full` option specified, all files in the current directory including these log files will be deleted.

| No. | Log |
|---|---|
| 1 | Thread dump log |
| 2 | Memory dump log |
| 3 | Java VM output message log (error report file) |
| 4 | Compiler replay file |

## 2.5 Connection configurations for Application Server

Application Server operates by communicating with processes on its components (such as Java EE servers, web servers, and domain administration servers), and peripherals (such as load balancers, DBMSs, and clients). This section explains connection relationships between these processes, and the port numbers used by these processes.

The following figure illustrates connection relationships for Application Server.


Management terminal machine
asadmin command (remote command)
Web browser
Administration Console
HTTPS (REST)
HTTP
Application execution environment computer
SSH daemon
24
Cluster configuration
JDBC driver
25
11
15
16
17
13
12
14
32
18
22
23
20
21
Java EE server
Web server
1
2
HTTP
PRF daemon
EJB client
RMI/IIOP
RMI/IIOP over SSL
RMI/IIOP over SSL with mutual authentication
Application Client container
Web browser
HTTP/ HTTPS
Load balancer
HTTP
HTTPS
SOAP client
Mail server machine
Mail Server
SMTP
Database server machine
DBMS
Each DBMS's own protocol
HTTPS (REST)
3
4
5
6
7
8
9
10
26
31
27
28
29
30
Domain administration server
SSH
Operation management server machine
JMXMP
JMXMP
JMX client
Legend:
Process name
: Process
Process name
Name of function that operates on the process
: Process and function that operates on the process
: Application Server product
: Connection relation (management LAN)
: Connection relation (business LAN)
: Connection relation (localhost)
: Listening port
: Other product

The following table shows the port numbers of the ports used by each process of Application Server, whether port numbers can be changed, whether bound IP addresses can be specified, whether ports can be opened and closed, and communication clients.

| Process | Numbers in the diagram | Description | Default port number/TCP or UDP | Can the port number be changed? | Can the IP address be specified? | Can the port be opened and closed?#1 | Communication client |
|---|---|---|---|---|---|---|---|
| Web server | 1 | HTTP request reception port | 80/TCP | Y | Y | Y | Web client on the local machine or a remote machine |
| | 2 | HTTPS request reception port | 443/TCP | Y | Y | Y | Web client on the local machine or a remote machine |
| Domain administration server | 3 | Port for the domain administration server | 4848/TCP | Y | Y | N#2 | The `asadmin` utility command on a remote machine or the local machine including an application execution environment computer |
| | 4 | HTTP request reception port | 8080/TCP | Y | Y | Y | Administration Console on the local machine or a remote machine |
| | 5 | HTTPS request reception port | 8181/TCP | Y | Y | Y | None |
| | 6 | JMX request reception port | 8686/TCP | Y | Y | Y | JMX client on the local machine or a remote machine |
| | 7 | Communication port for IIOP | 3700/TCP | Y | Y | Y | None |
| | 8 | Communication port for IIOP/SSL | 3820/TCP | Y | Y | Y | None |
| | 9 | Communication port for IIOP/SSL with mutual authentication | 3920/TCP | Y | Y | Y | None |
| | 10 | Communication port for the Java debugger | 9009/TCP | Y | Y | Y | None |
| | 26 | Port for the JMS Provider | 7676/TCP | Y | Y | Y | None |
| | 27 | GMS listener port | 9090-9200/TCP | Y | Y | Y | Java EE server on the local machine or a remote machine |
| | 28 | GMS multicast port | 2048-49151/UDP | Y | Y | Y | Java EE server on the local machine or a remote machine |
| | 29 | TCP port for Message Queuing | Ephemeral port/TCP | Y | Y | Y#3 | None |
| | 30 | Port for Message Queue Administration | Ephemeral port/TCP | Y | Y | Y#3 | None |

| Process | Numbers in the diagram | Description | Default port number/TCP or UDP | Can the port number be changed? | Can the IP address be specified? | Can the port be opened and closed?[#1] | Communication client |
|---|---|---|---|---|---|---|---|
| | 31 | Port for Message Queue RMI Registry | Ephemeral port/TCP | N | N | Y[#3] | JMX client on the local machine or a remote machine |
| Java EE server | 11 | Port for the domain administration server | 24848[#4]/TCP | Y | Y | N[#2] | Domain administration server on the local machine or a remote machine |
| | 12 | HTTP request reception port | 28080[#4]/TCP | Y | Y | Y | Web Server on the local machine or a remote machine |
| | 13 | HTTPS request reception port | 28181[#4]/TCP | Y | Y | Y | None |
| | 14 | Communication port for JMX | 28686[#4]/TCP | Y | Y | Y | JMX client on the local machine or a remote machine |
| | 15 | Communication port for IIOP | 23700[#4]/TCP | Y | Y | Y | • EJB client on the local machine or a remote machine<br>• Application Client Container on the local machine or a remote machine |
| | 16 | Communication port for IIOP/SSL | 23820[#4]/TCP | Y | Y | Y | • EJB client on the local machine or a remote machine<br>• Application Client Container on the local machine or a remote machine |
| | 17 | Communication port for IIOP/SSL with mutual authentication | 23920[#4]/TCP | Y | Y | Y | • EJB client on the local machine or a remote machine<br>• Application Client Container on the local machine or a remote machine |
| | 18 | Port for the JMS Provider | 27676[#4]/TCP | Y | Y | Y | Java EE server on the local machine or a remote machine |
| | 20 | GMS listener port | 9090-9200/TCP | Y | Y | Y | Java EE server on the local machine or a remote machine |
| | 21 | GMS multicast port | 2048-49151/UDP | Y | Y | Y | Java EE server on the local machine or a remote machine |
| | 22 | TCP port for Message Queuing | Ephemeral port/TCP | Y | Y | Y[#3] | Java EE server on the local machine or a remote machine |

| Process | Numbers in the diagram | Description | Default port number/TCP or UDP | Can the port number be changed? | Can the IP address be specified? | Can the port be opened and closed? #1 | Communication client |
|---|---|---|---|---|---|---|---|
| | 23 | Port for Message Queue Administration | Ephemeral port/TCP | Y | Y | Y#3 | Java EE server on the local machine or a remote machine |
| | 32 | Port for Message Queue RMI Registry | Ephemeral port/TCP | N | N | Y#3 | JMX client on the local machine or a remote machine |
| SSH daemon | 24 | Listening port on an SSH server | User-specified (22)/TCP | #5 | #5 | #5 | Domain administration server on the local machine or a remote machine |
| Java EE server | 25 | Port for connecting to the Felix Shell | 26666#4 /TCP | Y | N#6 | N | Telnet client on the local machine |

Legend:

Y: It is possible to change port numbers, specify IP addresses, and open and close ports.

N: It is not possible to change port numbers, specify IP addresses, or open or close ports.

#1

Indicates whether users can open and close the port.

#2

This port is required for the corresponding process and always opens when that process starts.

#3

These ports open and close in sync with the JMS Provider port.

#4

You can specify the IP address with a command option. If you omit the option, the first Java EE server is created by using the default IP address. For the second and subsequent Java EE servers, IP addresses are automatically set by the domain administration server during server creation.

#5

This possibility differs depending on whether an SSH daemon is installed.

#6

This port is closed by default.

# 2.6 Directory structure after installation

The following tables show the directory structure immediately after Application Server is installed.

## Directory structure for Application Server

| Directory name | | Description |
|---|---|---|
| *installation_directory_for_Application_Server* | -- | Application Server installation directory |
| | common | Application Server - Base installation directory |
| | httpsd | Web Server installation directory |
| | javaee | Java EE Server installation directory |
| | jdk | Developer's Kit for Java installation directory |

## 2.7 Application Server administration

Application Server administration is to construct and operate Application Server systems by managing Application Server components based on a domain administration server. You can perform Application Server administration from the command line interface (CLI) or the Administration Console.

Application Server administration manages each Application Server component (Java EE servers, web servers, and performance tracers) by grouping them into units called *domains*. Application Server systems with different configurations can be managed by domain according to the business requirements.

The command line interface (CLI) and Administration Console are provided as interfaces with the domain administration server on which Application Server administration is based. You can use either the CLI or the Administration Console to perform various operations for Application Server administration.

Command line interface

The interface for users to execute the various functions for creating and operating an Application Server system by using commands. A user with execution privileges can execute these commands. These commands can be executed when the directory that contains them is the current directory.

Administration Console

An interface for users to execute the various functions for creating and operating an Application Server system from a web browser. Users can log in to the Administration Console from a web browser.

## 2.8 Compatibility and upgradability with Application Server V9

Application Server V10 enables you to use the functions and interfaces provided by Application Server V9 within the range of the Java EE standard specifications. Note that only some product-specific functions of Application Server V9 can be used with Application Server V10.

### Availability of the functions provided by Application Server V9

Application Server V10 adopts Java EE RI (GlassFish 4.0) to support Java EE 7. The functions of Application Server V9 can be used with Application Server V10 within the range of the Java EE standard specifications, but only some product-specific functions of Application Server V9 can be used with Application Server V10.

### Compatibility and upgradability of user interfaces

The following table shows the compatibility of user interfaces between Application Server V9 and V10. We do not guarantee the upgradability of product-specific items from Application Server V9 to V10.

| Interface type | Compatibility |
|---|---|
| Standard APIs and DDs | Compatible |
| Product-specific APIs, GUIs, commands, log formats, and definitions | Incompatible |

# 3 Application Server design items

This chapter describes the information necessary for designing a system that uses Application Server. To ensure the reliability and performance of the system, select the system design items to fit your needs.

# 3.1 Memory management in Java

This section describes the memory management mechanisms that need to be considered and the parameters to be set during the design of memory management in Java.

## 3.1.1 Memory management methods for Java

The memory management for Java uses GC and Application Server functions to control the memory area used by programs. To properly manage the Java memory, you need to understand the structure of the memory area to be used and the flow of GC processing. You can select from three Java memory management methods: SerialGC, a combination of SerialGC and the explicit memory management functionality (the functionality that suppresses GC), or G1GC. By selecting the appropriate memory management method based on the system requirements, you can improve the system processing performance.

Use GC to manage memory for Application Server and for applications running on Application Server. If a request is sent while GC is being performed, processing of the request stops until GC ends. For this reason, whether GC can be properly executed greatly affects the processing performance of the system.

The following table shows the memory management method for Java for Application Server. Depending on the memory management method, the operation of GC varies. Therefore, you must select the memory management method for Java based on the system requirements.

| No. | Memory management method of Java | Appropriate system and characteristics |
|---|---|---|
| 1 | SerialGC | This method is suitable for systems that put importance on throughput.<br>• This method provides high throughput.<br>• The time when GC is performed cannot be controlled.<br>• By tuning the memory size, you can suppress the occurrence of FullGC. |
| 2 | Combination of SerialGC and the explicit memory management functionality | This method is suitable for general web front systems that use sessions.<br>• This method provides high throughput.<br>• The time when GC is performed cannot be controlled.<br>• In a system that uses sessions, occurrence of FullGC can be suppressed by tuning the memory size or by managing sessions by using the explicit memory management functionality in the Explicit heap. |
| 3 | G1GC | This method is suitable for systems that use large amount of memory, or systems that put importance on responses.<br>• Throughput is low.<br>• The time when GC is performed can be partially controlled.<br>• By tuning the memory size, you can suppress the occurrence of FullGC.<br>• By increasing the number of threads where GC is performed, you can suppress the occurrence of FullGC. |

### Related topics

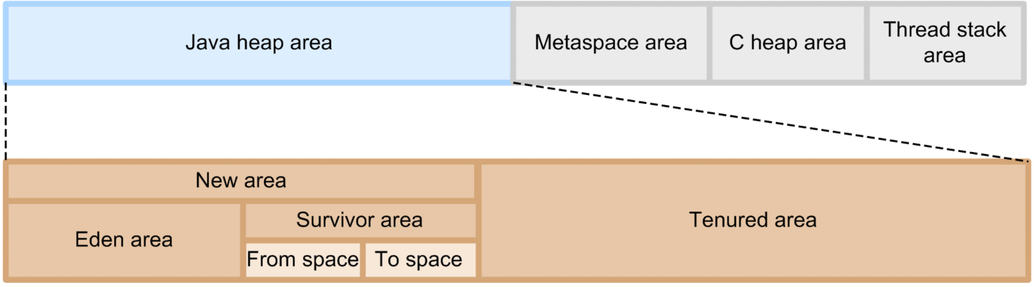
- 3.1.2 Memory structure and GC flow for SerialGC
- 3.1.3 Memory structure and GC flow when SerialGC and the explicit memory management functionality are combined
- 3.1.4 Memory structure and GC flow for G1GC

## 3.1.2 Memory structure and GC flow for SerialGC

If you select SerialGC as the Java memory management method, CopyGC or FullGC occurs according to the usage of the memory area. When you select SerialGC, specify the `-XX:+UseSerialGC` option, and the options related to the memory area.

### Memory structure of SerialGC

The following figure illustrates the memory structure when SerialGC is selected as the Java memory management method.

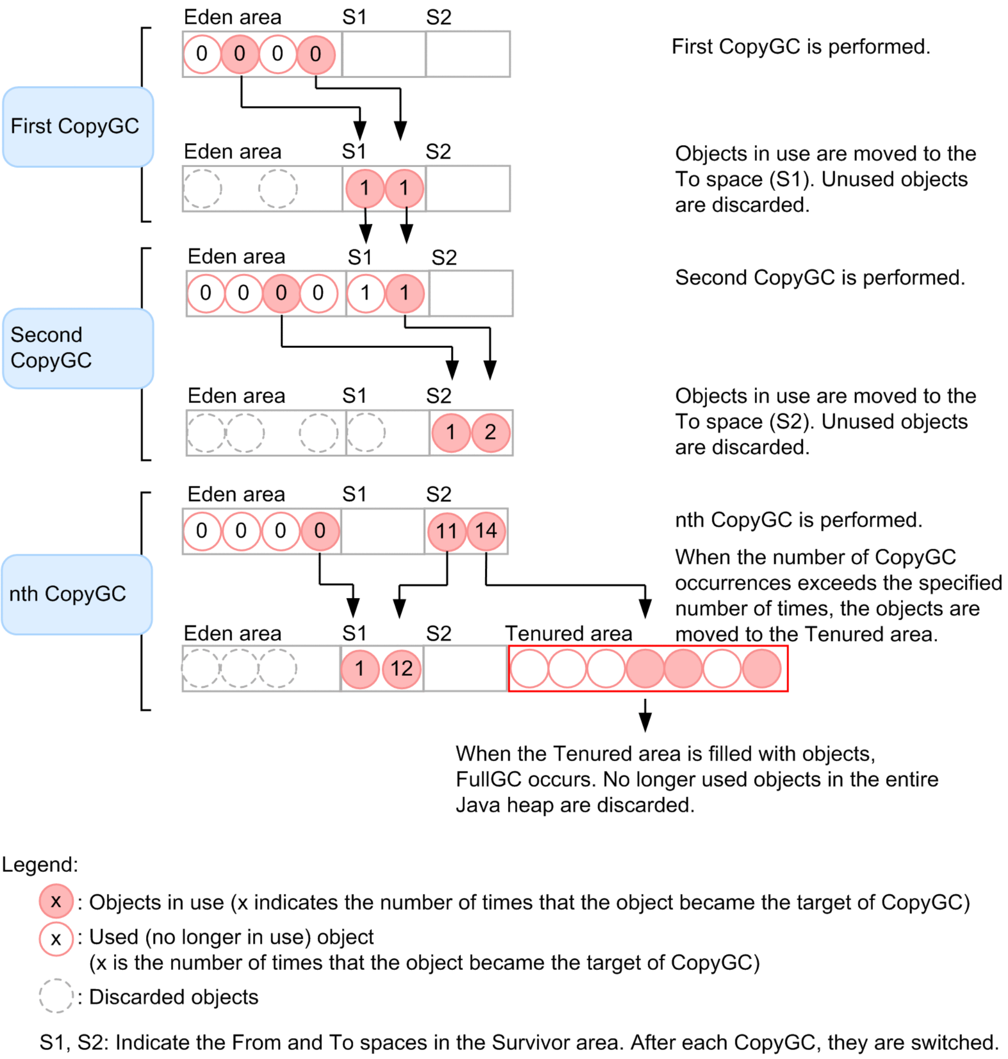


The following table shows the roles of each area.

| No. | Area name | Role |
|---|---|---|
| 1 | Java heap area | The memory area used by Java programs. This area is divided into the New area and the Tenured area. |
| 2 | New area | The area that stores new objects. This area is divided into the Eden area and the Survivor area. |
| 3 | Eden area | The area that stores objects immediately after they are created |
| 4 | Survivor area | The area that stores Java objects that are in use in the New area, and which were not discarded during execution of CopyGC. This area is divided into the From space and the To space. |
| 5 | Tenured area | The area that stores objects to be used for a long time |
| 6 | Metaspace area | The area that stores loaded class information and method information |
| 7 | C heap area | The area used when Developer's Kit for Java executes the native library. This area is specific to the OS. |
| 8 | Thread stack area | The stack area retained for each thread. This area is specific to the OS. |

### GC flow for SerialGC

The following figure illustrates the GC flow for SerialGC.



First CopyGC is performed.

Objects in use are moved to the To space (S1). Unused objects are discarded.

Second CopyGC is performed.

Objects in use are moved to the To space (S2). Unused objects are discarded.

nth CopyGC is performed.

When the number of CopyGC occurrences exceeds the specified number of times, the objects are moved to the Tenured area.

When the Tenured area is filled with objects, FullGC occurs. No longer used objects in the entire Java heap are discarded.

Legend:

x : Objects in use (x indicates the number of times that the object became the target of CopyGC)

x : Used (no longer in use) object
(x is the number of times that the object became the target of CopyGC)

: Discarded objects

S1, S2: Indicate the From and To spaces in the Survivor area. After each CopyGC, they are switched.

1. First CopyGC

   Generated objects are stored in the Eden area in the New area. When the Eden area is filled with objects, the first CopyGC occurs. The entire New area is the target of the CopyGC. When CopyGC occurs, no longer used objects are deleted, and objects in use are moved to the To space (S1) in the Survivor area.

2. Second CopyGC

   After the first CopyGC, when the Eden area is filled with objects again, the second CopyGC occurs. Because CopyGC targets the entire New area, objects moved to the To space (S1) in the Survivor area from the Eden area for the first CopyGC are subject to CopyGC. Objects in use stored in the Eden area when the second CopyGC occurs are moved to the To space (S2), and the objects in use in the From space (S1) are moved to the To space (S2).

3. nth CopyGC

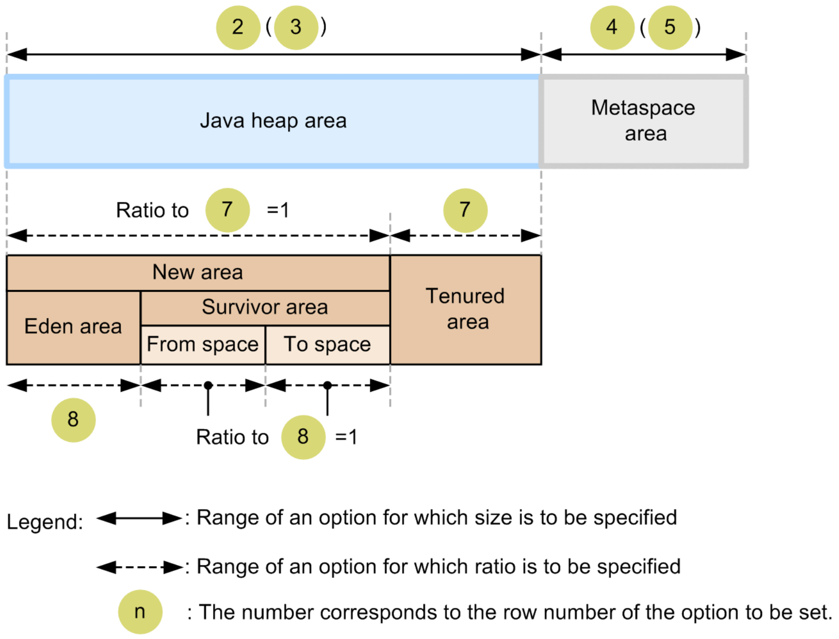
   If objects moved to the Survivor area are being used when CopyGC occurs, they keep moving between the From space and the To space in the Survivor area alternately. If the number of times specified for CopyGC occurrences of an object exceeds the specified number (15 times in the figure), the object is moved from the Survivor area to the Tenured area.

4. FullGC

   When the Tenured area is filled with objects, FullGC occurs, and no longer used objects in the entire Java heap area are deleted.

## Parameters to set for SerialGC

For SerialGC, FullGC, which will stop applications for a long time, will always occur. Therefore, you must design and tune memory so that FullGC will not occur until a time in which it can occur without issues. The following table lists the parameters to set for SerialGC.


2 ( 3 )
4 ( 5 )
Java heap area
Metaspace area
Ratio to 7 =1
7
New area
Eden area
Survivor area
From space
To space
Tenured area
8
Ratio to 8 =1
Legend:
: Range of an option for which size is to be specified
: Range of an option for which ratio is to be specified
n
: The number corresponds to the row number of the option to be set.

| No. | Item | Option name | Description |
|---|---|---|---|
| 1 | SerialGC settings | `-XX:+UseSerialGC` | Select SerialGC as the memory management method. This parameter is enabled by default. |
| 2 | Designing memory | `-Xmx`*`maximum_Java_heap_area_size`* | Set the maximum size for the Java heap area. |
| 3 | | `-Xms`*`initial_Java_heap_area_size`* | Set the initial size for the Java heap area. For this parameter, we recommend setting the same value as the `-Xmx` option. |
| 4 | | `-XX:MaxMetaspaceSize =` *`maximum_Metaspace_area_size`* | Set the maximum size for the Metaspace area. |
| 5 | | `-XX:MetaspaceSize=`*`standard_value_for_FullGC_that_originates_from_the_Metaspace_area`* | Set the standard value for FullGC that originates from the Metaspace area. FullGC occurs when the Metaspace area size exceeds the standard value, so specify a value estimated from the size of class information required for applications. For this parameter, we recommend setting the same value as the `-XX:MaxMetaspaceSize` option. |
| 6 | | `-XX:CompressedClassSpaceSize =` *`maximum_Compressed_Class_Space_size`* | Set the maximum size of the Compressed Class Space area to be created in the Metaspace area when the compressed object pointer functionality is available. For this parameter, we recommend setting the same value as the `-XX:MaxMetaspaceSize` option. |
| 7 | | `-XX:NewRatio =` *`ratio_of_Tenured_area_to_New_area`* | Set the ratio of the Tenured area when the New area is assumed to be 1. |

| No. | Item | Option name | Description |
|---|---|---|---|
| 8 | | -XX:SurvivorRatio=*ratio_of_Eden_area_to_From_and_To_areas_of_Survivor_area* | Set the ratio of the Eden area when the ratio of the From and To spaces of the Survivor area is assumed to be 1. |

**Related topics**

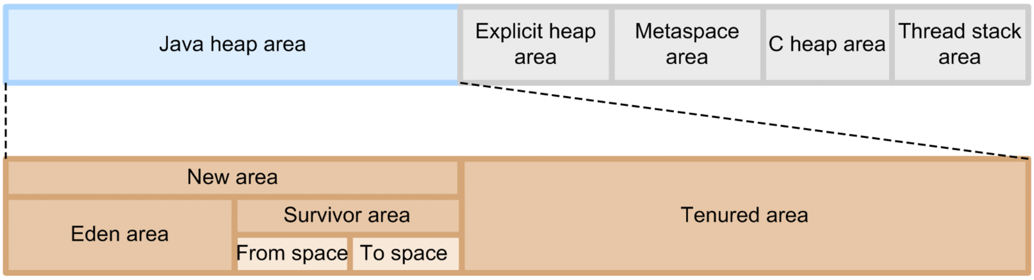
- 4.6.5 Changing Java VM options by using the create-jvm-options subcommand
- 8.2.3 Changing Java VM options by using the create-jvm-options subcommand

## 3.1.3 Memory structure and GC flow when SerialGC and the explicit memory management functionality are combined

If you select the SerialGC and explicit memory management functionality combined method as the Java memory management method, you can suppress FullGC that occurs in SerialGC by using the explicit memory management functionality. If you select the SerialGC and explicit memory management functionality combined method, specify `-XX:+UseSerialGC` options, memory options, and other options related to the settings on the explicit memory management functionality and the memory design for the Explicit heap area.

### Memory structure when SerialGC and the explicit memory management functionality are combined

The following figure illustrates the memory structure when SerialGC and the explicit memory management functionality are combined as the Java memory management method.



The following table lists the roles of the areas.

| No. | Area name | Role |
|---|---|---|
| 1 | Java heap area | The memory area used by Java programs. The area is divided into the New area and the Tenured area. |
| 2 | New area | The area that stores new objects. This area is divided into the Eden area and the Survivor area. |
| 3 | Eden area | The area that stores objects immediately after they are created |
| 4 | Survivor area | The area that stores objects for which GC has been performed at least once and which are in use |
| 5 | Tenured area | The area that stores objects to be used for a long time |
| 7 | Explicit heap area | The area that stores session objects |

| No. | Area name | Role |
|---|---|---|
| 6 | Metaspace area | The area that stores loaded class information or method information |
| 8 | C heap area | The area used when Developer's Kit for Java executes the native library |
| 9 | Thread stack area | A stack area retained for each thread. |

## Memory management mechanism when SerialGC and the explicit memory management functionality are combined

If you select SerialGC as the memory management method, and use the Java heap area, FullGC might occur. To prevent FullGC from occurring, use the explicit memory management functionality to store objects that cause FullGC in the Explicit heap area instead of the Java heap area. Because the objects that cause FullGC are not stored in the Tenured area, which is a FullGC target, you can prevent FullGC from occurring. Note that the objects stored in the Explicit heap area are released explicitly when the object's lifetime ends.

Objects causing FullGC

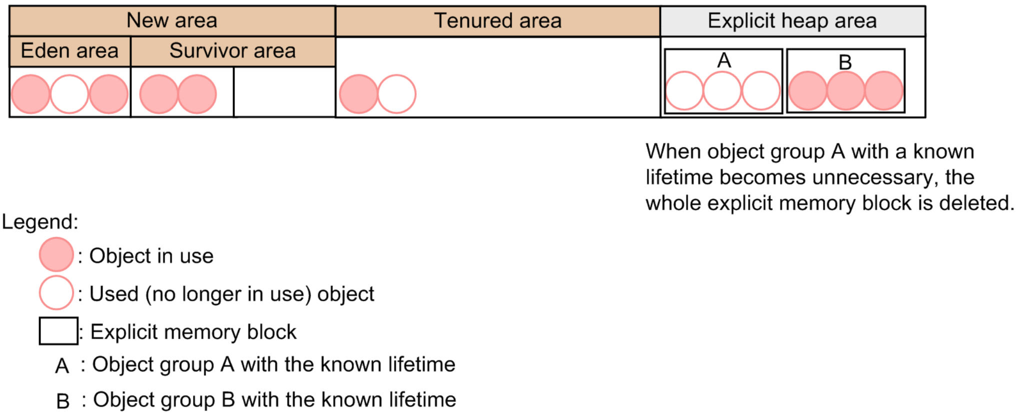
Objects that have a long lifetime and which become unnecessary after a certain period of time cause FullGC. This applies, for example, to objects relating to session information which is used for a series of processes from login to logout. Because session information is used across multiple requests, it is used for a long time, and will become unnecessary after a certain period of time and when the user logs out. For Application Servers, objects relating to session information are set as objects to be stored in the Explicit heap area by default.

Memory status when objects become unnecessary

The figure below illustrates the memory status when objects with a known lifetime become unnecessary after a specified period of time. Note that the Explicit heap area when the explicit memory management functionality is used consists of memory blocks named explicit memory blocks. Memory in the Explicit area is initialized or released for each explicit memory block.

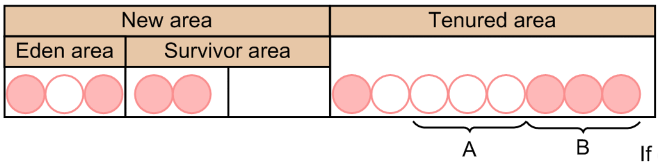
When the explicit memory management functionality is used

When object group A with a known lifetime in the Explicit heap area becomes unnecessary after a specified period, the explicit memory block where object group A is stored will be deleted.


New area
Eden area
Survivor area
Tenured area
Explicit heap area
A
B
When object group A with a known lifetime becomes unnecessary, the whole explicit memory block is deleted.
Legend:
: Object in use
: Used (no longer in use) object
: Explicit memory block
A : Object group A with the known lifetime
B : Object group B with the known lifetime

When the explicit memory management functionality is not used

When object group A with a known lifetime in the Tenured area becomes unnecessary after a specified period, the objects in the group remain in the Tenured area until FullGC occurs.



If object group A with a known lifetime becomes unnecessary, it remains in the Tenured area until FullGC occurs.

Legend:

● : Object in use

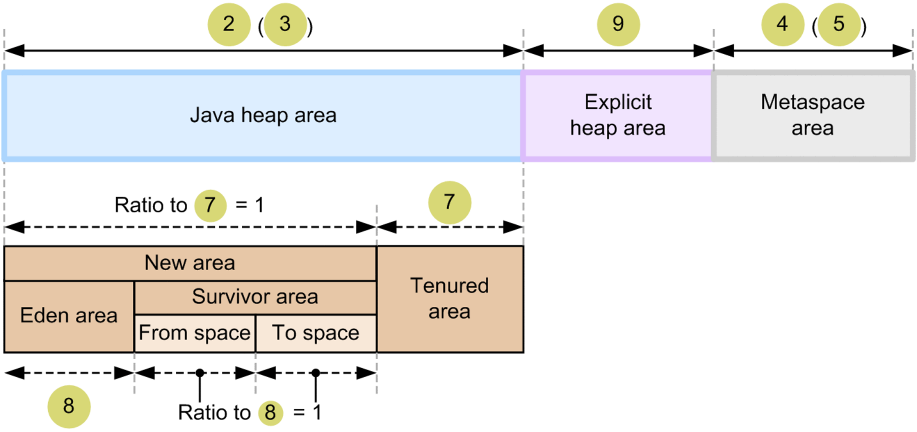
○ : Used (no longer in use) object

A : Object group A with the known lifetime

B : Object group B with the known lifetime

## Parameters to set when SerialGC and the explicit memory management functionality are combined

If you select the SerialGC and explicit memory management functionality combined method as the memory management method, you need to design and tune the memory in the Java heap area and the Explicit heap area. The following figure illustrates the parameters to set when SerialGC and the explicit memory management functionality are combined.



Legend: ◄──► : Range of an option for which size is to be specified

◄---► : Range of an option for which ratio is to be specified

*n* : The number corresponds to the row number of the option to be set.

| No. | Item | Option name | Description |
|---|---|---|---|
| 1 | SerialGC settings | `-XX:+UseSerialGC` | Select SerialGC as the memory management method. This parameter is enabled by default. |
| 2 | Designing memory | `-Xmx`*`maximum_Java_heap_area_size`* | Set the maximum size of the Java heap area. |
| 3 | | `-Xms`*`initial_Java_heap_area_size`* | Set the initial size of the Java heap area. For this parameter, we recommend setting the same value as the `-Xmx` option. |
| 4 | | `-XX:MaxMetaspaceSize =` *`maximum_Metaspace_area_size`* | Set the maximum size of the Metaspace area. |

| No. | Item | Option name | Description |
|---|---|---|---|
| 5 | | -XX:MetaspaceSize=*standard_value_for_FullGC_that_originates_from_the_Metaspace_area* | Set the standard value for FullGC that originates from the Metaspace area. FullGC occurs when the Metaspace area size exceeds the standard value, so specify a value estimated from the size of class information required for applications. For this parameter, we recommend setting the same value as the -XX:MaxMetaspaceSize option. |
| 6 | | -XX:CompressedClassSpaceSize = *maximum_Compressed_Class_Space_size* | Set the maximum size of the Compressed Class Space area to be created in the Metaspace area when the compressed object pointer functionality is available. For this parameter, we recommend setting the same value as the -XX:MaxMetaspaceSize option. |
| 7 | | -XX:NewRatio = *ratio_of_Tenured_area_relative_to_New_area* | Set the ratio of the Tenured area when the New area is assumed to be 1. |
| 8 | | -XX:SurvivorRatio = *ratio_of_Eden_area_relative_to_From_space_and_To_space_in_Survivor_area* | Set the ratio of the Eden area when the From and To spaces of the Survivor area are assumed to be 1. |
| 9 | Explicit memory management functionality settings | -XX:+HitachiUseExplicitMemory | Enable the explicit memory management functionality. This parameter is enabled by default. |
| 10 | Memory design for the Explicit heap area | -XX:HitachiExplicitHeapMaxSize = *maximum_Explicit_heap_area_size* | Set the maximum size of the Explicit heap area size. |

### Related topics

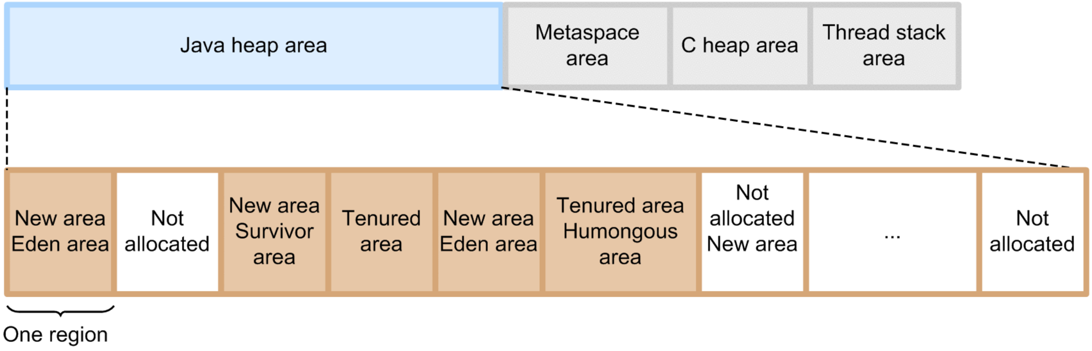
- 4.6.5 Changing Java VM options by using the create-jvm-options subcommand
- 8.2.3 Changing Java VM options by using the create-jvm-options subcommand

## 3.1.4 Memory structure and GC flow for G1GC

When G1GC is selected as the Java memory management method, three types of GCs including YoungGC, MixedGC, and FullGC occur for the memory areas. For YoungGC and MixedGC, you can control when to stop applications when GC occurs. To select G1GC, specify the -XX:+UseG1GC option, the target time for stopping applications when GC occurs, the options related to Java heap, and the options for tuning processing performance.

### Memory structure for G1GC

The following figure illustrates the memory structure when G1GC is selected as the memory management method of Java.



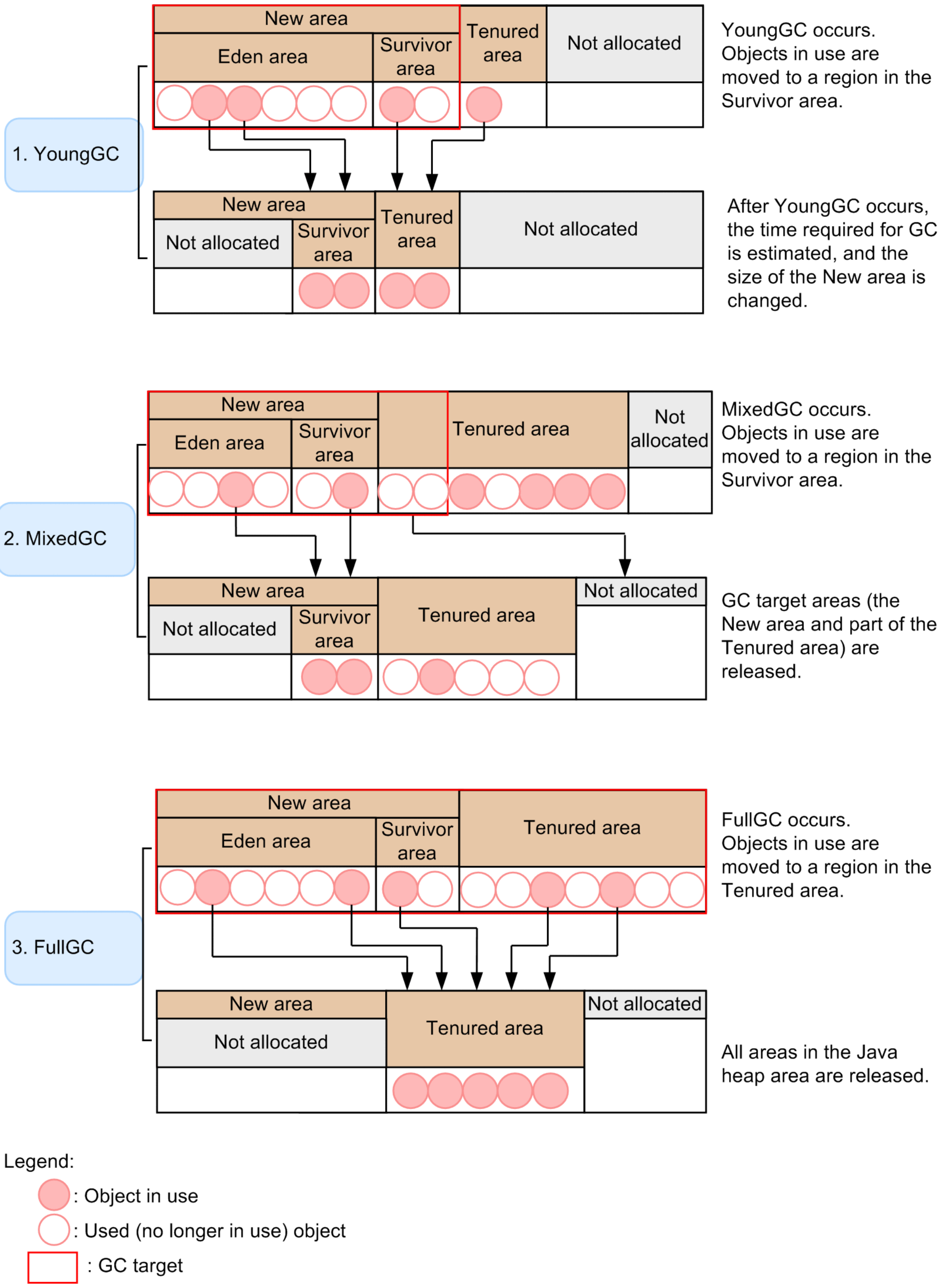
The following table shows the roles of each area.

| No. | Area name | Role |
|---|---|---|
| 1 | Java heap area | The memory area used by Java programs. This area is divided into the New area and the Tenured area. |
| 2 | New area | The area that stores new objects. This area is divided into the Eden area and the Survivor area. |
| 3 | Eden area | The area that stores objects immediately after they are created |
| 4 | Survivor area | The area that stores objects that are in use and for which GC has been performed at least once |
| 5 | Tenured area | The area that stores objects to be used for a long time |
| 6 | Metaspace area | The area that stores loaded class information and method information |
| 7 | C heap area | The area used when Developer's Kit for Java executes the native library |
| 8 | Thread stack area | A stack area that is retained for each thread |
| 9 | Not allocated area in the New area | An area in the New area. This area is not allocated to the Eden or Survivor area. |
| 10 | Humongous area | The area that stores large objects. Part of the Survivor area and continuous regions are allocated. |
| 11 | Not allocated area | An area in the Java heap area. This area is not allocated to any other area. |

## GC flow for G1GC

G1GC manages the Java heap area for each memory block (called a region). For this reason, it does not reserve the New area or the Tenured area as a continuous area. Objects are stored in regions. If there is no available space in a region, G1GC allocates a region which is not currently allocated to the area, and then stores objects there.

The following figure illustrates the GC flow when G1GC is selected as the memory management method.


1. YoungGC
New area
Eden area
Survivor area
Tenured area
Not allocated
YoungGC occurs. Objects in use are moved to a region in the Survivor area.
New area
Not allocated
Survivor area
Tenured area
Not allocated
After YoungGC occurs, the time required for GC is estimated, and the size of the New area is changed.
2. MixedGC
New area
Eden area
Survivor area
Tenured area
Not allocated
MixedGC occurs. Objects in use are moved to a region in the Survivor area.
New area
Not allocated
Survivor area
Tenured area
Not allocated
GC target areas (the New area and part of the Tenured area) are released.
3. FullGC
New area
Eden area
Survivor area
Tenured area
FullGC occurs. Objects in use are moved to a region in the Tenured area.
New area
Not allocated
Tenured area
Not allocated
All areas in the Java heap area are released.
Legend:
● : Object in use
○ : Used (no longer in use) object
□ : GC target

1. YoungGC

   YoungGC occurs when there is no more space in a region allocated to the New area. When this occurs, objects in use are moved to the region allocated to the Survivor area, and objects which are no longer used are released for each region. In addition, each time YoungGC occurs, the objects in use are moved across the regions allocated to the Survivor area. When the number of moves reaches a specified number, the objects move to a region allocated to the Tenured area. After YoungGC occurs, G1GC estimates the time required for the next GC based on the time spent for the current GC, and then changes the size of the New area. The figure after YoungGC is an example of the New area being decreased due to the GC taking more time than expected.

2. MixedGC

   MixedGC occurs when the usage rate of the Tenured area increases. When this occurs, the GC target is the regions allocated to the New area and the regions allocated to part of the Tenured area within the target down time. Based on analysis information about whether the objects that are executed in parallel to applications are in use, these regions allocated to the Tenured region become the GC target in order of the region for which the size of the region to be released is the largest. For this reason, if the analysis of object information is insufficient or the analysis results indicate that MixedGC has little effect, MixedGC does not occur.

3. FullGC

   FullGC occurs for the entire Java heap area, when there is no more space in a region in the heap area and MixedGC does not occur.

## Parameters to set for G1GC

To select G1GC as the memory management method, you need to design and tune the memory so that the application down time and the throughput by GC meet the system requirements. Note that G1GC cannot control application down time triggered by FullGC. For this reason, when FullGC occurs, you need to tune the memory to prevent FullGC from occurring. The following figure shows the parameters to be set for G1GC.

| No. | Item | Option name | Description |
|---|---|---|---|
| 1 | G1GC settings | `-XX:+UseG1GC` | Selects G1GC as the memory management method. |
| 2 | | `-XX:MaxGCPauseMillis =` *`target_down_time`* | Sets the target down time in milliseconds. |
| 3 | Designing the memory | `-Xmx`*`maximum_Java_heap_area_size`* | Sets the maximum size for the Java heap area. |
| 4 | | `-Xms`*`initial_Java_heap_area_size`* | Sets the initial size for the Java area. We recommend that you set the same value for this parameter as the `-Xmx` option. |
| 5 | | `-XX:MaxMetaspaceSize =` *`maximum_Metaspace_area_size`* | Sets the maximum size for the Metaspace area. |
| 6 | | `-XX:MetaspaceSize =` *`initial_Metaspace_area_size`* | Set the standard value for FullGC that results from the Metaspace area. This value must be designed in the same manner as the initial size because the Metaspace area will be expanded if the Metaspace area size exceeds the standard value. We recommend that you set the same value for this parameter as the `-XX:MaxMetaspaceSize` option |
| 7 | | `-XX:CompressedClassSpaceSize =` *`initial_Compressed_Class_Space_area_size`* | Sets the size of the Compressed Class Space area to be created in the Metaspace area when the compressed object pointer functionality is enabled. Specify the same value as the `-XX:MaxMetaspaceSize` option. |
| 8 | | `-XX:SurvivorRatio =` *`ratio_for_maximum_Survivor_area_size_for_New_area`* | Sets the ratio when the size of the Survivor area is at its maximum for the New area. |
| 9 | Tuning processing performance | `-XX:ParallelGCThreads =` *`number_of_threads_for_YoungGC_and_MixedGC`* | Sets the number of threads where YoungGC and MixedGC are performed. Set this value according to the number of threads for `ConcGCThreads`. |
| 10 | | `-XX:ConcGCThreads =`*`number_of_threads_where_processing_is_performed_in_parallel_to_applications`* | Sets the number of threads to be processed in parallel with application. |

## Related topics

- 4.6.5 Changing Java VM options by using the create-jvm-options subcommand
- 8.2.3 Changing Java VM options by using the create-jvm-options subcommand

# 3.2 Load balancing

Load balancing is used to distribute requests to be processed to multiple Application Servers to balance the load of processing requests. To use this functionality, use the hardware load balancer or software load balancer.

## Selecting the load balancing method

For load balancing, use either the hardware load balancer or the software load balancer. Application Server provides a load balancing functionality that uses Web Server as the software load balancer. This functionality distributes processing across servers according to the number of requests.

The following table compares the hardware load balancer and the software load balancer with regard to installation cost and functionality.

| Load balancing method | Cost | Functionality |
|---|---|---|
| Hardware load balancer | High | High |
| Software load balancer | Low | Low |

### Related topics

- 5.1 System environment settings
- 5.2 Setting the software load balancer

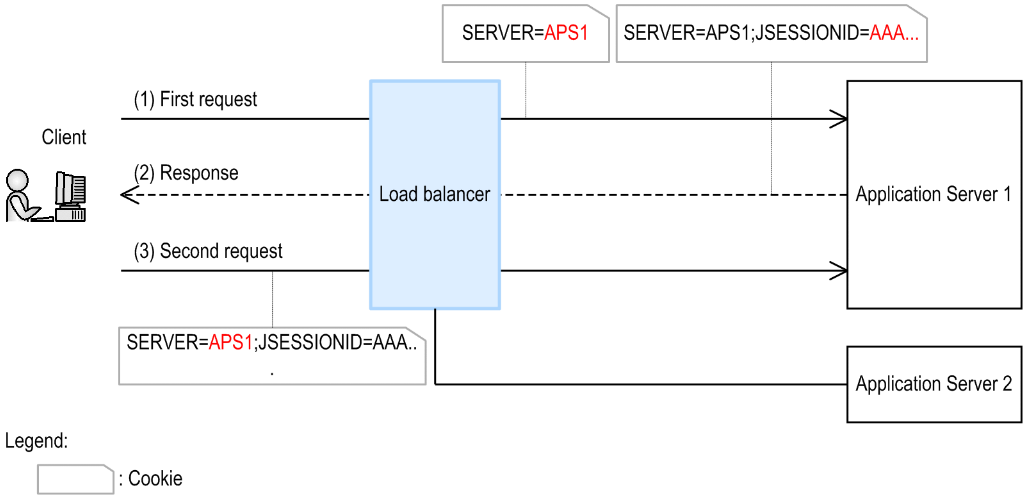
# 3.3 Overview of session management

Session management means managing the association between requests and web clients by using server IDs and session IDs. A "session" is a mechanism that associates business information with requests from a business system user's login to logout. Session management allows a system that uses a load balancer to continue sessions.

## Session management mechanism

Application Server adds the same session ID to cookies and query strings in a series of requests during a session to continue the session.

The following figure shows how the session is maintained in the system configuration including a load balancer.



1. First request from the client

   When receiving the first request from the client, the load balancer transfers the request to Application Server 1. At this time, the load balancer includes a server identification cookie in the request to Application Server 1.

2. Response to the first request

   Application Server 1 sends a response containing the server identification cookie and the session ID to the client.

3. Second request from the client

   The client sends the second request containing the server identification cookie to the load balancer. When receiving the second request, the load balancer determines the request transfer destination from the server identification cookie in the request, and then sends the request to the server (Application Server 1) that has established the session.

## Session ID structure

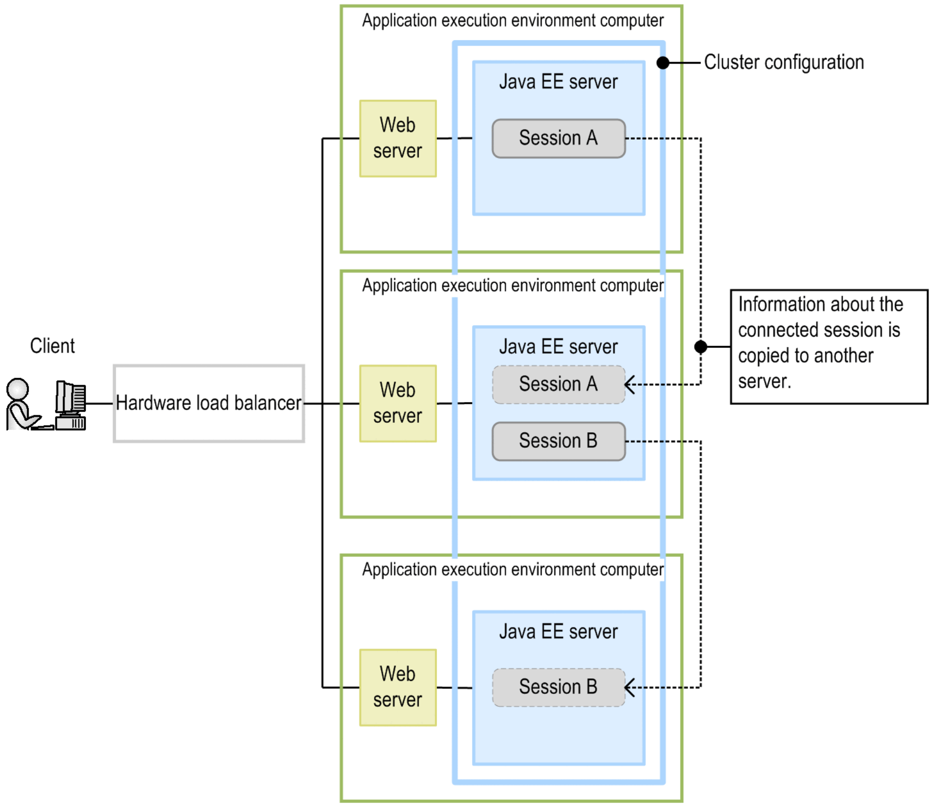
Application Server adds a server ID to a session ID in order to ensure that unique session IDs are used for session management. The following shows the session ID structure.

| | | |
|---|---|---|
| Unique character in Web applications<br>(31 characters) | Period<br>(1 character) | Server ID<br>(A maximum of 63<br>characters) |

The 33rd and subsequent characters in the session ID contain a server ID, which is different for each Java EE server. This ensures that the session IDs are unique among multiple Java EE servers in the system.

## Combining a cluster configuration and session replication to continue a session

If a server processing a request cannot continue processing due to an error, Application Server uses session replication to copy information about the ongoing session to another server that is free of error. Then, Application Server continues the session based on the replicated session information. Session replication can be used in a cluster configuration. The following figure shows the in-memory replication configuration in which each server individually operates as a backup of the session.


Application execution environment computer
Java EE server
Cluster configuration
Web server
Session A
Application execution environment computer
Client
Java EE server
Information about the connected session is copied to another server.
Session A
Hardware load balancer
Web server
Session B
Application execution environment computer
Java EE server
Web server
Session B

## Enabling the web server settings to ensure the continuity of sessions

When you execute the `create-relation` subcommand of the `asadmin` utility command for the cluster from the web server to create a dependency relation, the web server settings that ensure the continuity of sessions are enabled.

```
asadmin create-relation --relationtype redirect --from web_server_name --to
cluster_name --properties
property_name_of_dependency_relation=value[:property_name_of_dependency_relati
on=value] dependency_relation_name
```

## Settings to enable session replication

The following describes the settings that enable session replication.

- Settings for cluster configuration
  Execute the following command to enable session replication:
  ```
  asadmin set cluster_name -config.availability-service.availability-
  enabled=true
  ```
- Application settings
  In `web.xml` file of the application to be deployed, specify the `distributable` element as a child element of the `web-app` element.

- Settings for deploying applications

  When executing a deploy command, enable the `--availabilityenabled` option.

  ```
  asadmin deploy --target=cluster_name --availabilityenabled=true
  path_to_the_archive_file_including_applications
  ```

# 3.4 Overview of transaction management

You need transaction management in any enterprise information system (EIS) or a company's back-end system containing databases and transaction servers to ensure the consistency of data during business transactions. We recommend that you use local transactions for transactions on Application Servers.

## Transaction management mechanism

The transaction management methods available for Application Servers include a local transaction and a global transaction. We recommend that you use a local transaction unless a global transaction must be used for system integration or for other reasons.

Local transaction

A local transaction involves a single resource.

Global transaction

A global transaction involves multiple resources. You must synchronize resources to control global transactions. Control of synchronization processing is expensive.

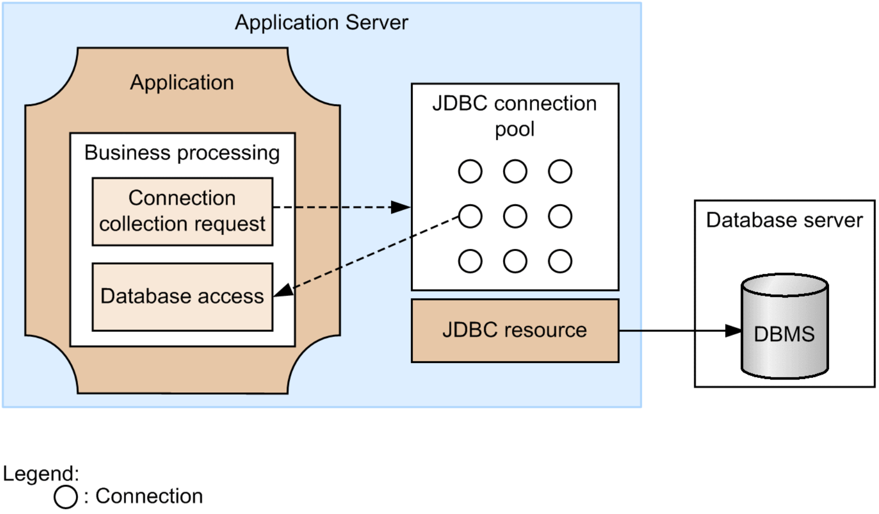
# 3.5 Connection management

Connection management is used to manage connections to a database server by using the JDBC connection pool. By managing the number of connections to be pooled in the JDBC connection pool, the processing time for database access can be reduced. In addition, by using the connection error detection function, processing can continue even if a failure occurs in a connection of the JDBC connection pool.

## Connection management mechanism

Because it takes time to establish a connection to the database server, a certain number of connections are pooled in the JDBC connection pool. When a connection collection request is received, a connection pooled in the JDBC pool is returned. By reusing connections, this mechanism reduces the time required for database access.

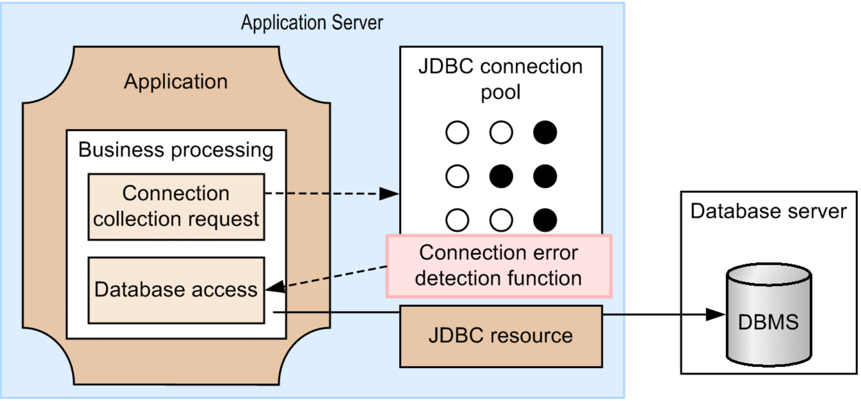
The following figure shows the connection management mechanism.



## Connection error detection function

When the connection error detection function receives a connection collection request, it checks whether there is a failure in the connections of the JDBC connection pool and then returns a connection for which no failure has occurred. Through this operation, even if a failure occurs in a connection due to a resource shutdown or network failure, processing can continue by using a connection for which no failure has occurred.

The following figure shows the connection error detection function.



Legend:

○ : Connection for which no failure has occurred

● : Connection for which a failure has occurred

If a large number is set for the maximum number of connections in a connection pool, a larger amount of memory is required.

If server failures or network failures occur frequently, the resources required for detecting connection failures become insufficient, and connection collection requests fail. When this happens, the KDKD20004-W message is output. Similarly, when you want to discard connections because a timeout error occurred in the connection error detection function, if the resources required for this operation become insufficient due to a large number of failures, a response for discarding connections might not be returned. When this happens, the KDKD20001-W message is output.

Note that, when the connection error detection function is being used, if a new connection is established before a failed connection is discarded, the maximum number of concurrent connections for DBMS might temporarily exceed the connection pool size.

## Tuning the JDBC connection pool

When creating a connection pool, you need to tune the values so that the number of connections set in the connection-destination database for physical connections is greater than or equal to the maximum size of the connection pool.

For details on the number of connections set in the connection-destination database, see the documentation of the applicable database.

## Estimating the maximum number of concurrent connections for DBMS

The formula for estimating the maximum number of concurrent connections for DBMS is shown below. The maximum number of concurrent connections when using multiple JDBC connection pools is the sum of the results of this formula.

```
Maximum number of concurrent connections for DBMS
= JDBC_connection_pool_size + a#
```

#:

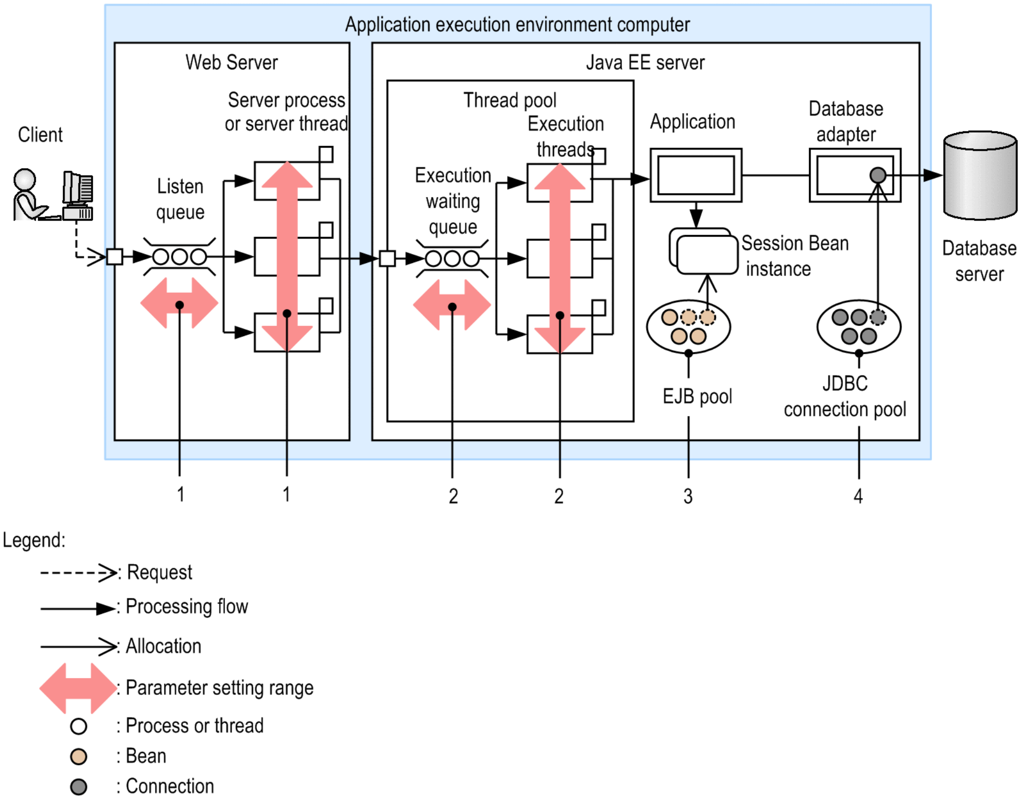
+ a indicates that the maximum number of concurrent connections for DBMS might temporarily exceed the size of the JDBC connection pool. When the connection error detection function is being used, if a new connection is established before a failed connection is discarded, the maximum number of concurrent connections for DBMS might temporarily exceed the connection pool size. Note that the maximum value of + a is the same size as the connection pool size.

## 3.6 Flow control

Flow control controls the number of requests processed by a system. This provides stable operation of the system even when request processing is congested. By controlling the maximum number of requests executed by a Java EE server at one time, or the number of beans to be pooled in the EJB pool, or the number of JDBC connections to be pooled in the JDBC connection pool, the load on the Java EE server can be balanced, and stable throughput can be attained.

### Flow control mechanism

The following figure illustrates the flow of request processing and the flow control targets.



The table below lists the resources subject to control and the parameters to be set. The numbers in the above figure correspond to the numbers in the following table.

| No. | Resources subject to control | Description | Parameter to set |
|---|---|---|---|
| 1 | Web Server server process or server thread | The main role of the Web Server server process is to forward requests to Java EE servers. This server process is used exclusively until request processing on Java EE servers is complete.<br>You can change the maximum number of server process and control the maximum number of connections. One server process is allocated to one request. In addition, you can change the size of the queue where requests that cannot be processed are stored. | • Queue size:<br>`ListenBackLog`<br>• Maximum number of concurrent connections:<br>`MaxRequestWorkers`<br>• Maximum number of running server processes:<br>`HWSKeepStartServers`<br>`MinSpareServers`<br>`MaxSpareServers`<br>`StartServers` |

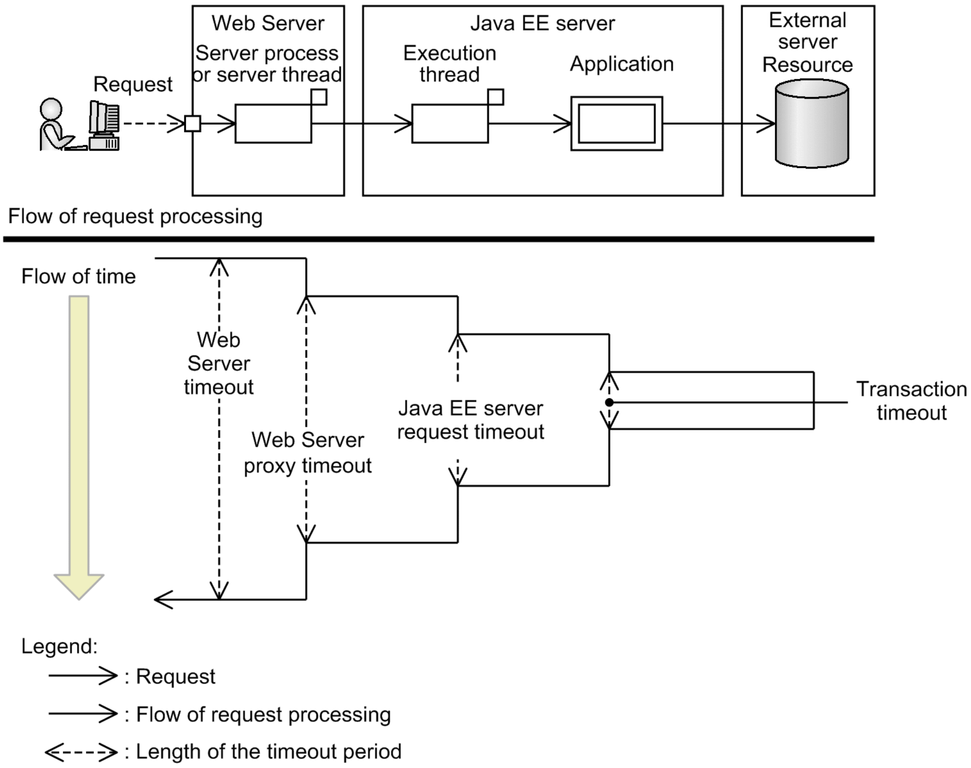
| No. | Resources subject to control | Description | Parameter to set |
|---|---|---|---|
| 2 | Java EE server execution thread pool | The Java EE server execution thread interprets requests, executes web application programs, and constructs responses.<br>You can change the maximum number of threads, and control the maximum number of execution requests. One thread is allocated to one request. In addition, you can change the size of the queue where requests that cannot be processed are stored. | - Maximum number of execution threads:<br>configs.config.*configuration_name_of_the_Java_EE_server*.thread-pools.thread-pool.*thread_pool_ID*.max-thread-pool-size<br>- Queue size:<br>configs.config.*configuration_name_of_the_Java_EE_server*.thread-pools.thread-pool.*thread_pool_name*.max-queue-size |
| 3 | EJB pool | You can change the maximum number of beans to be pooled, and control the number of threads to be executed concurrently. One session bean is allocated to one request.<br>Note that when the number of beans in use reaches the maximum, and a new bean is needed, the client has to wait until a bean becomes available. As a result, requests are queued. | Maximum pool size:<br>configs.config.*configuration_name_of_the_Java_EE_server*.ejb-container.max-pool-size |
| 4 | JDBC connection pool | You can control the maximum number of JDBC connections to be pooled.<br>Note that when the number of connections in use reaches the maximum, and a new connection is needed, the client has to wait until a connection becomes available. As a result, requests are queued. | Maximum pool size:<br>resources.jdbc-connection-pool.*JDBC_connection_pool_ID*.max-pool-size |

# 3.7 Overview of timeout control

Timeout control means setting the timeout for a processing. If the timeout is set, processing is canceled when the processing time exceeds the specified time. This prevents unintended use of resources by a system engineer, and avoids any slowdown of the system for business system users.

## Timeout control mechanism

The following figure shows the relationship between the flow of request processing and the timeout period.



Because all call relationships in this figure are synchronized, the nearer the distance to the client, the longer the execution time. Therefore, in order to help identify where a timeout occurred, make sure the timeout value specified for a caller is greater than the timeout value of a called target. In other words, the timeout settings must be as follows: Web Server timeout > Web Server proxy timeout > Java EE server request timeout > transaction timeout.

In addition to the above case, you can also set the connection timeout and request timeout for callers and called targets when you use JAX-RS or JAX-WS to link multiple business programs.

The following describes the timeouts that can be set for Application Server.

Server processing timeouts

The following table describes what is controlled by a server processing timeout and the parameters to be specified.

| No. | Item | Controlled item | Control target | Parameter to be specified |
|---|---|---|---|---|
| 1 | Web Server timeout | Wait time for sending data to and receiving data from the client | Web Server server process or server thread | `Timeout` |

| No. | Item | Controlled item | Control target | Parameter to be specified |
|---|---|---|---|---|
| 2 | Web Server proxy timeout | Amount of time that elapses before a response is received after a request was sent to the Java EE server | Web Server server process or server thread | `ProxyTimeout` |
| 3 | Java EE server request timeout | Application execution time | Java EE server execution thread | `configs.config.`*`configuration_name`*`.network-config.protocols.protocol.`*`configuration_name`*`.http.request-timeout-seconds` |
| 4 | Transaction timeout | Transaction processing time | Transaction service | `configs.config.`*`configuration_name`*`.transaction-service.timeout-in-seconds` |
| 5 | JAX-WS timeout | Web service execution time | JAX-WS and WebService | `configs.config.`*`configuration_name`*`.hitachi-jaxws-config.request-timeout` |
| 6 | JDBC SQL timeout | SQL execution time | JDBC connection pool | Use one of the following to specify the timeout:<br>• `--statementtimeout` option of the `create-jdbc-connection-pool` subcommand<br>• `statement-timeout-in-seconds` attribute of `/resources/jdbc-connection-pool` in `glassfish-resources.xml` |

## Important note

How to deal with a transaction timeout

When a transaction times out, the system operator should examine the message KDKD20033-W output to the log, and remove the cause of the timeout.

The message KDKD20033-W is output only when a local transaction times out.

Connection timeout

The following table describes what is controlled by the connection timeout and the parameters to be specified.

| No. | Item | Controlled item | Control target | Parameter to be specified |
|---|---|---|---|---|
| 1 | JAX-WS connection timeout | Amount of time required to establish a connection with the web service | JAX-WS and WebService | `configs.config.`*`configuration_name`*`.hitachi-jaxws-config.connect-timeout` |
| 2 | JAX-RS connection timeout | Amount of time required to establish a connection with the web service | JAX-RS and Resource | `configs.config.`*`configuration_name`*`.hitachi-jaxrs-config.connect-timeout` |
| 3 | JDBC connection collection timeout | Maximum wait time for collecting a connection from the JDBC connection pool | JDBC connection pool | Use one of the following to specify the timeout:<br>• `--maxwait` option of the `create-jdbc-connection-pool` subcommand<br>• `max-wait-time-in-millis` attribute of `/resources/jdbc-connection-pool` in `glassfish-resources.xml` |

Subcommand timeouts for the Java EE server

Processing of subcommands of the `asadmin` command requires HTTP communication between processes such as the process of the `asadmin` utility command and the domain administration server process. If the time required for inter-process communication or processing of a subcommand of the `asadmin` utility command exceeds a preset time, a timeout occurs. The subcommands of the `asadmin` utility command include local and remote subcommands. The following describes the timeout for each type of subcommand.

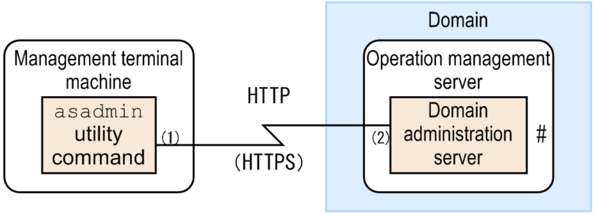
For local subcommands

A communication timeout does not occur because local subcommands are processed on the host on which the `asadmin` utility command is executed. However, for some subcommands for which the command processing timeout can be set, a timeout occurs if the specified timeout value is exceeded.

For remote subcommands

A timeout might occur because remote subcommands are processed on the domain administration server, which involves communication. The following figures show where a timeout might occur when a remote subcommand is executed.

Where a timeout of a remote subcommand might occur (without server instance execution)


Domain
Management terminal machine
asadmin utility command
(1)
HTTP
(HTTPS)
Operation management server
(2)
Domain administration server
#

Legend:

▭ : Process

(1) Where an `asadmin` utility command read timeout occurs

(2) Where a domain administration server request timeout occurs

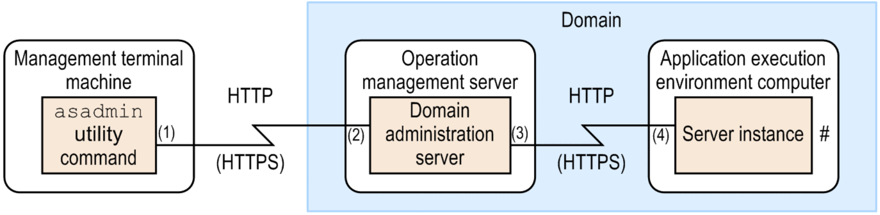
#: A timeout occurs here if the command processing timeout is individually specified for the `asadmin` utility command.

When a remote subcommand is executed, the process of the `asadmin` utility command transfers the subcommand processing to the domain administration server. At this time, the timeout is set for the following processing:

- HTTP-response read processing set for communication from the process of the `asadmin` utility command to the domain administration server

- Processing of the HTTP request containing the subcommand received from the process of the `asadmin` utility command

Where a timeout of a remote subcommand might occur (with server instance execution)


Domain
Management terminal machine
asadmin utility command
(1)
HTTP
(HTTPS)
Operation management server
(2)
Domain administration server
(3)
HTTP
(HTTPS)
Application execution environment computer
(4)
Server instance
#

Legend:

▭ : Process

(1) Where an `asadmin` utility command read timeout occurs

(2) Where a domain administration server request timeout occurs

(3) Where a domain administration server read timeout occurs

(4) Where an server instance request timeout occurs

#: A timeout occurs here if the command processing timeout is individually specified for the `asadmin` utility command.

When a remote subcommand is executed, the process of the `asadmin` utility command transfers the subcommand processing to the domain administration server. The domain administration server processes the subcommand received from the process of the `asadmin` utility command. Then, depending on the command and user-specified options, the same subcommand is also run on the server instance. At this time, the timeout is set for the following processing:

- HTTP-response read processing set for communication from the process of the `asadmin` utility command to the domain administration server

- Processing of the HTTP request containing the subcommand received from the process of the `asadmin` utility command

- HTTP-response read processing set for communication from the domain administration server to the server instance

- Processing of the HTTP request containing the subcommand received from the domain administration server

The following table shows subcommand timeout details.

Table 3-1: Subcommand timeout list

| No. | Name | Process that causes a timeout | Target processing | Operation when a timeout occurs | How to set the timeout |
|---|---|---|---|---|---|
| 1 | `asadmin` utility command read timeout | `asadmin` utility command process | HTTP-response read processing set for communication from the `asadmin` utility command process to the domain administration server | • Execution of the `asadmin` utility command fails, but subcommand processing continues on the domain administration server. A message indicating the command execution result on the domain administration server (success: KDKD10162-I, failure: KDKD10163-E) is output to the message log of the domain administration server host.#1<br>• A message indicating a timeout is displayed on the standard output of the `asadmin` utility command process.#2<br>• An error message (KDKD10164-E) is output to the message log of the `asadmin` utility command. | Specify the timeout value (in milliseconds) for the environment variable `AS_ADMIN_READTIMEOUT` for the host on which the `asadmin` utility command is run.#3 |
| 2 | Domain administration server request timeout | Domain administration server process | Processing of the HTTP request containing the subcommand received from the `asadmin` utility command | • Execution of the `asadmin` utility command fails, but subcommand processing itself might continue without causing an error.#4<br>• A message indicating a subcommand error is displayed in the standard output of the `asadmin` utility command process.#5<br>• An error message (KDKD10165-E) is output to the message log of the domain administration server. | Set the timeout (in seconds) in `set parameter (configs.config.`*`configuration_name_of_the_Java_EE_server`*`.network-config.protocols.protocol.`*`listener_name`*`.http.request-timeout-seconds)` .<br>*`listener_name`*:<br>`admin-listener` (if `secure admin` is disabled)<br>`sec-admin-listener` (if `secure admin` is enabled) |
| 3 | Domain administration server read timeout | Domain administration server process | HTTP-response read processing set for communication from the domain administration server to the server instance | • Execution of the `asadmin` utility command fails, but subcommand processing continues on the server instance. A message indicating the subcommand execution result on the server instance (success: KDKD10162-I, failure: KDKD10163-E) is output to the server instance message log.#1<br>• A message indicating a subcommand error is displayed on the standard output of the `asadmin` utility command process.#5<br>• An error message (KDKD10164-E) is output to the message log of the domain administration server. | Specify the timeout value (in milliseconds) for the environment variable `AS_ADMIN_READTIMEOUT` for the domain administration server host. #5 |

| No. | Name | Process that causes a timeout | Target processing | Operation when a timeout occurs | How to set the timeout |
|---|---|---|---|---|---|
| 4 | Server instance request timeout | Server instance process | Processing of the HTTP request containing the subcommand received from the domain administration server | - Execution of the `asadmin` utility command fails, but subcommand processing itself might continue without causing an error.#4<br>- A message indicating a subcommand error is displayed in the standard output of the `asadmin` utility command process.#5<br>- An error message (KDKD10165-E) is output to the message log of the domain administration server. | Set the timeout (in seconds) in `set parameter (configs.config.`*`configuration_name_of_the_Java_EE_server`*`.network-config.protocols.protocol.sec-admin-listener.http.request-timeout-seconds)`. |

#1:

Some of the subcommands output execution result logs, but others do not. For details on whether a subcommand outputs execution result logs, see the table of subcommands that require a long processing time.

#2:

The following message is output:

```
No response from Domain Admin Server after timeout_value seconds.
The command is either taking too long to complete or the server has failed.
Please see the server log files for command status.
Command command_name failed.
```

#3:

The values specified for the `asadmin` utility command read timeout and domain administration server read timeout are applied to all subcommands. If a timeout occurs when the default timeout values are used, you need to increase the timeout values according to the processing time of the most time-consuming command.

#4:

For details on the subcommands that continue processing if a timeout occurs, see the table of subcommands that require a long processing time.

#5:

The following message is output:

```
Command command_name failed.
```

The following table shows subcommands that require a long processing time:

### Table 3-2: Subcommands that require a long processing time

| No | Command name | Condition | | | | Execution result logs | Consequences of the request timeout |
|---|---|---|---|---|---|---|---|
| | | Number of applications to be deployed | Size of applications to be deployed | Time to wait until the request is finished | Size and number of files to be collected | | |
| 1 | `start-domain` | Y | -- | -- | -- | The command does not output logs. | The consequences depend on the implementation. |

| No | Command name | Condition | | | | Execution result logs | Consequences of the request timeout |
|---|---|---|---|---|---|---|---|
| | | Number of applications to be deployed | Size of applications to be deployed | Time to wait until the request is finished | Size and number of files to be collected | | |
| 2 | `stop-domain` | Y | -- | Y | -- | The command does not output logs. | The consequences depend on the implementation. |
| 3 | `restart-domain` | Y | -- | -- | -- | The command does not output logs. | The consequences depend on the implementation. |
| 4 | `start-instance` | Y | -- | -- | -- | The command outputs logs. | The consequences depend on the implementation. |
| 5 | `stop-instance` | Y | -- | Y | -- | The command does not output logs. | The consequences depend on the implementation. |
| 6 | `start-cluster` | Y | -- | -- | -- | The command does not output logs. | The consequences depend on the implementation. |
| 7 | `stop-cluster` | Y | -- | -- | -- | The command does not output logs. | The consequences depend on the implementation. |
| 8 | `start-servers` | Y | -- | -- | -- | The command does not output logs. | The command processing continues. |
| 9 | `stop-servers` | Y | -- | Y | -- | The command does not output logs. | The command processing continues. |
| 10 | `deploy` | -- | Y | -- | -- | The command does not output logs. | The consequences depend on the implementation. |
| 11 | `undeploy` | -- | Y | Y | -- | The command does not output logs. | The consequences depend on the implementation. |
| 12 | `enable` | -- | Y | -- | -- | The command does not output logs. | The consequences depend on the implementation. |
| 13 | `disable` | -- | -- | Y | -- | The command does not output logs. | The consequences depend on the implementation. |
| 14 | `collect-snapshot` | -- | -- | -- | Y | The command does not output logs. | The command processing continues. |

Legend:

Y: The subcommand processing time changes depending on the given conditions.

--: The subcommand processing time does not change.

### Reference note

Subcommands that require a long processing time might time out, because the processing time of the commands might increase depending on given conditions, such as the number of applications to be deployed.

## 3.8 Overview of security measures

Security measures for a system using Application Server use functions such as encryption and user authentication to protect the data and system from threats such as data falsification and hacking. In addition to the Application Server security functions, security measures that define how to use the system to protect the data and system from threats are also required.

### Security policies

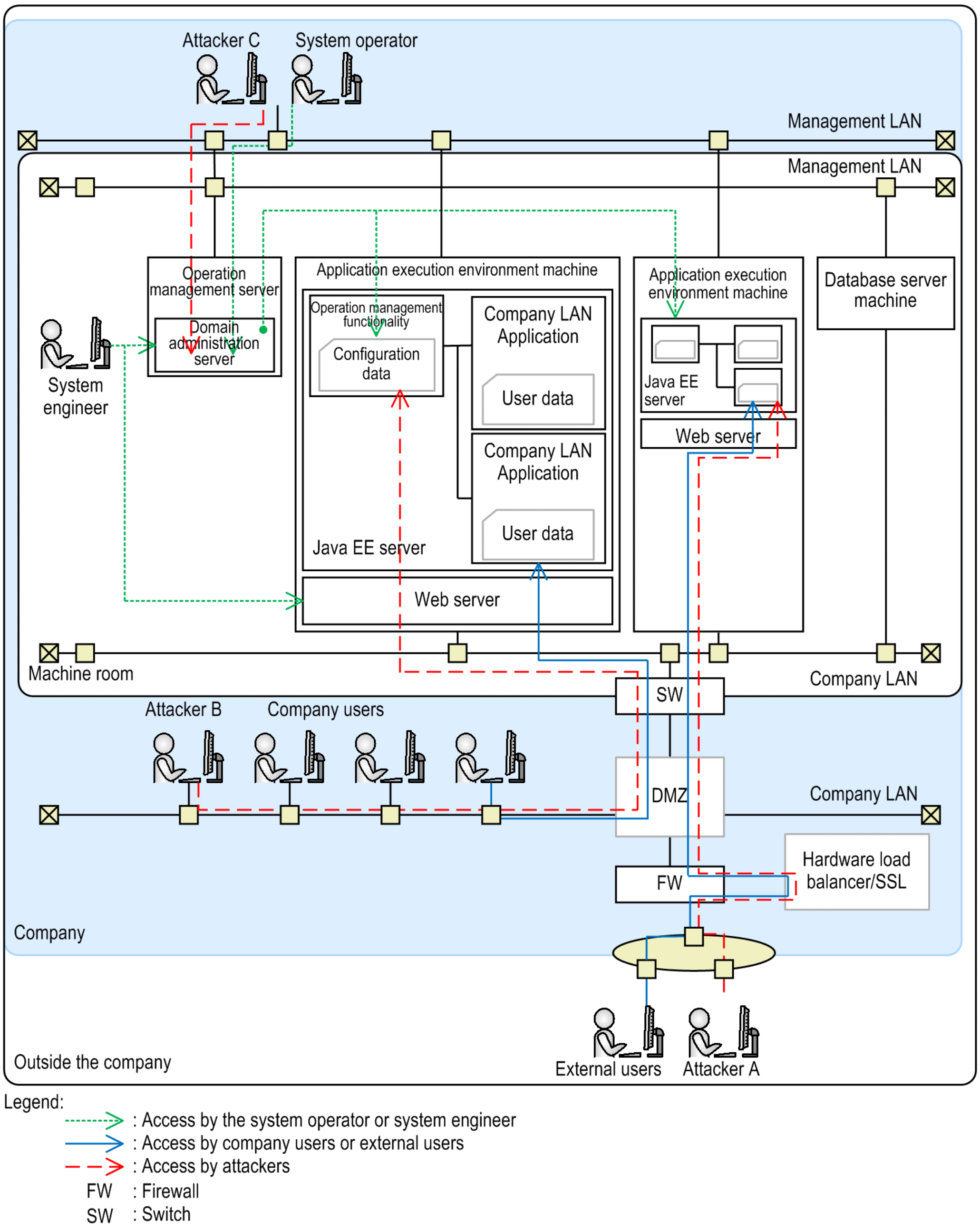
Based on the standard system configuration using Application Server, the following describes possible threats in this configuration and policies for taking measures against those threats. Note that security information for applications (UAPs) that run on Application Server is not covered here. Such information must be determined by the specific application developers.

### Standard system configuration using Application Server, and possible threats

The following shows an example standard system configuration using Application Server, and describes the operation scenario and possible threats.

Standard system configuration using Application Server

The following figure shows a standard system configuration using Application Server.



System configuration elements

The following figure shows configuration elements of the standard system using Application Server.

### Table 3-3: System configuration elements

| No. | Element | Description |
|---|---|---|
| 1 | Company | An area where company users and system operators work. A company LAN and a management LAN are installed in this area. They are physically separated from each other. It is assumed that the management LANs in the company and company LANs are physically separated from outside the company. |
| 2 | Machine room | An area where the system engineer works In order to ensure security, the operation management server machine, application execution environment computers, and database server machines are installed.<br>This area is assumed to have the highest security level in the company and therefore requires strict management of room access. |
| 3 | Company LAN | A LAN used to exchange business data between company users' PCs and applications |

| No. | Element | Description |
|---|---|---|
| 4 | Management LAN | A LAN used to exchange management data between the system operator's PC and the domain administration server |
| 5 | DMZ | An isolated network area established between the company LAN and the WWW. |
| 6 | Domain administration server | A server that manages operation of multiple application execution environment computers (on which Application Server is installed) by grouping them on the basis of domain. |
| 7 | Operation management server | A machine on which the domain administration server is installed. This machine manages operation of application execution environment computers. |
| 8 | Application execution environment computer | A machine on which Application Server is installed. This machine manages applications. |
| 9 | Database server machine | A server on which DBMS operates. This server is assumed to be used by applications.<br>Note that the database security function is not managed by Application Server. |
| 10 | Switch | Provides routing control using IP addresses. |
| 11 | Firewall | Provides access control using IP addresses and port numbers. |
| 12 | Hardware load balancer/SSL | Distributes requests to application execution environment computers. Because SSL communication is supported, encrypted communication is also possible. |

System operation scenario

The following describes an operation scenario based on the standard system configuration diagram using Application Server.

1. An external user uses the web browser installed on the PC to access applications via HTTP or WebSocket, and uses the system via the Internet.

   Note that the external user is assumed not to use Web services (JAX-WS or JAX-RS) or RMI/RMI-IIOP to attempt access.

2. Company users access applications from within the company via HTTP (including SOAP/REST) or WebSocket by using the web browser installed on business PCs or an application client, and then execute jobs via the company LAN.

   Note that the company users are assumed not to use RMI/RMI-IIOP to attempt access.

3. The system operator accesses the domain administration server from within the company by using the web browser installed on a management PC or Application Server management commands, and then manages operation via the management LAN.

4. The system engineer directly accesses the operation management server, application execution environment computer, and database server machine to perform setup and change the settings.

5. The system engineer accesses the domain administration server from the machine room by using the web browser installed on a management PC or Application Server management commands, and then manages multiple application execution environment computers.

6. The domain administration server sends instructions received from the system operator or system engineer to each application execution environment computer in order to start, stop, and change settings of Application Server processes on the application execution environment computers.

Possible attacks to the system

The following describes the threats of attackers to the system that uses Application Server.

1. Attacker A directly attacks application execution environment computers on the Internet.
2. Attacker B attacks applications by using unauthorized access from a business PC connected to the company LAN.
3. Attacker C impersonates the system operator to perform unauthorized access to the system from a business PC connected to the company LAN, and then attacks the domain administration server and application execution environment computers.

## Security policies against possible threats to the system

The following describes possible security policies for the system using Application Server.

Assuming network security policies

1. When you design the network security for layer 3 or lower, consider the following:

   Use the hardware load balancer or firewall to restrict packets from outside the company to access to ports for normal use only.

   Prevent packets from the company LAN and from outside the company from being sent to the management LAN.
2. When you design the network security for layer 4 or higher, consider the following:

   Support the security within the scope of the Java EE standard specifications, and let the application developer be responsible for other security issues.
3. When you design the network security for the management LAN, consider the following:

   Although the management LAN is independent of the company LAN, prevent attackers from connecting to the management LAN regardless of any malicious intent.

Assuming security policies for physical operations

1. Consider the following when you design the security that applies outside the company:

   Because there might be malicious persons outside the company, prevent a physical connection with the company LAN or management LAN from being established from outside the company.
2. When you design the security that applies within the company, consider the following:

   Because there are malicious attackers, allow physical access to Application Server only via the network
3. When you design the machine room area, consider the following:

   Prevent attackers from entering the area regardless of malicious intent.

Assuming security policies for the development environment

1. Use a machine in the company LAN to develop applications.
2. Make sure there is no possibility that a developer can include malicious code in applications.
3. An expert must review the developed applications.

   Confirm that the developed applications are free of any viruses.
4. The system engineer must deploy the applications (developed in the company LAN) on application execution environment computers via the management LAN.

   Do not allow the application developer to directly deploy the applications on an application execution environment computers in the management LAN.

Application Server security policies

The following describes the Application Server security policies against possible threats.

1. Make sure that access from the company LAN to applications is used within the scope of the authority specified by the system engineer.

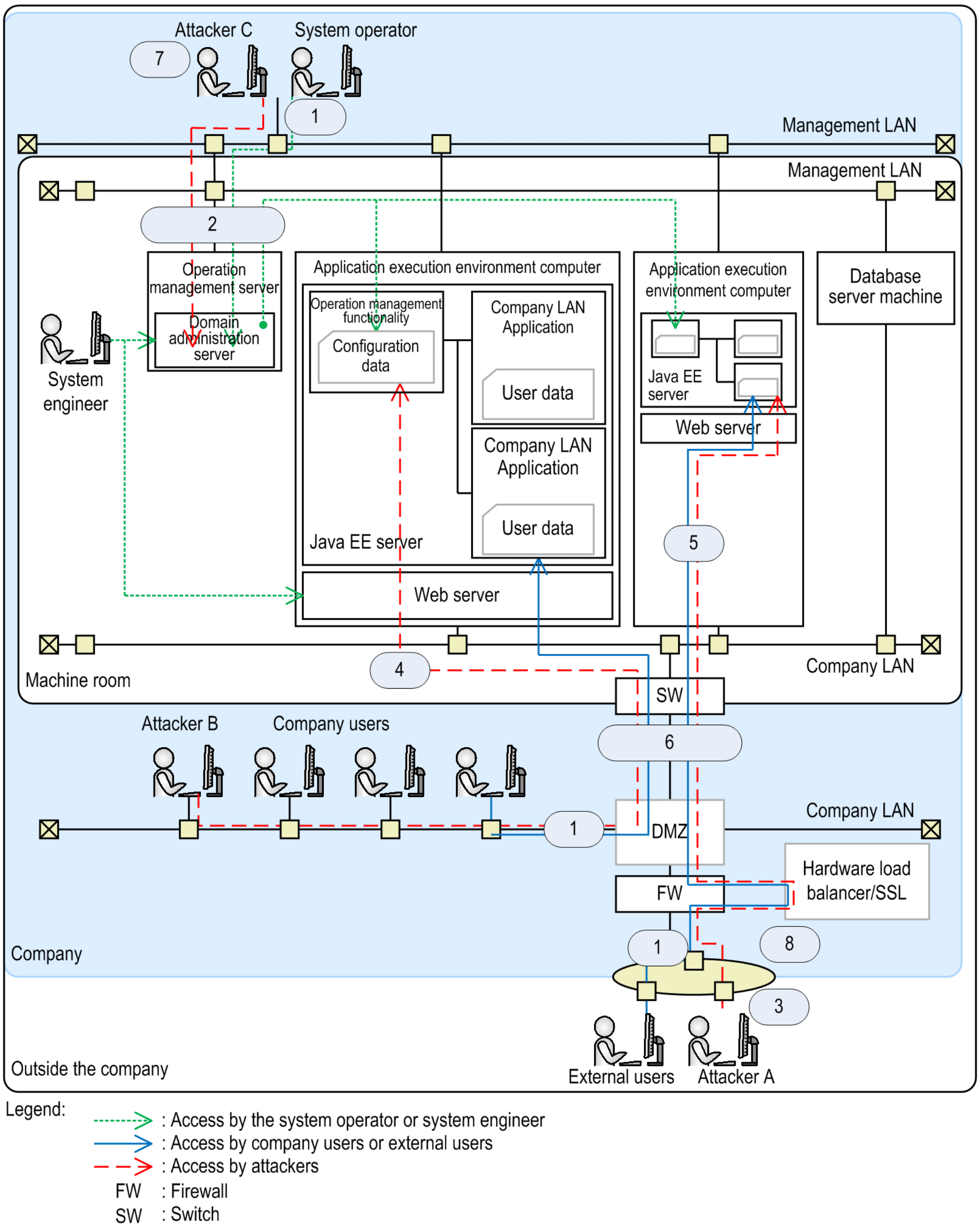
   Make sure that access from the company LAN to applications does not deviate from the authority specified by the system engineer.

   Make sure that file access deviating from the specified authority does not occur.
2. Support encryption of access from the company LAN to applications in order to prevent wiretapping and falsification.
3. For access from the company LAN to applications, collect the application log and Application Server log separately.
4. For access from the company LAN to applications, provide the authentication functionality within the scope of the Java EE standard specifications.
5. Do not guarantee the security of ports used by the operation management functionality.
6. Do not guarantee the tamper resistance of files and memory for Application Server.
7. Do not restrict the output of data that must be protected in files and memory for Application Server.
8. If output from Application Server exceeds the network, security-related configuration data (such as the `domain.xml` configuration file for the Java EE server and a password management file) must be protected.

## Threats to the system and countermeasures

The following describes possible security threats to the system based on the standard system configuration diagram using Application Server.

Numbers in the figure indicate the locations that might be subject to threats.



The table below describes the countermeasures against threats to the system. The numbers in the figure correspond to the item numbers in the table.

| No. | Threats to the system | Overview of threats | Application Server functions and operations used for countermeasures | Security function settings |
|---|---|---|---|---|
| 1 | Peeking and falsification of communication data | An attacker illegally browses business data and management data by unauthorized acquisition of packets flowing in the network, or illegally changes data by altering a packet itself. | Function<br>Use the SSL and TLS encryption functions of Web Server.<br>Operation<br>Use the SSL and TLS encryption functions of the load balancer. | In the `httpsd.conf` file that configures the operating environment of the web server, specify the encryption function for SSL authentication. |
| 2 | Falsification of | An attacker alters the packets flowing in the network to change the instructions issued by a | Operation<br>Implement access control for the management LAN. | -- |

| No. | Threats to the system | Overview of threats | Application Server functions and operations used for countermeasures | Security function settings |
|---|---|---|---|---|
| | instructions | system engineer or system operator. | | |
| 3 | Application access by unauthorized users | An attacker without access authority sends a request to applications and executes jobs illegally. | Function<br>Use the Java EE standard security roles. | Use the `create-auth-realm` subcommand to create a named authentication realm, and then define the security role in the application DD. |
| 4 | Execution of instructions by unauthorized users | An attacker without access authority accesses the operation management function and executes instructions illegally. | Function<br>Use the management password of the domain administration server for authentication.<br>Operation<br>Only save and store the user authentication data (such as passwords) on a device that is connected to the management LAN only.<br>Permit access to management ports of the domain administration server only from the management LAN. | When creating a domain by using the `create-domain` subcommand, specify the management user password. Use the `change-admin-password` subcommand to change the management user password. |
| 5 | Chain-reaction effect of the vulnerability of applications on the whole Java EE server | A vulnerability that is revealed by an unexpected message received by a specific application affects the whole Java EE server in a chain reaction manner. | Function<br>Use the Java security policies to protect the system (by restricting the use of API). | Specify the location of the system properties (in the `server.policy` file) provided by Java SE. |
| 6 | Denial of unauthorized access | An attacker who attempts unauthorized access denies the fact of unauthorized access. | Function<br>Set Web Server access logging. | In the `httpsd.conf` file that configures the operating environment of the web server, set web server access logging. |
| 7 | Shoulder hacking | An attacker acquires information, such as a password, by looking over the shoulder of the system engineer or system operator. | Function<br>In order to prevent critical information such as passwords from being viewed, use a function that hides the characters entered in the text box and displays black dots or other characters instead. | -- |
| 8 | Denial of service (DoS) attack | A type of attack designed to make services unavailable | Function<br>Use the functions for timeout control and packet size restriction.<br>Operation<br>Link with JP1 to monitor security errors.<br>Other countermeasures<br>You need to use network products such as a firewall and load balancer to reduce DoS attack access. | In the `httpsd.conf` file that configures the operating environment of the web server, specify timeout control and packet size restriction. |

### Related topics

- 6.2.1 Setting up reverse proxies
- 6.2.2 Setting up SSL

# 3.9 Application authentication usable on Application Server

Application authentication is a general term for the user authentication methods used to ensure security. Client authentication, BASIC authentication, FORM authentication, and DIGEST authentication are available as application authentication methods usable on Application Server.

## Application authentication categories

The following table lists application authentication methods usable on Application Server. If one authentication method can authenticate multiple locations, specify one of the locations.

| No. | Authentication method | Locations where authentication is performed | Authentication mechanism |
|---|---|---|---|
| 1 | Client authentication | Web Server | SSL |
| 2 | | SSL accelerator | |
| 3 | BASIC authentication | Web Server | Password file |
| 4 | | | LDAP repository |
| 5 | | Java EE server | JAAS |
| 6 | FORM authentication | Java EE server | JAAS |
| 7 | DIGEST authentication | Java EE server | |

# 3.10 Estimating resources

Estimate the total resource requirements by summing the estimated shared memory size, number of processes, threads, and file descriptors, as well as the estimated database capacity.

## Estimating the size of the shared memory

The following describes how to estimate the size of the shared memory used by the web server and the performance tracer.

Size of the shared memory used by the web server

The size of the shared memory used by the web server is calculated as described below.

Note that the size of the shared memory used by the web server is the same as the size of shared memory segments.

```
Size of shared memory used by web server = 409,600 + 7,168 * MaxRequestWorkers
```

Legend:

`MaxRequestWorkers`: Maximum number (bytes) of clients that can be connected concurrently

Size of the shared memory used by the performance tracer

The size of the shared memory used by the performance tracer is calculated as described below.

Note that the size of the shared memory used by the performance tracer is the same as the size of shared memory segments.

```
Size of shared memory used by performance tracer =
value_specified_for_PrfTraceBufferSize * 1,024 + 18,496
```

Legend:

*value_specified_for_PrfTraceBufferSize*: Trace buffer size (bytes) for the performance tracer retained in the shared memory

## Estimating the number of processes

The following describes how to estimate the number of processes for the Java EE server, the web server, and performance tracer.

Number of Java EE server processes

The number of resident Java EE server processes is calculated as follows:

```
Number of Java EE server resident processes = number_of_domain_management_servers
 + number_of_server_instances
```

Number of web server processes

The number of web server processes is calculated as follows:

```
Number of web server processes = control_processes + server_processes
 + CGI_processes + gcache_server
 + rotatelogs_processes + rotatelogs2_processes
```

Legend:

*control_processes*: 1

*server_processes*: Number of the specified `MaxRequestWorkers` directives

*CGI_processes*: Number of server processes

*gcache_server*: 1

*rotatelogs_processes*: Number of rotatelogs programs specified for the `CustomLog` directive, the `ErrorLog` directive, the `HWSRequestLog` directive, and the `TransferLog` directive

*rotatelogs2_processes*: Number of rotatelogs2 programs specified for the `CustomLog` directive, the `ErrorLog` directive, the `HWSRequestLog` directive, and the `TransferLog` directive

Number of performance tracer processes

The number of resident performance tracer processes is 1.

## Estimating the number of threads

The following describes how to estimate the numbers of threads for the Java EE server, the web server, and performance tracer.

Number of Java EE server threads

The number of threads to be used by the Java EE server is calculated as follows:

$$\text{Total number of threads to be used by the Java EE server} = \sum_{i=1}^{m} \text{Dt}(i) + \sum_{i=1}^{n} \text{SIt}(i)$$

Legend:

*Dt*: Total number of threads for each domain management server process

*SIt*: Total number of threads for each server instance process

*m*: Number of domain management server processes

*n*: Number of server instance processes

To calculate the total number of threads for each domain management server process (*Dt*) and the total number of threads for each server instance process (*SIt*), use one of the following formulas:

$$\text{Total number of threads for each domain administration server process} = \sum_{i=1}^{m} \text{Tm}(i) + \text{Tn}$$

$$\text{Total number of threads for each server instance process} = \sum_{i=1}^{n} \text{Tm}(i) + \text{Tn}$$

Legend:

*Tm*: Maximum value for thread pools

*Tn*: Number of threads that do not belong to thread pools. Among these threads, 44 threads are created by default by, for example, the domain administration server and server instance.

*m*: Number of thread pools for domain management servers

*n*: Number of thread pools for server instances

Number of web servers threads

The number of threads used by the web server is the same as the number of processes used by the web server.

Number of performance tracer threads

The number of threads used by performance tracer is 10.

## Estimating the number of file descriptors

This subsection describes how to estimate the number of file descriptors for the Java EE server, web server, and performance tracer.

Number of file descriptors for the Java EE server

The number of file descriptors for the Java EE server increases or decreases according to the issuance of OS system calls such as `open()` and `socket()`. In addition, the number of file descriptors varies depending on the number of concurrent connections of DBMS.

The method for estimating the number of file descriptors used by the Java EE server based on the maximum number of concurrent connections of DBMS is described here.

To calculate the number of file descriptors used by the Java EE server, use the following formula:

```
Number of Java EE server file descriptors = JDBC_connection_pool_size x 2
```

Number of file descriptors for the web server

To calculate the number of file descriptors used by the web server, use the following formula:

- For Linux

```
Number of file descriptors used by the web server = (50 + A x B + C + 11 x C x
D
+ 8 x E + (F + H) x G) x 1.2
```

Legend:

*A*: Number of the specified `Listen` directives (If the `Listen` directive is not specified, use the value 1.)

*B*: Number of IP addresses assigned to the host

*C*: Total number of the `CustomLog`, `ErrorLog`, `HWSRequestLog`, and `TransferLog` directives that you specified

*D*: Use 1 if you use the rotatelogs or rotatelogs2 program, and use 0 if you do not use either program.

*F*: Set 3 if you use SSL, and set 2 if you do not use SSL.

*G*: Number of concurrent execution requests (This is the value specified for `MaxRequestWorkers`.)

*H*: Set 1 if you use the reverse proxy, and set 0 if you do not use the reverse proxy.

Note that the number of file descriptors used in external modules that are not contained in CGI programs or the web server is not included.

Number of file descriptors for performance tracer

Set 32 or a greater value for the number of file descriptors used when performance tracer starts.

## Estimating database capacity

If you want to use the EJB timer service and Java Batch as functions of Application Server, DBMS is required. DBMS stores the DDL file where table information is defined. Based on the table information of this DDL file, estimate the database capacity.

The following table shows where the DDL files necessary for estimation are stored.

Table 3-4: DDL file locations

| No. | Target | Storage directory | File name | DBMS |
|---|---|---|---|---|
| 1 | EJB timer service | `installation_directory_for_Application_Server/javaee/glassfish/lib/install/databases` | `ejbtimer_oracle.sql` | Oracle |

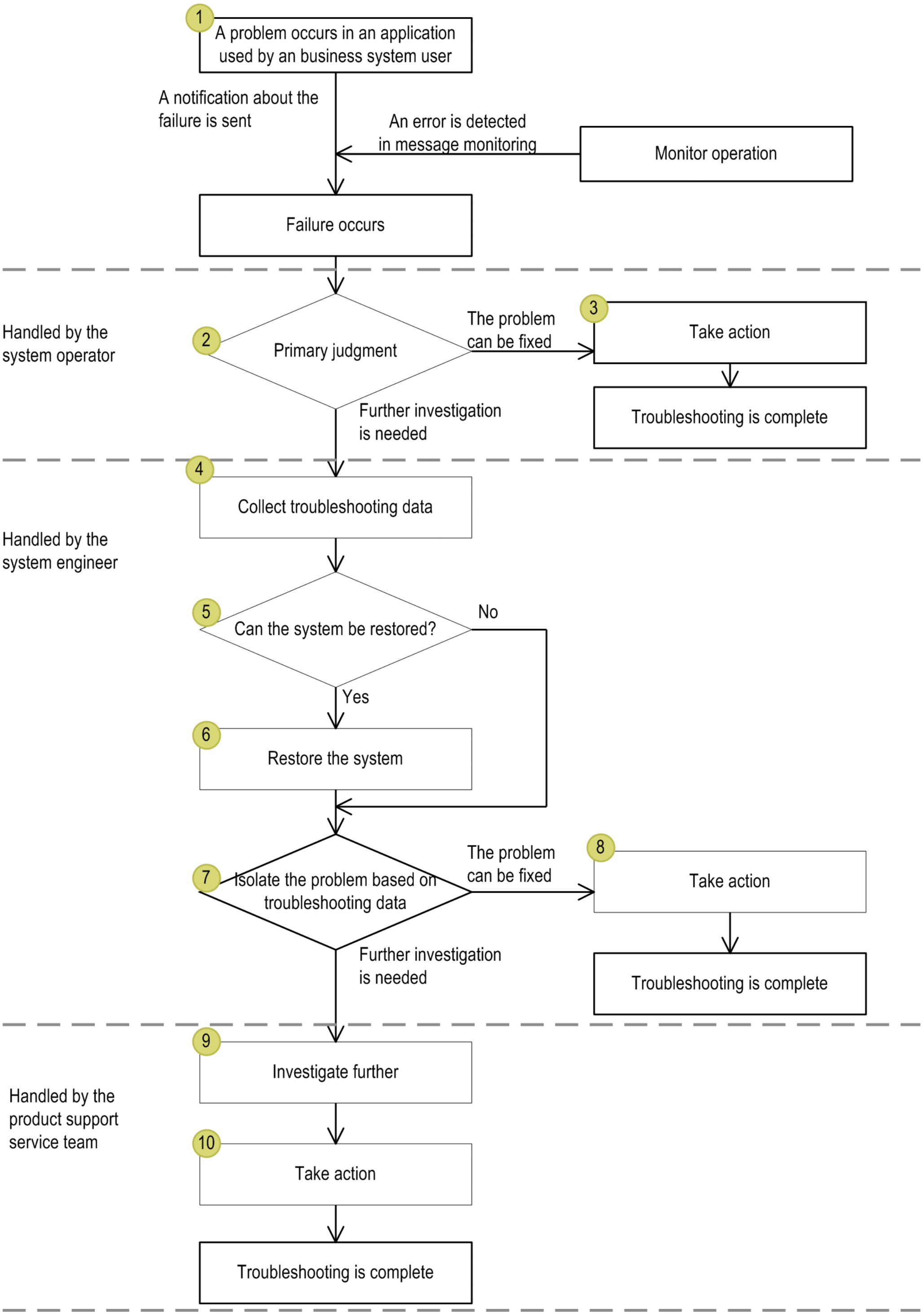
# 3.11 Troubleshooting workflow

When a failure occurs in an Application Server system, investigate and resolve the failure according to the troubleshooting workflow. Note that the workflow to be used when the system is running is different from the workflow to be used when the system is being constructed.

## Troubleshooting workflow when the system is running

The troubleshooting workflow when the system is running is described here.

The following figure shows the troubleshooting workflow for a failure that occurred while the Application Server system was running.

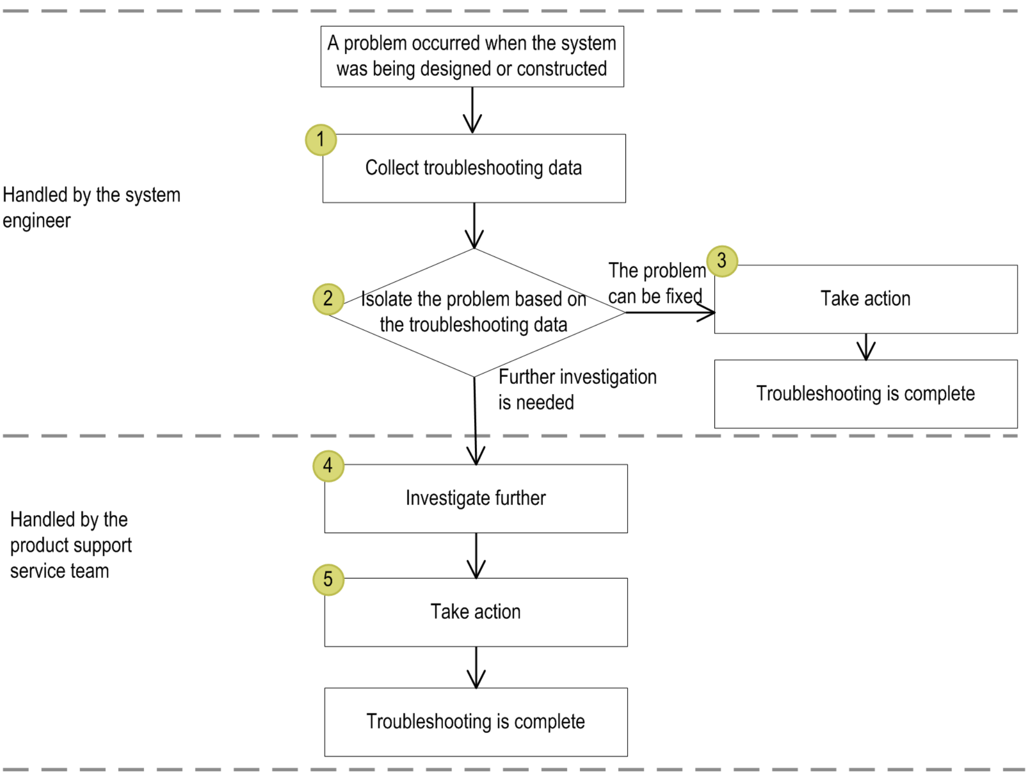

1
A problem occurs in an application used by an business system user
A notification about the failure is sent
An error is detected in message monitoring
Monitor operation
Failure occurs
Handled by the system operator
2
Primary judgment
The problem can be fixed
3
Take action
Troubleshooting is complete
Further investigation is needed
4
Collect troubleshooting data
Handled by the system engineer
5
Can the system be restored?
No
Yes
6
Restore the system
7
Isolate the problem based on troubleshooting data
The problem can be fixed
8
Take action
Troubleshooting is complete
Further investigation is needed
9
Investigate further
Handled by the product support service team
10
Take action
Troubleshooting is complete

1. When a problem occurs in an application being used by an business system user, the user notifies the system operator of the failure. A message indicating the problem is output to the message monitoring functionality of Application Server, and then the problem is reported to the system operator.
2. Upon receiving the notification, the system operator performs the primary judgment.
3. If the primary judgment result indicates that the problem can be fixed, the system operator takes the appropriate actions.
4. If the primary judgment result indicates that further investigation is needed, the system engineer collects the troubleshooting data.
5. Based on the collected data, the system engineer determines whether the system can be restored.

6. If the recovery investigation result indicates that the system can be restored, the system engineer fixes the problem.
7. Regardless of whether the recovery investigation result indicates that the system can be restored, the system engineer isolates the problem based on the troubleshooting data.
8. After the problem is isolated, if the problem can be fixed, the system engineer takes the appropriate actions to fix the problem.
9. If further investigation is needed, the product support service team performs further investigation.
10. Based on the result of the investigation, the team takes the appropriate actions to fix the problem.

## Troubleshooting workflow when the system is being constructed

The following figure shows the troubleshooting workflow for a failure that occurred while the system was being constructed.



A problem occurred when the system was being designed or constructed
1
Collect troubleshooting data
Handled by the system engineer
2
Isolate the problem based on the troubleshooting data
The problem can be fixed
3
Take action
Troubleshooting is complete
Further investigation is needed
4
Investigate further
Handled by the product support service team
5
Take action
Troubleshooting is complete

1. The system engineer collects troubleshooting data for the failure that occurred.
2. Based on the collected data, the system engineer isolates the problem.
3. After the problem is isolated, if the problem can be fixed, the system engineer takes the appropriate actions to fix the problem.
4. If further investigation is needed, the product support service team performs further investigation.
5. After the investigation, the team takes the appropriate actions to fix the problem.

### Related topics

- 9.1 Troubleshooting data output by Application Server

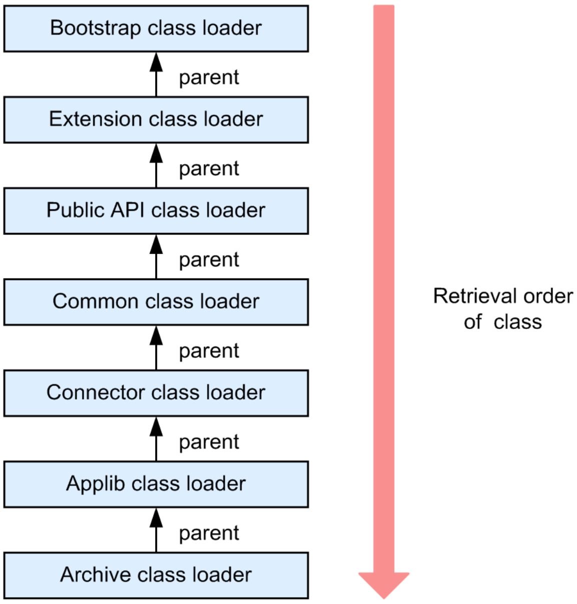
# 3.12 Class loader configuration

A class loader is an object that is responsible for loading classes. Before you develop libraries to be used by applications, you must define the scope of the libraries.

## Class loader configuration

Before developing libraries to be used by applications, application developers must understand the hierarchical structure of the class loaders of Java EE servers, so that they are able to define the scope of the libraries.

The hierarchical structure of the class loaders of Java EE servers is illustrated below.



The following table provides detailed information about the class loaders:

| No | Class loader name | Description |
|---|---|---|
| 1 | `Bootstrap` | Loads the classes provided by the Java VM. |
| 2 | `Extension` | Loads classes from JAR files stored in the `system extensions` directory (*`domain-dir`*`/lib/ext`). |
| 3 | `Public API` | Loads Java EE APIs, Application Server specific APIs, and related implementation classes |
| 4 | `Common` | Loads the library (JAR file) used by applications and the resource adapter.<br>Loads the following classes:<br>1. Classes from the JAR files stored in the *`installation_directory_for_Application_Server`*`/javaee/glassfish/lib` directory<br>2. Classes in the *`installation_directory_for_Application_Server`*`/javaee/glassfish/domains/`*`domain_name`*`/lib` directory. |
| 5 | `Connector` | Loads the resource adapter archives which are shared across all applications. |
| 6 | `Applib` | Loads the library classes (JAR files) specified during deployment.# |
| 7 | `Archive` | Loads the following files or classes, which are included in deployed applications or modules.<br>• WAR, EAR, and JAR files<br>• Application-specific classes generated by the server instance, such as stub classes or servlet generated by JSP pages. |

#:

If two or more deployed applications use the same library, the applications share instances of the same library. One library cannot reference classes of another library.

# 4 Configuring an application execution environment

This chapter describes the procedure for configuring an application execution environment. You can configure a system in which one Java EE server is deployed, or a system in a cluster configuration in which multiple Java EE servers are deployed for load balancing. To configure a system, you can use commands or Administration Console. Note that a cluster configuration is a configuration for managing multiple Java EE servers by grouping.

# 4.1 Application execution environment to be created

The application execution environment can be created in either of two configurations: a configuration with a single Java EE server allocated, or a cluster configuration with multiple Java EE servers allocated to distribute load. This section provides an overview of the application execution environment, such as the configuration of the application execution environment that can be created, and the creation method.

## Configuration of the application execution environment to be created

In this manual, by following the procedure for creating the application execution environment, you can create either of the following configurations:

- Configuration with a single Java EE server allocated

  This is a configuration with a single Java EE server allocated. The configuration consists of the following two machines:

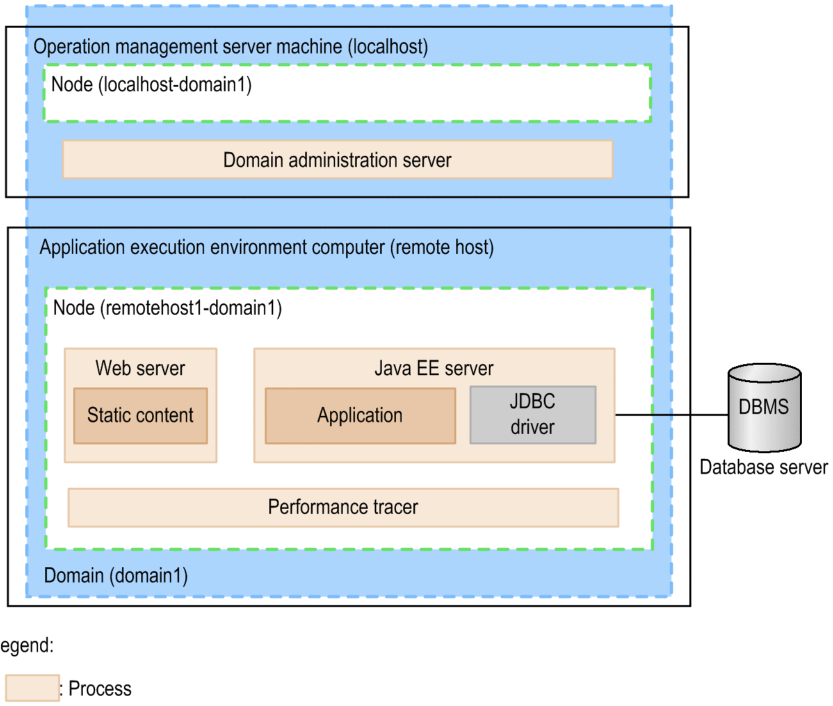
  - Machine used for the application execution environment

    Allocates a single Java EE server, web server, and performance tracer on a single machine that is used for the application execution environment. Processes are managed from the domain and nodes.

  - Machine used for administration

    Allocates a domain administration server.

  The following shows the logical configuration of the application execution environment for a configuration with a single Java EE server allocated.


Operation management server machine (localhost)
Node (localhost-domain1)
Domain administration server
Application execution environment computer (remote host)
Node (remotehost1-domain1)
Web server
Static content
Java EE server
Application
JDBC driver
DBMS
Database server
Performance tracer
Domain (domain1)
Legend:
: Process

- Configuration with multiple Java EE servers allocated

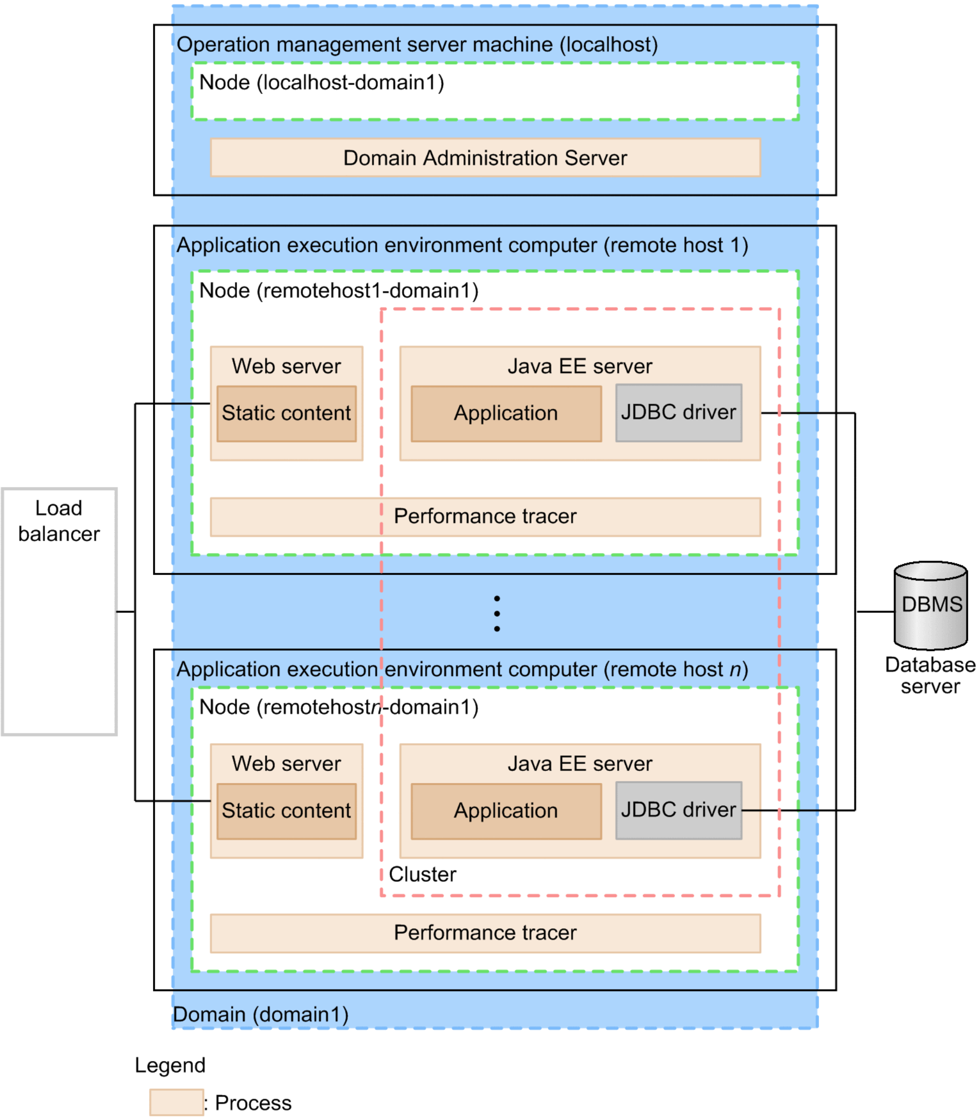
  This is a cluster configuration with multiple Java EE servers allocated. The configuration groups multiple Java EE servers on different machines, and distributes requests to multiple Java EE servers within the cluster. A load balancer is used to distribute requests. The configuration consists of the following machines:

  - Machine used for the application execution environment

    Allocates a Java EE server, web server, and performance tracer on each machine, for multiple application execution environments. Processes are managed from the domain, nodes, and clusters.

- Machine used for administration
  Allocates a domain administration server.

The following shows the logical configuration of the application execution environment for a configuration with multiple Java EE servers allocated.



If configuration procedures or setting values differ depending on the application execution environments to be configured, the phrases "In a configuration where only one Java EE server is deployed" or "In a cluster configuration where more than one Java EE server is deployed" are used to indicate the differences.

## Elements created by default after installing and setting up Application Server

When installation of Application Server is complete, the following components are created (their names are enclosed by square brackets) on the installation destination host.

- Default domain [`domain1`]
- Default node[`localhost-domain1`]
- Thread pool [`admin-thread-pool`, `http-thread-pool`, `thread-pool-1`]

## Methods for creating the application execution environment

An application execution environment can be created by using either of the following methods. This manual gives the procedure for creating an environment by using commands.

- Commands

  Create an environment by performing the procedures described in this manual. Execute commands by using administrator privileges.

- Administration Console

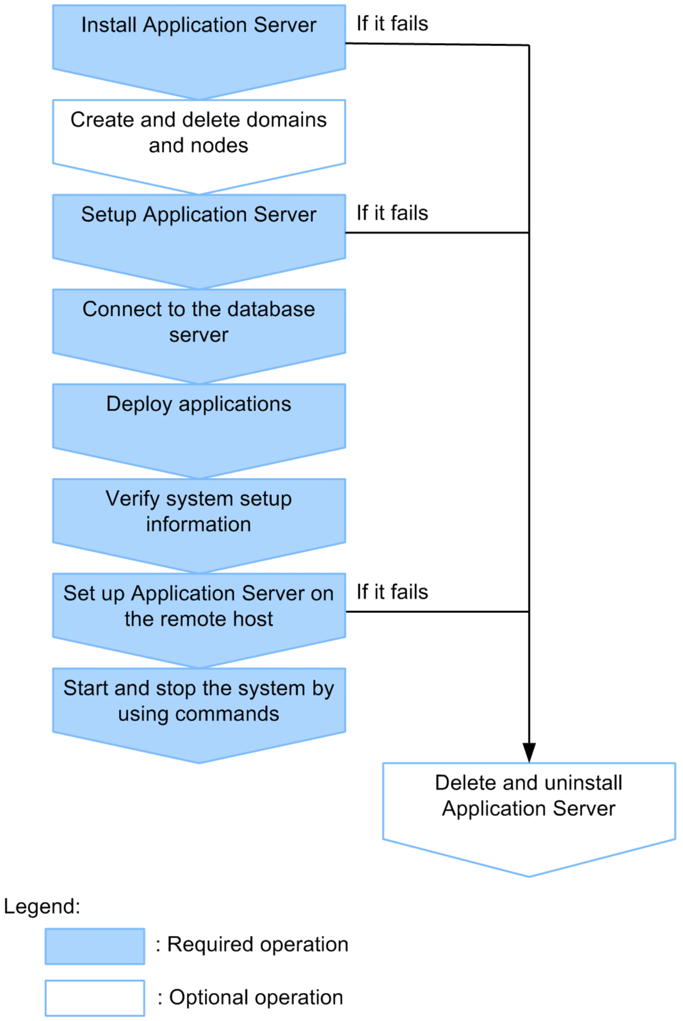
  Log in to the Administration Console and create an environment by using the GUI. Except for operations that are related to the domain, you can use the same procedure as when using commands.

### Related topics

- 7.4.1 Logging into Administration Console

# 4.2 Workflow for configuring an application execution environment

This section describes the processes for configuring an application execution environment, and the workflow for these processes. To configure an application execution environment, you must install Application Server and set up the application execution environment. After this, establish a connection to the database server, and then deploy an application. Additionally, configure Application Server on a remote host and set up it to use the operation management server. After all settings are complete, you must stop the system. If you use commands to configure an application execution environment, execute them with administrator privileges.


Install Application Server
If it fails
Create and delete domains and nodes
Setup Application Server
If it fails
Connect to the database server
Deploy applications
Verify system setup information
Set up Application Server on the remote host
If it fails
Start and stop the system by using commands
Delete and uninstall Application Server
Legend:
: Required operation
: Optional operation

## Related topics

- 4.4.1 About installation types of Application Server
- 4.5.1 Creating domains
- 4.5.2 Creating nodes
- 4.5.3 Deleting nodes
- 4.5.4 Deleting domains
- 4.6.1 Workflow for setting up Application Server
- 4.7.1 Workflow for connecting to a database server
- 4.8.1 Workflow for deploying an application
- 4.9.1 Confirming the system settings information
- 4.10.1 Workflow for configuring an application execution environment on a remote host

- 7.2.1 Starting your system with a command
- 7.2.2 Stopping your system with a command
- 4.11.1 Deleting Application Server
- 4.11.2 Uninstalling Application Server

# 4.3 Overview of disk usage and memory requirements

This section specifies the disk usage and memory requirements of Application Server.

## Disk usage

Linux

| Function name | Disk usage (MB) |
|---|---|
| Application Server - Base | 44.0 |
| Developer's Kit for Java | 179.0 |
| Java EE Server | 140.0 |
| Web Server | 36.0 |

## Memory requirements

Linux

| Function name | Memory requirements (MB) |
|---|---|
| Application Server - Base | 20 |
| Developer's Kit for Java | Included in the requirements for each Java application |
| Java EE Server (for server instances) | 2,649 |
| Java EE Server (for the domain administration server) | 1,000 |
| Web Server | 90 |

## 4.4 Installing Application Server

This section describes the procedure to install Application Server. To install Application Server, start the installer from the provided media of the product on a computer where the operating system (OS) has already been installed.

### 4.4.1 About installation types of Application Server

You can use the following types of installation to install Application Server: new installation, multiple installations, additional installation, and overwrite installation.

The types of installation that you can use to install Application Server are as follows:

New installation

In this type of installation, you install Application Server on a computer in which Application Server has not been installed.

Multiple installations

In this type of installation, you install Application Server into a different directory on a computer in which Application Server has already been installed.

Use this type of installation if you want multiple instances of Application Server to co-exist in the same computer. For example, by having different versions of Application Server on the same computer, you can update an execution environment while leaving the execution environment before the update.

Additional installation

In this type of installation, install Application Server version 10 or later into a different directory on a computer on which one of the following environments has been installed:

- Application execution environment of version 9 or earlier
- Application development environment of version 9 or earlier

Use this type of installation if you want different versions or editions of Application Server to co-exist in the same computer.You can perform an additional installation of Application Server version 10 or later into a machine in which the application execution environment version 9 or earlier has already been configured.

Overwrite installation

In this type of installation, you install the same edition of Application Server on a computer in which Application Server has already been installed. Application Server that was installed earlier is overwritten.

#### Related topics

- 4.1 Application execution environment to be created
- 4.4.2 Installing a new Application Server
- 4.4.3 Installing multiple Application Servers
- 4.4.4 Installing an additional Application Server (where an application execution environment of version 9 or earlier has already been configured)
- 4.4.5 Overwriting an installation of Application Server

## 4.4.2 Installing a new Application Server

To install a new Application Server, start the Program Product Installer from the provided media of the product, and then select the program that you want to install from the program product installation window. After the installation is complete, set the environment settings.

### Prerequisites

- The system engineer has root privileges (super user).
- A prerequisite operating system (OS) and patches are installed.
- Application Server is not installed.

### Intended users

- System engineers

### Procedure

1. Ensure that the packages listed in the following table have been applied.

Table 4-1: Package to be applied for each OS

| Package Name | Red Hat Enterprise Linux 5 | Red Hat Enterprise Linux Server 6 |
|---|---|---|
| compat-libstdc++-296 (i386) | Y | -- |
| compat-libstdc++-33 (i386) | Y | -- |
| compat-libstdc++-33 (i686) | -- | Y |
| compat-libstdc++-33 (x86_64) | Y | Y |
| coreutils (x86_64) | Y | Y |
| findutils (x86_64) | Y | Y |
| gdb (x86_64) | Y | Y |
| glibc (i686) | Y | Y |
| glibc (i686) 2.5-24 or later | Y[#1] | -- |
| glibc (x86_64) | -- | Y |
| glibc (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-common (x86_64) | -- | Y |
| glibc-common (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-devel (i386) 2.5-24 or later | Y[#1] | -- |
| glibc-devel (i686) | -- | Y |
| glibc-devel (x86_64) | -- | Y |
| glibc-devel (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-headers (x86_64) | -- | Y |
| glibc-headers (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-utils (x86_64) | -- | Y |

| Package Name | Red Hat Enterprise Linux 5 | Red Hat Enterprise Linux Server 6 |
|---|---|---|
| glibc-utils (x86_64) 2.5-24 or later | Y[#1] | -- |
| gzip (x86_64) | Y | Y |
| ksh (x86_64) | -- | Y |
| libgcc (i386) | Y | -- |
| libgcc (i686) | -- | Y |
| libstdc++ (i386) | Y | -- |
| libstdc++ (i686) | -- | Y |
| lksctp-tools (x86_64) | Y | Y |
| ncompress (x86_64) | Y | Y[#2] |
| ncurses (x86_64) | Y | Y |
| net-tools (x86_64) | Y | Y |
| nscd (x86_64) | -- | Y |
| nscd (x86_64) 2.5-24 or later | Y[#1] | -- |
| nss-softokn-freebl (i686) | Y | Y |
| procps (x86_64) | Y | Y |
| rpm (x86_64) | Y | Y |
| sysstat (x86_64) | Y | Y |
| tar (x86_64) | Y | Y |

Legend

Y: Apply this package

--: Not applicable

#1

These packages might not be included in the installation media, depending on the version of Red Hat Enterprise Linux 5 that is being used.

#2

If Red Hat Enterprise Linux Server 6.2 is used, ncompress (x86_64) version 4.2.4-54.el6_2.1 or later needs to be applied.

To verify whether a package is installed, execute the `rpm` command. The following example displays both the command execution and execution result of the `rpm` command:

Example of execution:

```
#rpm -q --qf  '%{NAME}-%{ARCH}\n'  ncompress
```

Example of execution result:

```
ncompress-x86_64
```

This example displays that the 64 bit ncompress package was installed.

If the message `Package package_name is not installed` is displayed, this indicates that the package was not installed. In such a case, install the package displayed in *package_name*. During the installation, also install related packages as required.

2. Verify whether the language type of the Program Product Installer matches the language on the terminal where you will perform the installation. If the languages do not match, make them the same.

3. Insert the provided media of the product into the corresponding drive.

4. Run the `mount` command to mount the file system of the corresponding media.

   In Linux, use the following code:

   ```
   mount -r -o mode=0544 /dev/cdrom
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   Note that *special_file_name_of_device* and *mount_directory_name_of_filesystem_for_corresponding_media* vary depending on the operating system (OS), hardware, and environment.

5. Start the setup program from the media of the product.

   In Linux, use the following code:

   ```
   mount_directory_name_of_filesystem_for_corresponding/x64lin/setup -m
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   ### Important note

   The details of the directory name and file name of the corresponding media and how to view this information, might differ depending on the computer environment. Use the `ls` command to verify this information, and enter the displayed file name as is.

   The setup program installs the Program Product Installer and program required to start resident processes automatically, on the hard disk. The Program Product Installer starts automatically.

6. On the main menu of the Program Product Installer, press the **I** key.

7. In the program product installation window, move the cursor over the name of the product to be installed (`Hitachi Application Server`), and then press **Spacebar** key.

   A (`<@>`) icon, which indicates the selection state appears on the left side of the selected program.

8. To select and install the programs that will compose the product, move the cursor to the programs that you do not want to install, and then press the **Spacebar** key.

   The `[@]` icon appears on the left side of programs that are optional. When you press the **Spacebar** key, the icon within `[]` changes, and those programs will not be installed.

   Note that the `@` icon is displayed on the left side of programs that are mandatory.

9. Ensure that the `<@>` icon is displayed on the left side of the product to be installed and that `[@]` or `@` is displayed on the left side of the programs to be installed. Then, press the **I** key.

   A message `Install PP? (y: install, n: cancel)==>` appears at the bottom of the window.

10. Press the **y** or **Y** key.

    A message `Enter the installation path ==>`*`default_installation_destination`* appears at the bottom of the window.

    If you press the **Ctrl**+**n** or **Ctrl**+**N**, then the installation will stop and you will return to the program product installation window.

11. Verify the *default_installation_destination* value and change the installation destination directory as required. Then, press the **Enter** key.

    You can specify only alphanumeric characters, hyphens (-), and underscores (_) in the installation destination.

    If the specified installation directory does not exist, a message confirming the creation of the directory appears.

    ```
    The specified path does not exist. Do you want to create the path?
    Specified path: installation_destination
    ```

    Verify the *installation_destination* and then press the **y** or **Y** key.

    A message `Enter a display name ==>` *`default_display_name`* appears at the bottom of the window.

    To reconfigure the installation destination, press the **Ctrl**+**B** keys.

12. Verify the *default_display_name*, change the display name as required, and then press the **Enter** key.

    A maximum of 40 characters can be specified for the display name. You can only specify alphanumeric characters, hyphens (-), and underscores (_).

    The message asking whether you want to continue with the installation with the specified display name and installation destination is displayed in the window.

    ```
    This program product will be installed at the specified path,
    with the specified display name.
    Do you want to continue?
    Specified display name: display_name
    Specified path        : installation_directory
    ```

    To change the display name or installation destination, press the **n** or **N** key. You will be returned to step 10.

13. Verify the display name and installation destination, and then press the **y** or **Y** key.

    The installation starts.

14. When a message indicating the end of the installation appears, press the **Q** key.

15. On the main menu of the Program Product Installer, press the **L** key.

    The program product list window displays a list of the installed products.

16. Verify whether the product was installed successfully, and then press the **Q** key.

17. On the main menu of the Program Product Installer, press the **Q** key.

    The Program Product Installer closes, and the installation of Application Server is complete.

## Postrequisites

- After the installation is complete, set the environment variables.

### Related topics

- 4.4.6 Configuring the environment after the installation
- 4.1 Application execution environment to be created

## 4.4.3 Installing multiple Application Servers

To install multiple Application Servers in different directories in the same computer, start the Program Product Installer by specifying the `-m` option from the provided media of the product, and then select the program to be installed from the program product installation window.

### Prerequisites

- The system engineer has root privileges (super user).
- A prerequisite operating system (OS) and patches are installed.
- Multiple Application Servers that are the same edition have been installed.

### Intended users

- System engineers

### Procedure

1. Insert the provided media of the product into the corresponding drive.

2. Run the `mount` command to mount the file system of the corresponding media.

   In Linux, use the following code:

   ```
   mount -r -o mode=0544 /dev/cdrom
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   Note that *special_file_name_of_device* and *mount_directory_name_of_filesystem_for_corresponding_media* vary depending on the operating system (OS), hardware, and environment.

3. Start the setup program from the media of the product.

   In Linux, use the following code:

   ```
   mount_directory_name_of_filesystem_for_corresponding/x64lin/setup -m
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   > **Important note**
   >
   > The details of the directory name and file name of the corresponding media and how to view this information, might differ depending on the computer environment. Use the `ls` command to verify this information, and enter the displayed file name as is.

   The setup program installs the Program Product Installer and program required to start resident processes automatically, on the hard disk. The Program Product Installer starts automatically.

4. On the main menu of the Program Product Installer, press the **L** key.
   The program product list window displays a list of the installed products.

5. Verify whether the product was installed successfully, and then press the **Q** key.

6. On the main menu of the Program Product Installer, press the **I** key.

7. In the program product installation window, move the cursor over the name of the product to be installed (`Hitachi Application Server`), and then press **Spacebar** key.

   A (`<@>`) icon, which indicates the selection state appears on the left side of the selected program.

8. To select and install the programs that will compose the product, move the cursor to the programs that you do not want to install, and then press the **Spacebar** key.

   The `[@]` icon appears on the left side of programs that are optional. When you press the **Spacebar** key, the icon within `[]` changes, and those programs will not be installed.

   Note that the `@` icon is displayed on the left side of programs that are mandatory.

9. Ensure that the `<@>` icon is displayed on the left side of the product to be installed and that `[@]` or `@` is displayed on the left side of the programs to be installed. Then, press the **I** key.

   A message `Install PP? (y: install, n: cancel)==>` appears at the bottom of the window.

10. Press the **y** or **Y** key.

    A message `Enter the installation path ==>`*`default_installation_destination`* appears at the bottom of the window.

    If you press the **Ctrl**+**n** or **Ctrl**+**N**, then the installation will stop and you will return to the program product installation window.

11. Verify the *default_installation_destination* value and change the installation destination directory if required, and then press the **Enter** key.

    In *default_installation_destination*, specify a value that is different from the installation directory that you verified in step 5.

    You can specify only alphanumeric characters, hyphens (`-`), and underscores (_) in the installation destination.

    If the specified installation directory does not exist, then a message confirming the creation of the directory appears.

    ```
    The specified path does not exist. Do you want to create the path?
    Specified path: installation_destination
    ```

    Verify the *installation_destination* and then press the **y** or **Y** key.

    A message `Enter a display name ==>` *`default_display_name`* appears at the bottom of the window.

    To reconfigure the installation destination, press the **Ctrl**+**B** keys.

12. Verify the *default_display_name*, change the display name as required, and then press the **Enter** key.

    In *default_display_name*, specify a value that is different from the display name that you verified in step 5.

    A maximum of 40 characters can be specified in the display name.

    A message asking whether you want to continue with the installation with the specified display name and installation destination, is displayed in the window.

    ```
    This program product will be installed at the specified path,
    with the specified display name.
    Do you want to continue?
    Specified display name: display_name
    Specified path        : installation_directory
    ```

    To change the display name or the installation destination, press the **n** or **N** key. You will be returned to step 10.

13. Verify the display name and installation destination, and then press the **y** or **Y** key.

    The installation starts.

14. When a message indicating the end of the installation appears, press the **Q** key.

15. On the main menu of the Program Product Installer, press the **L** key.
    The program product list window displays a list of the installed products.

16. Verify whether the product was installed successfully, and then press the **Q** key.

17. On the main menu of the Program Product Installer, press the **Q** key.
    The Program Product Installer closes, and the installation of Application Server is complete.

## Postrequisites

- After the installation is complete, set the environment settings.

### Related topics

- 4.4.6 Configuring the environment after the installation
- 4.1 Application execution environment to be created

## 4.4.4 Installing an additional Application Server (where an application execution environment of version 9 or earlier has already been configured)

You can install a different edition of Application Server (version 10 or later) in a different directory on the same computer where an application execution environment of version 9 or earlier has already been configured. To install an additional Application Server, start the Program Product Installer by specifying the `-m` option from the provided media of the product, and then select the program that you want to install from the program product installation window.

## Prerequisites

- The system engineer has root privileges (super user).
- A prerequisite operating system (OS) and patches are installed.
- An application execution environment of version 9 or earlier has been configured.

## Intended users

- System engineers

## Procedure

1. Insert the provided media of the product into the corresponding drive.

2. Run the `mount` command to mount the file system of the corresponding media.

   In Linux, use the following code:

   ```
   mount -r -o mode=0544 /dev/cdrom
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   Note that *special_file_name_of_device* and *mount_directory_name_of_filesystem_for_corresponding_media* vary depending on the operating system (OS), hardware, and environment.

3. Start the setup program from the media of the product.

   In Linux, use the following code:

   ```
   mount_directory_name_of_filesystem_for_corresponding/x64lin/setup -m
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   **Important note**

   The details of the directory name and file name of the corresponding media and how to view this information, might differ depending on the computer environment. Use the `ls` command to verify this information, and enter the displayed file name as is.

   The setup program installs the Program Product Installer and program required to start resident processes automatically, on the hard disk. The Program Product Installer starts automatically.

4. On the main menu of the Program Product Installer, press the **I** key.

5. In the program product installation window, move the cursor over the name of the product to be installed (`Hitachi Application Server`), and then press **Spacebar** key.

   A (`<@>`) icon, which indicates the selection state appears on the left side of the selected program.

6. To select and install the programs that will compose the product, move the cursor to the programs that you do not want to install, and then press the **Spacebar** key.

   The `[@]` icon appears on the left side of programs that are optional. When you press the **Spacebar** key, the icon within `[]` changes, and those programs will not be installed.

   Note that the `@` icon is displayed on the left side of programs that are mandatory.

7. Ensure that the `<@>` icon is displayed on the left side of the product to be installed and that `[@]` or `@` is displayed on the left side of the programs to be installed. Then, press the **I** key.

   A message `Install PP? (y: install, n: cancel)==>` appears at the bottom of the window.

8. Press the **y** or **Y** key.

   A message `Enter the installation path ==>`*`default_installation_destination`* appears at the bottom of the window.

   If you press the **Ctrl**+**n** or **Ctrl**+**N**, then the installation will stop and you will return to the program product installation window.

9. Verify the *default_installation_destination* value and change the installation destination directory if required, and then press the **Enter** key.

   In *default_installation_destination*, specify a value that is different from the installation destination of the application execution environment of version 9 or earlier (`/opt/Cosminexus`).

   You can specify only alphanumeric characters, hyphens ( -), and underscores ( _) in the installation destination.

   If the specified installation directory does not exist, then a message confirming the creation of the directory is displayed.

   ```
   The specified path does not exist. Do you want to create the path?
   Specified path: installation_destination
   ```

   Verify the *installation_destination* and then press the **y** or **Y** key.

   A message `Enter a display name ==>` *`default_display_name`* appears at the bottom of the window.

To reconfigure the installation destination, press the **Ctrl**+**B** keys.

10. Verify the *default_display_name*, change the display name as required, and then press the **Enter** key.

    A maximum of 40 characters can be specified in the display name. You can only specify alphanumeric characters, hyphens (-), and underscores (_).

    A message asking whether you want to continue with the installation with the specified display name and installation destination is displayed in the window.

    ```
    This program product will be installed at the specified path,
    with the specified display name.
    Do you want to continue?
    Specified display name: display_name
    Specified path        : installation_directory
    ```

    To change the display name or installation destination, press the **n** or **N** key. You will be returned to step 8.

11. Verify the display name and installation destination, and then press the **y** or **Y** key.

    The installation starts.

12. When a message indicating the end of the installation appears, press the **Q** key.

13. On the main menu of the Program Product Installer, press the **L** key.

    The program product list window displays a list of the installed products.

14. Verify whether the product was installed successfully, and then press the **Q** key.

15. On the main menu of the Program Product Installer, press the **Q** key.

    The Program Product Installer closes, and the installation of Application Server is complete.

### Postrequisites

- After the installation is complete, set the environment settings.

#### Related topics

- 4.4.6 Configuring the environment after the installation
- 4.1 Application execution environment to be created

## 4.4.5 Overwriting an installation of Application Server

To overwrite an installation of Application Server in the same directory of the same computer where Application Server is installed, start the Program Product Installer from the provided media of the product, and then select the program that you want to install from the program product installation window.

### Prerequisites

- The system engineer has root privileges (super user).
- A prerequisite operating system (OS) and patches are installed.
- An Application Server with the same version as the Application Server to be updated is already installed.
- The system that uses the Application Server to be updated has been stopped.

## Intended users

- System engineers

## Procedure

1. If a server or domain is running on the installed Application Server, do the following to stop that server or domain:
   a. To stop the Application Servers, run the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

   b. To stop a domain, run the `stop-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-domain domain_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-domain executed successfully.
   ```

   c. To display the list of domains, run the `list-domains` subcommand of the `asadmin` utility command.

   ```
   asadmin list-domains
   ```

   When this command is run, the result is displayed as follows:

   ```
   Name: domain1 Status: not running
   Name: domain_name Status: not running
   Command list-domains executed successfully.
   ```

   Make sure that the statuses of the stopped domains are `not running`.

2. Insert the provided media of the product into the corresponding drive.

3. Run the `mount` command to mount the file system of the corresponding media.

   In Linux, use the following code:

   ```
   mount -r -o mode=0544 /dev/cdrom
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   Note that *special_file_name_of_device* and *mount_directory_name_of_filesystem_for_corresponding_media* vary depending on the operating system (OS), hardware, and environment.

4. Start the setup program from the media of the product.

   In Linux, use the following code:

   ```
   mount_directory_name_of_filesystem_for_corresponding/x64lin/setup -m
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

### Important note

The details of the directory name and file name of the corresponding media and how to view this information, might differ depending on the computer environment. Use the `ls` command to verify this information, and enter the displayed file name as is.

The setup program installs the Program Product Installer and program required to start resident processes automatically, on the hard disk. The Program Product Installer starts automatically.

5. On the main menu of the Program Product Installer, press the **L** key.

   The program product list window displays a list of the installed products.

6. Confirm the installation destination of the product that will overwrite the program product, and then press the **Q** key.

7. On the main menu of the Program Product Installer, press the **I** key.

8. In the program product installation window, move the cursor over the name of the product to be installed, and then press **Spacebar** key.

   A (`<@>`) icon, which indicates the selection state appears on the left side of the selected program.

9. To select and overwrite the programs that will compose the product and , move the cursor to the programs that you do not want to install, and then press the **Spacebar** key.

   The `[@]` icon appears on the left side of the programs that are optional. When you press the **Spacebar** key, the icon in `[]` changes, and that program will not be overwritten.

   Note that the `@` icon is displayed on the left side of programs that are mandatory.

10. Ensure that the `<@>` icon is displayed on the left side of the product to be overwritten, and that `[@]` or `@` is displayed on the left side of the programs to be overwritten. Then, press the **I** key.

    A message `Install PP? (y: install, n: cancel)==>` appears at the bottom of the window.

11. Press the **y** or **Y** key.

    A message `Enter the installation path ==>`*`installation_destination`* appears at the bottom of the window.

    If you press the **Ctrl**+**n** or **Ctrl**+**N**, then the installation will stop and you will return to the program product installation window.

12. Change the *installation_destination* that you checked in Step 6, and then press the **Enter** key.

    A message confirming whether you want to continue with the installation because the same program product is already installed in the specified *installation_destination* appears.

    ```
    This program product is already installed.
    Do you want to continue with the installation?
    Installation path: installation_destination
    ```

    To change the installation destination, press the **n** or **N** key.

13. Press the **y** or **Y** key.

    The message asking whether you want to continue with the installation with the specified display name and installation destination is displayed in the window.

    ```
    This program product will be installed at the specified path,
    with the specified display name.
    Do you want to continue?
    ```

```
Specified display name: display_name
Specified path        : installation_directory
```

To change the display name or installation destination, press the **n** or **N** key. You will be returned to step 11.

14. Ensure that *configured_display_name* is correct, and then press the **y** or **Y** key.
    The installation starts.

15. When a message indicating the end of the installation appears, press the **Q** key.

16. On the main menu of the Program Product Installer, press the **L** key.
    The program product list window displays a list of the installed products.

17. Verify whether the product was installed successfully, and then press the **Q** key.

18. On the main menu of the Program Product Installer, press the **Q** key.
    The Program Product Installer closes, and the overwrite installation of Application Server is complete.

### Related topics

- 4.1 Application execution environment to be created

## 4.4.6 Configuring the environment after the installation

After installing Application Server, configure the environment to use the system.

### Prerequisites

- Application Server is installed.

### Intended users

- System engineers

### Procedure

1. Specify the TZ environment variable in the setup file (`asenv.conf`)
   The setup file is stored in the `installation_directory_for_Java_EE_Server/glassfish/config/asenv.conf` directory. A definition example of the setup file (`asenv.conf`) is as follows:

```
TZ=JST-9
```

2. Allow the IP address to be resolved from the local host name by specifying an appropriate IP address assigned, for the network interface for the local host name in the `/etc/hosts` file.

## 4.5 Creating and deleting domains and nodes

This section describes the procedure for creating and deleting domains and nodes by using commands. The default domain and node are created after Application Server has been installed.

### 4.5.1 Creating domains

To create a domain, execute the `create-domain` subcommand of the `asadmin` utility command. The default domain is `domain1`.

#### Prerequisites

- Application Server has been installed.

#### Intended users

- System engineers

#### Procedure

1. To create a domain, run the `create-domain` subcommand of the `asadmin` utility command.

```
asadmin create-domain --adminport Admin_port --instanceport HTTP_port
--domainproperties domain.jmxPort=value:http.ssl.port=value:
java.debugger.port=value:jms.port=value:orb.listener.port=value:
orb.mutualauth.port=value:orb.ssl.port=value:
osgi.shell.telnet.port=value domain_name
```

**Important note**

When executing a subcommand of the `asadmin` utility command on a created domain, you must specify *Admin_port* by using the `--port` option of the `asadmin` utility command.

While running a command, specify the user name when the following text prompts you to enter the administrator's user name: `Enter admin user name [Enter to accept default ""admin""/no password]`. Also, specify the password when the following text prompts you to enter the administrator's password: `Enter the admin password [Enter to accept default of no password]`. The default value of the administrator's user name is `admin`. Note that there is no default value for the password.

When this command is run, the result is displayed as follows:

```
Command create-domain executed successfully.
```

2. To start the domain, run the `start-domain` subcommand of the `asadmin` utility command.

```
asadmin start-domain domain_name
```

When this command is run, the result is displayed as follows:

```
Command start-domain executed successfully.
```

3. To display a list of domains, run the `list-domains` subcommand of the `asadmin` utility command.

```
asadmin list-domains
```

Ensure that the statuses of the domains that were started at the step 2 is `running`. When this command is run, the result is displayed as follows:

```
Name: domain1 Status: running
Name: domain_name Status: running
Command list-domains executed successfully.
```

## Postrequisites

- Creating nodes

### Related topics

- 4.5.2 Creating nodes

## 4.5.2 Creating nodes

To create a node, run the `create-node-config` subcommand of the `asadmin` utility command. The default node is `localhost-domain1`.

### Prerequisites

- Installation of Application Server is complete.
- The created domain administration server (DAS) is running.

### Intended users

- System engineers

### Procedure

1. To create a node, execute the `create-node-config` subcommand of the `asadmin` utility command.

```
asadmin create-node-config
--nodehost host_name_to_node
--installdir installation_directory_
for_Application_Server_on_host/javaee_absolute_path
--nodedir directory_path_which_stores_node_information
node_name
```

> **Important note**
>
> To create a node other than the default node, specify the *admin_port* in the `--port` option of the `asadmin` utility command, and run the `create-node-config` subcommand.

When this command is run, the result is displayed as follows:

```
Command create-node-config executed successfully.
```

## 4.5.3 Deleting nodes

To delete a node, execute the `delete-node-config` subcommand of the `asadmin` utility command.

### Prerequisites

- Installation of Application Server is complete.
- The domain administration server (DAS) is running.
- Application Server has been deleted from the host corresponding to the node.

### Intended users

- System engineers

### Procedure

1. To display a list of nodes, run the `list-nodes` subcommand of the `asadmin` utility command.

```
asadmin list-nodes
```

> **Important note**
>
> To display a list of nodes other than the nodes in the default domain, specify *Admin_port* in the `--port` option of the `asadmin` utility command, and then run the `delete-node-config` subcommand.

When this command is run, the result is displayed as follows:

```
node_name  CONFIG  IP_address_or_host_name
node_name  CONFIG  IP_address_or_host_name
The command list-nodes has been executed successfully.
```

2. To delete the node, run the `delete-node-config` subcommand of the `asadmin` utility command.

```
asadmin delete-node-config node_name
```

> **Important note**
>
> - To delete nodes in any domain other than the default domain, specify *Admin_port* in the `--port` option of the `asadmin` utility command and then run the `delete-node-config` subcommand.
> - If another process is using a file or directory below the node directory of the node to be deleted, deletion of the node might fail. Before deleting the node, confirm that other running programs are not using any files or directories below the node directory.

When this command is run, the result is displayed as follows:

```
The command delete-node-config has been executed successfully.
```

## 4.5.4 Deleting domains

To delete a domain, first execute the `stop-domain` subcommand of the `asadmin` utility command to stop the domain, and then execute the `delete-domain` subcommand.

### Prerequisites

- Installation of Application Server is complete.
- The created domain administration server (DAS) is running.

### Intended users

- System engineers

### Procedure

1. To stop the domain, run the `stop-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-domain domain_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-domain executed successfully.
   ```

2. To display the list of domains, run the `list-domains` subcommand of the `asadmin` utility command.

   ```
   asadmin list-domains
   ```

   Ensure that the status of the domain that was stopped in step 1 is `not running`. When this command is run, the result is displayed as follows:

   ```
   Name: domain1 Status: running
   Name: domain_name Status: not running
   Command list-domains executed successfully.
   ```

3. To delete the domain, run the `delete-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin delete-domain domain_name
   ```

   **Important note**

   If another process is using a file or directory below the domain directory of the domain to be deleted, deletion of the domain might fail. Before deleting the domain, confirm that other running programs are not using any files or directories below the domain directory.

   When this command is run, the result is displayed as follows:

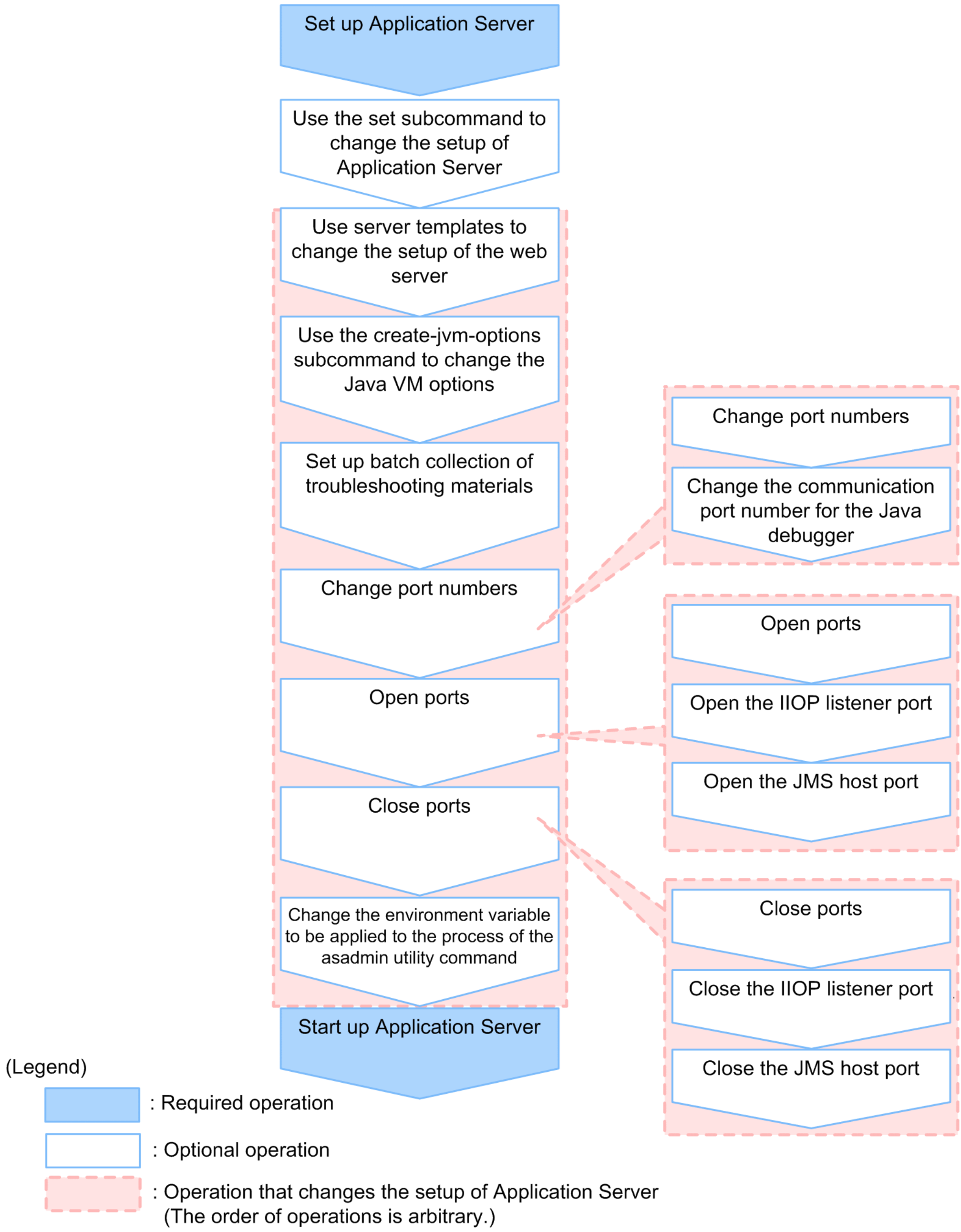
   ```
   Command delete-domain executed successfully.
   ```

# 4.6 Setting up Application Server

This section describes the procedure for setting up Application Server. When Application Server is set up, the performance tracer, Java EE server, and web servers are configured on the host. Additionally, the procedure for starting Application Server by using commands, is also described.

## 4.6.1 Workflow for setting up Application Server

This topic describes the processes for setting up Application Server, and the relevant flow of tasks. While setting up Application Server, configure the performance tracer, Java EE server, and web server on the host, and set up the dependency relations between the servers. After this,change the settings of Application Server, and then start it.


Set up Application Server
Use the set subcommand to change the setup of Application Server
Use server templates to change the setup of the web server
Use the create-jvm-options subcommand to change the Java VM options
Set up batch collection of troubleshooting materials
Change port numbers
Open ports
Close ports
Change the environment variable to be applied to the process of the asadmin utility command
Start up Application Server
Change port numbers
Change the communication port number for the Java debugger
Open ports
Open the IIOP listener port
Open the JMS host port
Close ports
Close the IIOP listener port
Close the JMS host port
(Legend)
: Required operation
: Optional operation
: Operation that changes the setup of Application Server
(The order of operations is arbitrary.)

### Related topics

- 4.6.2 Setting up Application Server
- 4.6.3 Changing Application Server settings by using the set subcommand
- 4.6.4 Changing web server settings
- 4.6.5 Changing Java VM options by using the create-jvm-options subcommand
- 4.6.6 Setting up batch collection of troubleshooting materials
- 4.6.7 Changing a port number
- 4.6.8 Changing the port number of the Java debugger communication port
- 4.6.9 Opening a port
- 4.6.10 Opening an IIOP listener port
- 4.6.11 Opening the port of a JMS host
- 4.6.12 Closing a port
- 4.6.13 Closing an IIOP listener port
- 4.6.14 Closing the port of a JMS host
- 4.6.15 Changing the environment variable to be applied to the process of the asadmin utility command
- 4.6.16 Starting Application Server

## 4.6.2 Setting up Application Server

To set up Application Server, you must set up the performance tracer by using the `create-prf` subcommand of `asadmin` utility command, the Java EE server by using the `create-cluster` and `create-instance` subcommands, and the web server by using the `create-webserver` subcommand. You must also set up the dependency relations by using the `create-relation` subcommand.

### Prerequisites

- Installation of Application Server is complete.

### Intended users

- System engineers

### Procedure

1. To start the DAS, run the `start-domain` subcommand of the `asadmin` utility command.

```
asadmin start-domain
```

When this command is run, the result is displayed as follows:

```
Command start-domain executed successfully.
```

2. To set up the performance tracer, run the `create-prf` subcommand of the `asadmin` utility command.

```
asadmin create-prf --node node_name performance_tracer_name
```

When this command is run, the result is displayed as follows:

```
Command create-prf executed successfully.
```

3. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   Verify whether the name of the performance tracer that was configured in step 2 is displayed in the list.

   ```
   performance_tracer_name not running
   Command list-prfs executed successfully.
   ```

4. For cluster configurations that contain multiple Java EE servers, to set up a cluster that groups the Java EE server, run the `create-cluster` subcommand of the `asadmin` utility command.

   ```
   asadmin create-cluster cluster_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command create-cluster executed successfully.
   ```

5. To set up the Java EE server (server instance), run the `create-instance` subcommand of the `asadmin` utility command.

   ```
   asadmin create-instance --node node_name --prf performance_tracer_name
   --cluster cluster_name server_instance_name
   ```

   - For *performance_tracer_name*, specify the name of the performance tracer that was configured in step 2.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name that was configured in step 4 for the `--cluster` option.
   - By specifying the `--prf` option, a dependency relation between the server instance and the performance tracer is created (the relation type: prf-relation).

   When this command is run, the result is displayed as follows:

   ```
   Command create-instance executed successfully.
   ```

6. To display a list of Java EE servers, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Verify whether the name of the server instance that was configured in step 5 is displayed.

   ```
   server_instance_name host_name port_number process_ID cluster_name not running
   Command list-instances executed successfully.
   ```

   - *cluster_name* is displayed only in a cluster configuration where more than one Java EE server is deployed.

7. To set up the web server, run the `create-webserver` subcommand of the `asadmin` utility command.

   ```
   asadmin create-webserver --node node_name
   --prf performance_tracer_name web_server_name
   ```

   - For *performance_tracer_name*, specify the name of performance tracer that was specified in step 2.

- By specifying the `--prf` option, a dependency relation between the web server and the performance tracer is created (the relation type: prf-relation).

When this command is run, the result is displayed as follows:

```
Command create-webserver executed successfully.
```

8. To display the list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as mentioned below. Verify whether the name of the web server that was specified in step 7 is displayed.

```
web_server_name not running
Command list-webservers executed successfully.
```

9. To associate the Java EE server (server instance) that is used as the redirection destination for the request that is received by the web server, run the `create-relation` subcommand of the `asadmin` utility command.

```
asadmin create-relation --relationtype redirect
--from web_server_name --to server_instance_name
[--properties property_name_of_dependency_relation=value
[:property_name_of_dependency_relation=value]...]
dependency_relation_name
```

- For *web_server_name*, specify the name of the web server that was specified in step 7.
- For *server_instance_name*, specify the server instance that was specified in step 5.
- When setting up the relation for redirection, specify `path` and `network-listener` for the `--properties` option to process static content by the web server, and process requests other than static content by the Java EE server.

  Example:

  `path=/apserver/:network-listener=http-listener-1`

  For `path`, specify the URL path, starting with a slash (`/`). Do not specify a slash only (`path=/`). In this example, a request that does not contain `apserver` as the first part of the file path in the URL following the domain name (for example, `http://xxxxxxxxxxx/index.html`) is accessed as static content of a web server. A request containing `apserver` as the first part of the file path in the URL following the domain name (for example, `http://xxxxxxxxxxx/apserver/sample/index.jsp`) is redirected to the Java EE server. In this case, the request is redirected to the URL `http://yyyyyyyyyyy/sample/index.jsp` on the Java EE server.

  For `network-listener`, specify a network listener name that indicates the HTTP listener or HTTPS listener of the redirection-target Java EE server. On the Java EE server, the following network listeners are defined by default: the HTTP listener `http-listener-1`, and the HTTPS listener `http-listener-2`.

When this command is run, the result is displayed as follows:

```
Command create-relation executed successfully.
```

10. To display a list of dependency relations, run the `list-relations` subcommand of the `asadmin` utility command.

```
asadmin list-relations
```

When this command is run, the result is displayed as follows. Verify whether the following dependency relations are displayed.

- Performance tracer relation between the server instance and performance tracer
- Performance tracer relation between the web server and the performance tracer
- Redirect relation between web server and server instance that was set in step 9.

```
dependency_relation_name prf-relation server_instance_name
performance_tracer_name
dependency_relation_name prf-relation web_server_name
performance_tracer_name
dependency_relation_name redirect web_server_name
server_instance_name
Command list-relations executed successfully.
```

## 4.6.3 Changing Application Server settings by using the set subcommand

To change the Application Server settings, run the `set` subcommand of the `asadmin` utility command to change the setting values of Application Server.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### Procedure

1. To view the settings of Application Server, run the `get` subcommand of the `asadmin` utility command.

```
asadmin get "*"
```

In the execution results of the command, verify the identifier of the setting to be changed and the value before the change.
If a value is specified for multiple parameters, the specified value is treated as follows:

For performance tracer-related parameters:

When parameter values that start with both `hitachi-prf.hitachi-prf.`*`performance_tracer_name`*`.` and `hitachi-prf-configs.hitachi-prf-config.`*`configuration_name_of_the_performance_tracer`* are set, the value of the parameter that starts with `hitachi-prf.hitachi-prf.`*`performance_tracer_name`*`.` takes effect.

For web server-related parameters

When parameter values that start with both `hitachi-webservers.hitachi-webserver.`*`web_server_name`* and `hitachi-webserver-configs.hitachi-webserver-config.`*`configuration_name_of_the_web_server`* are set, the value of the parameter that starts with `hitachi-webservers.hitachi-webserver.`*`web_server_name`* takes effect.

For a server instance-related parameter

When parameter values that start with both `servers.server.`*`Java_EE_server_name`* and `configs.config.`*`configuration_name_of_the_Java_EE_server`* are set, the value of the parameter that starts with `servers.server.`*`Java_EE_server_name`* takes effect.

2. Specify the value of the identifier to be changed, and then run the `set` subcommand of the `asadmin` utility command.

```
asadmin set target_identifier_to_be_changed=value
```

> **Reference note**
>
> When you change values other than standard properties (directives) of the web server, use the server template to change the web server settings.

When this command is run, the result is displayed as follows:

```
Command set executed successfully.
```

3. To verify the settings of Application Server after the change, run the `get` subcommand of the `asadmin` utility command.

```
asadmin get "*"
```

Verify that the value that was specified for the identifier by running the `set` subcommand in step 2 is reflected in the execution result of the command.

### Related topics

- 4.6.4 Changing web server settings

## 4.6.4 Changing web server settings

To change the web server settings (other than standard properties), use the server template that describes settings required to run the web server. When changing the web server settings, either set extended properties or enter a directive directly into the server template. When editing a server template, we recommend setting the extended properties.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### Storage location and file name of the server template

The server template is deployed after starting the domain for the first time.

The file name of the server template is as follows:

- `httpsd.conf@linux.vtl` (in Linux)

  The server template for the web server basic settings. This template contains basic settings other than those for request transmission to the web server and load balancing.

- `reverse_proxy.conf@.vtl`

  The server template for the request transmission settings of the web server. This template contains the settings for when the server instance of the request transmission destination is not in a cluster configuration.

- `proxy_balancer.conf@.vtl`

  The server template for the load balancing settings of the web server. This template contains the settings for when the server instance of the request transmission destination is in a cluster configuration.

When the domain is started for the first time, files in the server template are deployed to the following location: *`installation_directory_for_Application_Server`*`/javaee/glassfish/domains/`*`domain_name`*`/server_templates/webserver/conf`.

## Editing the server template

The server template can be edited as follows:

- Use VTL syntax to set the extension properties.

  Specify the VTL syntax in the server template, and then set up the web server by processing the extended property values of the `set` subcommand of the `asadmin` utility command.

- Directly specify directives.

  Set up the web server by entering directives directly into the server template.

By using VTL syntax to specify extended properties, you will be able to change the web server settings by using the `set` subcommand of the `asadmin` utility command, and verify the changes by using the `get` subcommand.

When the `create-webserver` and `start-webserver` subcommands of the `asadmin` utility command is run, the server template is read by the DAS. The settings are then applied to the definition file that is read by the web server.

## Procedure

1. To set extended properties by using VTL syntax, do the following:

   The following uses an example of setting the `ProxyPreserveHost` directive to `reverse_proxy.conf` to explain the procedure.

   a. Using a text editor, open the server template file (`reverse_proxy.conf@.vtl`), and then specify the VTL syntax of extended properties.

   To specify extended properties, add `ex_` at the beginning of the directive name.

   For example, for the `ProxyPreserveHost` directive, extended properties can be set by using `ex_ProxyPreserveHost.value`.

   ```
   ProxyPreserveHost ${property.ex_ProxyPreserveHost.value}
   ```

   b. Save the server template file.

   c. To specify `On` for the value of the `ex_ProxyPreserveHost.value` extended property of the web server settings, run the `set` subcommand of the `asadmin` utility command.

   ```
   asadmin set hitachi-webservers.hitachi-
   webserver.web_server_name.property.ex_ProxyPreserveHost=On
   ```

When this command is run, the result is displayed as follows:

```
hitachi-webservers.hitachi-
webserver.web_server_name.property.ex_ProxyPreserveHost=On
Command set executed successfully.
```

d. Run the `get` subcommand of the `asadmin` utility command to verify whether the value of the `ex_ProxyPreserveHost.value` extended property is the same as intended.

```
asadmin get hitachi-webservers.hitachi-
webserver.web_server_name.property.ex_ProxyPreserveHost
```

When this command is run, the result is displayed as follows:

```
hitachi-webservers.hitachi-
webserver.web_server_name.property.ex_ProxyPreserveHost=On
Command get executed successfully.
```

2. When specifying a directive directly, do the following:

   a. Using a text editor, open the server template file, and then specify the directive of the Web Server directly.

   To specify the directive as a comment, start the line with a hash mark (#) followed by a half-width space.

   b. Save the server template file.

Initial values for the web server settings

When a web server is configured by using individual server template files, the initial values for settings in each definition file of the web server as read by the DAS are as follows:

`httpsd.conf`

```
Listen 80

StartServers 20
MinSpareServers 10
MaxSpareServers 20
MaxRequestWorkers 150
MaxConnectionsPerChild 10000
Timeout 30
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 3
HostnameLookups Off

User bin
Group bin


ServerRoot "installation_directory_for_Application_Server/httpsd"

ServerName www.example.com
DocumentRoot "installation_directory_for_Application_Server/javaee/
glassfish/nodes/localhost-domain1/Web1/root/htdocs"
DirectoryIndex index.html
UseCanonicalName Off
ServerSignature Off
ServerTokens ProductOnly
TraceEnable Off

LogLevel warn
ErrorLog "|installation_directory_for_Application_Server/httpsd/sbin/
rotatelogs Installation directory of Application Server/javaee/logs/nodes/
localhost-domain1/Web1/error 86400 -fnum 8 -diff 540"
```

```
HWSRequestLog "|installation_directory_for_Application_Server/httpsd/
sbin/rotatelogs Installation directory of Application Server /javaee/logs/
nodes/localhost-domain1/Web1/hwsrequest 86400 -fnum 8 -diff 540"
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O"
combinedio
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
LogFormat "%h %l %u %t \"%r\" %>s %b %P %{hws_ap_root}n %I %O %X %D
\"%{Referer}i\" \"%{User-Agent}i\"" hws_trace
LogFormat "%h %l %u %t \"%r\" %>s %b %T %P %{hws_ap_root}n" hws_std
HWSLogTimeVerbose On
CustomLog "|installation_directory_for_Application_Server/httpsd/sbin/
rotatelogs  Installation directory of Application Server /javaee/logs/
nodes/localhost-domain1/Web1/access 86400 -fnum 8 -diff 540" hws_std
PidFile "installation_directory_for_Application_Server/javaee/logs/nodes/
localhost-domain1/Web1/httpd.pid"
HWSTraceIdFile "installation_directory_for_Application_Server/javaee/
logs/nodes/localhost-domain1/Web1/hws.trcid"
HWSTraceLogFile "installation_directory_for_Application_Server/javaee/
logs/nodes/localhost-domain1/Web1/hws.trclog"

SSLDisable

TypesConfig "installation_directory_for_Application_Server/httpsd/conf/
mime.types"
AddEncoding x-compress .Z
AddEncoding x-gzip .gz .tgz
AddLanguage ca .ca
AddLanguage cs .cz .cs
AddLanguage da .dk
AddLanguage de .de
AddLanguage el .el
AddLanguage en .en
AddLanguage eo .eo
AddLanguage es .es
AddLanguage et .et
AddLanguage fr .fr
AddLanguage he .he
AddLanguage hr .hr
AddLanguage it .it
AddLanguage ja .ja
AddLanguage ko .ko
AddLanguage ltz .ltz
AddLanguage nl .nl
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pl .po
AddLanguage pt .pt
AddLanguage pt-BR .pt-br
AddLanguage ru .ru
AddLanguage sv .sv
AddLanguage tr .tr
AddLanguage zh-CN .zh-cn
AddLanguage zh-TW .zh-tw

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
```

```
redirect-carefully
BrowserMatch "MS FrontPage" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[01234]" redirect-carefully
BrowserMatch "^gnome-vfs/1.0" redirect-carefully
BrowserMatch "^XML Spy" redirect-carefully
BrowserMatch "^Dreamweaver-WebDAV-SCM1" redirect-carefully
BrowserMatch " Konqueror/4" redirect-carefully

Alias /icons/ "installation_directory_for_Application_Server/httpsd/icons/"
IndexOptions FancyIndexing
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
DefaultIcon /icons/unknown.gif
ReadmeName README.html
HeaderName HEADER.html
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

<Directory />
        Options None
        AllowOverride None
</Directory>

<Directory "installation_directory_for_Application_Server/httpsd/htdocs">
        Options None
        AllowOverride None
</Directory>

<FilesMatch "^\.(ht|key)">
        Order allow,deny
        Deny from all
</FilesMatch>

Include "installation_directory_for_Application_Server/javaee/glassfish/
nodes/localhost-domain1/Web1/root/conf/reverse_proxy.conf"

HWSGracefulStopLog On
HWSGracefulStopTimeout 0

HWSPrfId PRF1
```

reverse_proxy.conf

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
HWSSuppressModuleTrace mod_proxy.c hook
ProxyVia Off
ProxyTimeout 200

ProxyPass / http://localhost:28080/  connectiontimeout=2
ProxyPassReverse / http://localhost:28080/
```

proxy_balancer.conf

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
HWSSuppressModuleTrace mod_proxy.c hook
HWSSuppressModuleTrace mod_proxy_balancer.c
ProxyVia Off
ProxyTimeout 200
```

## 4.6.5 Changing Java VM options by using the create-jvm-options subcommand

Java VM options can be specified for a server instance and for the domain administration server (DAS). To change a Java VM option, which has already been specified, use the `delete-jvm-options` subcommand of the `asadmin` utility command to delete the existing Java VM option, and then use the `create-jvm-options` subcommand to specify a new Java VM option.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### Procedure

1. To display a list of Java VM options for all server instances in the cluster, run the `list-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin list-jvm-options --target server_instance_name_cluster_name
```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

2. When you change a Java VM option from the options listed in step 1, to delete the earlier option, run the `delete-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin delete-jvm-options --target server_instance_name_or_cluster_name
[option_name[=value][:option_name[=value]]...]
```

- In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.

  In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.
- When you specify multiple options, separate them with a colon (`:`).

#### Important note

If an option name or value contains symbols, the backslash (`\`) must be used as an escape character for the symbols in the name or value.

For example, to specify `-XX:MaxMetaspaceSize=192m`, insert an escape character before the colon (`:`) by using double backslashes (`\\`), and specify the text as `-XX\\:MaxMetaspaceSize=192m`.

When this command is run, the result is displayed as follows:

```
Deleted n option(s)
Command delete-jvm-options executed successfully.
```

*n* indicates the number of options that are specified.

3. To specify a value for the Java memory, such as the Java heap for all the server instances, run the `create-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin create-jvm-options --target server_instance_name_cluster_name
[option_name[=value][:option_name[=value]]...]
```

- In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.

  In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.
- Some default values of the Java VM options differ from the values that are set for Java EE Server. The following table displays the difference between these values.

### Table 4-2: Options for which the default Java VM values differ from the values set for the server instance

| Option name | Values set for Java EE Server |
|---|---|
| -XX:HitachiExplicitHeapMaxSize | -XX:HitachiExplicitHeapMaxSize=512m |
| -XX:SurvivorRatio | For Linux<br>-XX:SurvivorRatio=8 |
| -XX:[+\|-]HitachiUseExplicitMemory | -XX:+HitachiUseExplicitMemory |
| -XX:MaxMetaspaceSize | -XX:MaxMetaspaceSize=256m |
| -XX:MetaspaceSize | -XX:MetaspaceSize=256m |

| Option name | Values set for Java EE Server |
|---|---|
| -Xms | -Xms1536m |
| -Xmx | -Xmx1536m |

### Important note

- When you change the value of an option that has already been specified, use the `delete-jvm-options` subcommand to delete the earlier option. If the specified option is not deleted, then a warning is displayed informing you that the same option is registered multiple times. When this warning is displayed, verify the specified option using the `list-jvm-options` subcommand and then delete the options that are not required.
- The DAS runs SystemGC every hour and a server instance runs SystemGC every 24 hours. Use the `sun.rmi.dgc.server.gcInterval` and `sun.rmi.dgc.client.gcInterval` system properties settings to change the execution-interval of the SystemGC process. If the occurrence of the GC processes do not reduce even when you extend the occurrence interval of FullGC by changing the values of these system properties, then the Java heap might be insufficient. In this case, you might be able to improve the occurrence interval of FullGC by tuning the Java heap.

When this command is run, the result is displayed as follows:

```
Created n option(s)
Command create-jvm-options executed successfully.
```

*n* indicates the number of options that are specified.

4. To display a list of Java VM options for all server instances, run the `list-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin list-jvm-options --target server_instance_name_or_cluster_name
```

- In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
- In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

Verify and confirm that the value of the option specified in step 3 has changed.

5. To display a list of Java VM options for the domain administration server, run the `list-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin list-jvm-options
```

6. When you change a Java VM option from the options listed in step 5, to delete the earlier option, run the `delete-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin delete-jvm-options [option_name[=value]
[:option_name[=value]]...]
```

When this command is run, the result is displayed as follows:

```
Deleted n option(s)
Command delete-jvm-options executed successfully.
```

*n* indicates the number of options that are specified.

7. To specify a value for the Java memory, such as the Java heap for the DAS, run the `create-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin create-jvm-options [option_name[=value]
[:option_name[=value]]...]
```

You can specify the value for `option_name[= value]`, such as `-Xms1024m -Xmx1024m`.

Some of the default values of the Java VM options, which differ from the values set for Java EE Server. The following table displays the difference between these values.

Table 4-3: Options for which the default Java VM values differ from the values set for the DAS

| Option name | Values set for Java EE Server |
|---|---|
| -XX:MaxMetaspaceSize | -XX:MaxMetaspaceSize=192m |
| -XX:MetaspaceSize | -XX:MetaspaceSize=192m |
| -Xms | -Xms512m |
| -Xmx | -Xmx512m |

**Important note**

Adjust the Java heap size of the DAS, based on the archive file size of the application that is deployed. Depending on the archive file size, the Java heap size of the DAS may be full and this may lead to insufficient memory.

Additionally, when an inappropriate value (an extremely small or large value) is specified for the Java heap size of the DAS, then the DAS might not start and you might have to reconfigure the domain.

To prevent a case where memory might be insufficient, it is recommended that you run the `backup-domain` command in advance to back up the domain.

When this command is run, the result is displayed as follows:

```
Created n option(s)
Command create-jvm-options executed successfully.
```

*n* indicates the number of options that are specified.

8. To list the options of the Java VM options for DAS, run the `list-jvm-options` subcommand of the `asadmin` utility command.

```
asadmin list-jvm-options
```

Verify that the value of the option specified in step 7 has changed.

9. To specify the options (except Java memory-related options) for the server instance and DAS, repeat steps 1 to 8.

Some of the default values of the Java VM options, which differ from the values set for Java EE Server. The following table displays the difference between these values.

Table 4-4: Options for which the default Java VM values differ from the set values

| Classification | Option name | Values set for Java EE Server |
|---|---|---|
| For the server instance | -XX:HitachiExplicitMemoryJavaLog | -XX:HitachiExplicitMemoryJavaLog:*installation_directory_for_Java_EE Server*/logs/nodes/*node_name*/*server_instance_name*/je_eheap_event |
| | -XX:HitachiExplicitMemoryLogLevel | -XX:HitachiExplicitMemoryLogLevel:normal |
| | -XX:HitachiJavaLog | -XX:HitachiJavaLog:*installation_directory_for_Java_EE Server*/logs/nodes/*node_name*/*server_instance_name*/je_javavm |
| | -XX:[+\|-]HitachiOutOfMemoryCause | -XX:+HitachiOutOfMemoryCause |
| | -XX:[+\|-]HitachiOutOfMemorySize | -XX:+HitachiOutOfMemorySize |
| | -XX:[+\|-]HitachiTrueTypeInLocals | -XX:+HitachiTrueTypeInLocals |
| | -Xhras | -Xhras |
| For the DAS | -XX:HitachiJavaLog | -XX:HitachiJavaLog:*installation_directory_for_Java_EE Server*/logs/domains/*domain_name*/das_javavm |
| | -XX:[+\|-]HitachiOutOfMemoryCause | -XX:+HitachiOutOfMemoryCause |
| | -XX:[+\|-]HitachiOutOfMemorySize | -XX:+HitachiOutOfMemorySize |
| | -XX:[+\|-]HitachiTrueTypeInLocals | -XX:+HitachiTrueTypeInLocals |
| | -Xhras | -Xhras |

## 4.6.6 Setting up batch collection of troubleshooting materials

By using the system information collection functionality, you can collect troubleshooting materials in a batch. To use this functionality to collect troubleshooting materials in a batch, edit the script file of the command for batch collection of troubleshooting materials. If a failure occurs, this command is executed automatically and collects troubleshooting materials. Note that if you have not changed the administrative user and password from the domain default, you do not need to edit them.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### Procedure

1. Edit the script file of the command for batch collection of troubleshooting materials.
   If you have not changed the administrative user and password from the domain default, you do not need to edit them.

Script file to be edited

*installation_directory_for_Java_EE_Server*/glassfish/config/manager/snapshot_event-hook

Settings to be edited

Open the script file in a text editor, and edit the following setting values:

```
ADMIN_HOST=host_name_of_domain_administration_server
USER_ID=user_name_of_domain_administration_server
PWDFILE=path_of_password_file#
```

#: Specify the path of the password file that contains `AS_ADMIN_PASSWORD=`*domain_administration_server_password*. If no password is set, specify an empty value for `PWDFILE`.

Example: If no password is set:

```
PWDFILE=
```

### Related topics

- 9.1 Troubleshooting data output by Application Server

## 4.6.7 Changing a port number

To change a port number, set the new port number for a port-related parameter by using the `set` subcommand of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### The port-related parameters to be set when changing a port number

When you change a port number, the following port-related parameters are to be specified in the `get` subcommand and the `set` subcommand of the `asadmin` utility command:

- `configs.config.`*configuration_name_of_the_Java_EE_server*`.iiop-service.iiop-listener.`*ID*`.port`
- `configs.config.`*configuration_name_of_the_Java_EE_server*`.network-config.network-listeners.network-listener.`*listener_name*`.port`
- `configs.config.`*configuration_name_of_the_Java_EE_server*`.admin-service.jmx-connector.system.port`
- `configs.config.`*configuration_name_of_the_Java_EE_server*`.jms-service.jms-host.`*JMS_host_name*`.port`
- `hitachi-webservers.hitachi-webserver.`*web_server_name*`.property.listen-port`

- `hitachi-webservers.hitachi-webserver.`*`web_server_name`*`.property.listen-add-port`*`n`*
- `hitachi-webservers.hitachi-webserver.`*`web_server_name`*`.property.server-name`
- `hitachi-webserver-configs.hitachi-webserver-config.`*`configuration_name_of_the_web_server`*`.property.listen-port`
- `hitachi-webserver-configs.hitachi-webserver-config.`*`configuration_name_of_the_web_server`*`.property.listen-add-port`*`n`*
- `hitachi-webserver-configs.hitachi-webserver-config.`*`configuration_name_of_the_web_server`*`.property.server-name`

### Procedure

1. To acquire the setting value of a port-related parameter and verify the port number before the change, run the `get` subcommand of the `asadmin` utility command.

   ```
   asadmin get port_number_related_parameter
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command get executed successfully.
   ```

2. To set a changed port number, run the `set` subcommand of the `asadmin` utility command.

   ```
   asadmin set port_number_related_parameter=changed_port_number
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command set executed successfully.
   ```

## 4.6.8 Changing the port number of the Java debugger communication port

To change the port number of the Java debugger communication port, use the `delete-system-property` subcommand of the `asadmin` utility command to delete the port number before the change, and then use the `create-system-properties` subcommand to specify a new port number.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

## Procedure

1. To acquire the setting value of the system property `JAVA_DEBUGGER_PORT` of the Java debugger communication port and verify the port number before the change, run the `list-system-properties` subcommand of the `asadmin` utility command.

   ```
   asadmin list-system-properties server_instance_name_or_cluster_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name.

   When this command is run, the result is displayed as follows:

   ```
   JAVA_DEBUGGER_PORT=29010
       :
   Command list-system-properties executed successfully.
   ```

   ### Reference note

   If the system property (`JAVA_DEBUGGER_PORT`) is set, perform steps 2 and 3. If this property is not set, perform step 3.

2. To delete the system property `JAVA_DEBUGGER_PORT`, run the `delete-system-property` subcommand of the `asadmin` utility command.

   ```
   asadmin delete-system-property --target server_instance_name_or_cluster_name
   JAVA_DEBUGGER_PORT
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   When this command is run, the result is displayed as follows:

   ```
   Command delete-system-property executed successfully.
   ```

3. To set a new port number for the system property `JAVA_DEBUGGER_PORT`, run the `create-system-properties` subcommand of the `asadmin` utility command.

   ```
   asadmin create-system-properties --target server_instance_name_or_cluster_name
   JAVA_DEBUGGER_PORT=changed_port_number
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   When this command is run, the result is displayed as follows:

   ```
   Command create-system-properties executed successfully.
   ```

## 4.6.9 Opening a port

To open a closed port, run the `set` subcommand of the `asadmin` utility command to enable the setting of the port-related parameter.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.

### Intended users

- System engineers

### Port-related parameter to be specified for opening a port

When opening a port, specify the following port-related parameter for the `get` and `set` subcommands of the `asadmin` utility command:

- `configs.config.`*`configuration_name_of_the_Java_EE_server`*`.network-config.network-listeners.network-listener.`*`listener_name`*`.enabled`

  This parameter opens the port with the port number specified by `configs.config.`*`configuration_name_of_the_Java_EE_server`*`.network-config.network-listeners.network-listener.`*`listener_name`*`.port`.

### Procedure

1. Run the `get` subcommand of the `asadmin` utility command to get the setting value for the port-related parameter, and confirm that the port is closed (the setting value is `false`).

   ```
   asadmin get port_related_parameter_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command get executed successfully.
   ```

2. To open the closed port, run the `set` subcommand of the `asadmin` utility command.

   ```
   asadmin set port_related_parameter_name=true
   ```

   Specifying `true` for the value of the port-related parameter opens the port by enabling the setting of this parameter.

   When this command is run, the result is displayed as follows:

   ```
   Command set executed successfully.
   ```

## 4.6.10 Opening an IIOP listener port

To open an IIOP listener port, run the `create-iiop-listener` subcommand of the `asadmin` utility command to create an IIOP listener.

## Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The available port number has already been confirmed.

## Intended users

- System engineers

## Procedure

1. Run the `create-iiop-listener` subcommand of the `asadmin` utility command to create an IIOP listener and open the port.

```
asadmin create-iiop-listener --listeneraddress 0.0.0.0 --iiopport port_number
--target server_instance_name_or_cluster_name IIOP_listener_name
```

   - In a configuration where only one Java EE server is deployed, the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   When this command is run, the result is displayed as follows:

```
Command create-iiop-listener executed successfully.
```

# 4.6.11 Opening the port of a JMS host

To open the port of a JMS host, run the `create-jms-host` subcommand of the `asadmin` utility command to create a JMS host.

## Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The available port number has already been confirmed.

## Intended users

- System engineers

## Procedure

1. Run the `create-jms-host` subcommand of the `asadmin` utility command to create a JMS host and open the port.

```
asadmin create-jms-host --mqhost host_name_of_JMS_service
--mqport port_number_of_JMS_service --mquser user_name_of_JMS_service
--mqpassword password_of_JMS_service
--target server_instance_name_or_cluster_name JMS_host_name
```

- In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
- In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

When this command is run, the result is displayed as follows:

```
Command create-jms-host executed successfully.
```

## 4.6.12 Closing a port

To close an open port, run the `set` subcommand of the `asadmin` utility command to disable the setting of the port-related parameter.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.

### Intended users

- System engineers

### Port-related parameter to be specified for closing a port

When closing the port, specify the following port-related parameter for the `get` and `set` subcommands of the `asadmin` utility command:

- configs.config.*configuration_name_of_the_Java_EE_server*.network-config.network-listeners.network-listener.*listener_name*.enabled

  This parameter closes the port that has the port number specified by configs.config.*configuration_name_of_the_Java_EE_server*.network-config.network-listeners.network-listener.*listener_name*.port.

### Procedure

1. Run the `get` subcommand of the `asadmin` utility command to get the setting value for the port-related parameter, and confirm that the port is open (the setting value is `true`).

   ```
   asadmin get port_related_parameter
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command get executed successfully.
   ```

2. To close the opened port, run the `set` subcommand of the `asadmin` utility command.

   ```
   asadmin set port_related_parameter=false
   ```

   Specifying `false` for the port-related parameter closes the port by disabling the setting of this parameter.

   When this command is run, the result is displayed as follows:

```
Command set executed successfully.
```

## 4.6.13 Closing an IIOP listener port

To close an IIOP listener port, run the `delete-iiop-listener` subcommand of the `asadmin` utility command to delete the IIOP listener.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.

### Intended users

- System engineers

### Procedure

1. Run the `list-iiop-listeners` subcommand of the `asadmin` utility command to confirm the IIOP listener whose port you want to close.

   ```
   asadmin list-iiop-listeners server_instance_name_or_cluster_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for *server_instance_name_or_cluster_name*.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for *server_instance_name_or_cluster_name*.

   When this command is run, the result is displayed as follows:

   ```
   Command list-iiop-listeners executed successfully.
   ```

2. Run the `delete-iiop-listener` subcommand of the `asadmin` utility command to delete the IIOP listener and close the port.

   ```
   asadmin delete-iiop-listener
   --target server_instance_name_or_cluster_name IIOP_listener_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   When this command is run, the result is displayed as follows:

   ```
   Command delete-iiop-listener executed successfully.
   ```

## 4.6.14 Closing the port of a JMS host

To close the port of a JMS host, run the `delete-jms-host` subcommand of the `asadmin` utility command to delete the JMS host.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.

### Intended users

- System engineers

### Procedure

1. Run the `list-jms-hosts` subcommand of the `asadmin` utility command to confirm the JMS host whose port you want to close.

   ```
   asadmin list-jms-hosts --target server_instance_name_or_cluster_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for *server_instance_name_or_cluster_name*.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for *server_instance_name_or_cluster_name*.

   When this command is run, the result is displayed as follows:

   ```
   Command list-jms-hosts executed successfully.
   ```

2. Run the `delete-jms-host` subcommand of the `asadmin` utility command to delete the JMS host and close the port.

   ```
   asadmin delete-jms-host
   --target server_instance_name_or_cluster_name JMS_host_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   When this command is run, the result is displayed as follows:

   ```
   Command delete-jms-host executed successfully.
   ```

## 4.6.15 Changing the environment variable to be applied to the process of the asadmin utility command

To change the environment variable to be applied to the process of the `asadmin` utility command, edit the environment definition file of the Java EE server (`asenv.conf`). For the environment variable to be applied to the process of the `asadmin` utility command, specify values for Java memory, such as the Java heap; the log for the `asadmin` utility command; and other information. For example, if many applications are deployed or many files are included in applications, and a memory shortage thereby occurs when the Java EE server starts, change the size of the Java heap to be applied to the process of the `asadmin` utility command.

## Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

## Intended users

- System engineers

## Procedure

1. Edit the environment variable definition file of Java EE Server (`asenv.conf`), and change the environment variable to be applied to the process of the `asadmin` utility command.

   Change the environment variable for the remote host and the localhost in a cluster configuration that contains more than one Java EE server.

   Environment variable definition file of Java EE Server

   > *installation_directory_for_Application_Server*/javaee/glassfish/config/asenv.conf

   Example edits

   > If you want to change the maximum size of the Java heap, set a value for the environment variable HJES_ASADMIN_JVM_OPTIONS.

   ```
   HJES_ASADMIN_JVM_OPTIONS=-Xmx256m
   ```

# 4.6.16 Starting Application Server

To start Application Server, run the `start-servers` subcommand of the `asadmin` utility command.

## Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

## Intended users

- System engineers

## Procedure

1. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin start-servers
   ```

   When this command is run, the result is displayed as follows:

```
Command start-servers executed successfully.
```

2. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

   ```
   performance_tracer_name running
   Command list-prfs executed successfully.
   ```

3. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

   ```
   server_instance_name host_name port_number process_ID cluster_name running
   Command list-instances executed successfully.
   ```

   - *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

4. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

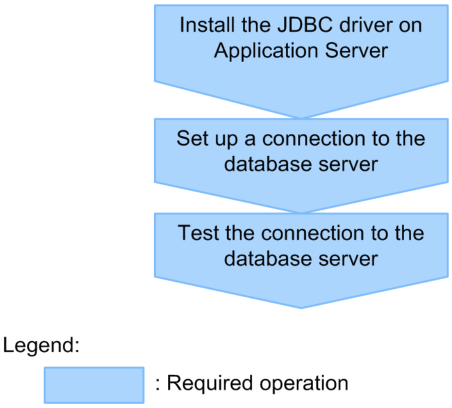
   ```
   web_server_name   running
   Command list-webservers executed successfully.
   ```

# 4.7 Connecting to a database server

This section describes the procedure for establishing a connection between Application Server and a database server. To connect to the database server, you must install a JDBC driver and set it up on Application Server.

## 4.7.1 Workflow for connecting to a database server

This topic describes the process required to establish a connection between Application Server and a database server, and the relevant flow of tasks. You must install and configure a JDBC driver to establish a connection between Application Server and the database server. To verify whether a connection can be established between Application Server and the database server, perform a connection test.


Install the JDBC driver on Application Server
Set up a connection to the database server
Test the connection to the database server
Legend:
: Required operation

### Related topics

- 4.7.2 Installing the JDBC driver for Application Server
- 4.7.3 Setting up a connection to the database server
- 4.7.4 Testing the connection to the database server

## 4.7.2 Installing the JDBC driver for Application Server

To install the JDBC driver for Application Server, copy the `.jar` file of the JDBC driver (provided by the database vendor) to the Application Server directory. In order to enable the JDBC driver for Application Server, restart the domain administration server (DAS) by using the `restart-domain` subcommand of the `asadmin` utility command, and Application Server by using the `stop-servers` and `start-servers` subcommands.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The database server (DBMS) is running.

### Intended users

- System engineers

## Procedure

1. Copy the `.jar` file of the JDBC driver to the Application Server directory.

   Use the `.jar` file (of the JDBC driver) that is provided by the database vendor. Copy the file to *installation_directory_for_Application_Server*/javaee/glassfish/domains/*domain_name*/lib.

2. To restart the DAS, run the `restart-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin restart-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command restart-domain executed successfully.
   ```

3. To stop Application Server run the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

4. To start Application Server run the `start-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin start-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-servers executed successfully.
   ```

## Postrequisites

- Set a connection to the database server.

### Related topics

- 4.7.3 Setting up a connection to the database server

## 4.7.3 Setting up a connection to the database server

To set up a connection to the database server, create a JDBC connection pool using the `create-jdbc-connection-pool` subcommand of the `asadmin` utility command, and then create a JDBC resource using the `create-jdbc-resource` subcommand.

## Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The database server (DBMS) is running.
- The JDBC driver has been installed for Application Server.

## Intended users

- System engineers

## Procedure

1. To create a JDBC connection pool, run the `create-jdbc-connection-pool` subcommand of the `asadmin` utility command.

```
asadmin create-jdbc-connection-pool connection_pool_setting
unique_connection_destination_vendor_setting connection_pool_ID
```

   - The size of the connection pool and the connection trouble detection function are set in the *connection_pool_setting* by using the subcommand options.
   - The driver-specific information provided by the database vendor is set in *unique_connection_destination_vendor_setting* by using the subcommand options.
   - The name used to identify the JDBC connection pool is set in *connection_pool_ID*.

   (Example 1) For Oracle:

```
asadmin create-jdbc-connection-pool
--datasourceclassname oracle.jdbc.pool.OracleDataSource
--restype javax.sql.DataSource
--property user=user_name:password=password:
url="jdbc\:oracle\:thin\:@IP_address\:port_number\:Oracle_SID"
connection_pool_name
```

   When this command is run, the result is displayed as follows:

```
JDBC connection pool connection_pool_ID created successfully.
Command create-jdbc-connection-pool executed successfully.
```

2. To create a JDBC resource for all server instances in a cluster, run the `create-jdbc-resource` subcommand of the `asadmin` utility command.

```
asadmin create-jdbc-resource --connectionpoolid connection_pool_ID
--target server_instance_name_or_cluster_name JNDI_name
```

   - In a configuration where only one Java EE server is deployed, the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.
   - The name required to acquire the connection of the database server (DBMS) is specified in *JNDI_name*.

   When this command is run, the result is displayed as follows:

```
JDBC resource JNDI_name created successfully.
Command create-jdbc-resource executed successfully.
```

## Postrequisites

- Testing the connection to the database server

**Related topics**

- 4.7.4 Testing the connection to the database server

## 4.7.4 Testing the connection to the database server

To test the connection to the database server, verify whether a connection can be established from the server instance to the database server by running the `ping-connection-pool` subcommand of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The database server (DBMS) is running.
- Setup to connect to the database server is complete.

### Intended users

- System engineers

### Procedure

1. To confirm whether a connection can be established to the database server from the server instance, run the `ping-connection-pool` subcommand of the `asadmin` utility command.

```
asadmin ping-connection-pool --targetserver_instance_name
connection_pool_ID
```

When this command is run, the result is displayed as follows:

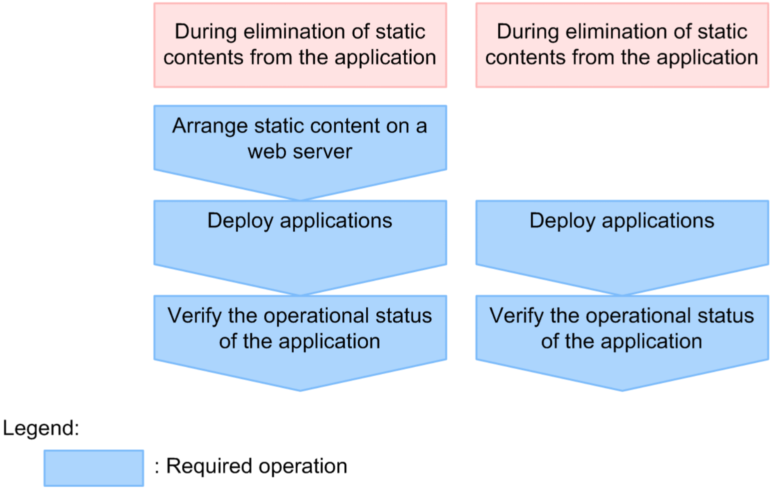
```
Command ping-connection-pool executed successfully.
```

## 4.8 Deploying an application

This section describes the procedure for deploying an application. The procedure for deploying an application is as follows: separate static content from the application, arrange the static content on the web server, and then deploy the application (dynamic contents) on the Java EE server (server instance). The application starts after it has been deployed.

### 4.8.1 Workflow for deploying an application

This topic describes the processes for deploying an application, and the relevant flow of tasks. You must separate the static content from the application and then arrange the static content on the web server to improve performance of the system. You must deploy the application (dynamic contents) in the Java EE server (server instance). After deploying the application, check the running status to verify whether the application has started.


During elimination of static contents from the application
During elimination of static contents from the application
Arrange static content on a web server
Deploy applications
Deploy applications
Verify the operational status of the application
Verify the operational status of the application
Legend:
: Required operation

#### Related topics

- 4.8.2 Deploying static content on the web server
- 4.8.3 Deploying applications
- 4.8.4 Confirming the operational status of applications

### 4.8.2 Deploying static content on the web server

To deploy static content on the web server, store the content in the document root directory of the web server. By reducing the frequency at which the network is accessed and the size of the data that is sent, the performance of applications will improve.

#### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The relevant applications have been obtained from the application developers.

## Intended users

- System engineers

## Procedure

1. Store the static content in the document root directory of the web server.

   The document root directory of the web server is `installation_directory_for_Application_Server/javaee/glassfish/nodes/node_name/web_server_name/root/htdocs`.

## Postrequisites

- Deploying applications

### Related topics

- 4.8.3 Deploying applications

## 4.8.3 Deploying applications

To deploy an application, execute the `deploy` subcommand of the `asadmin` utility command. To start the application, you must deploy it on a Java EE server (server instance).

## Prerequisites

- The domain administration server (DAS) is running.
- Setup of Application Server is complete.
- The relevant applications have been received from the application developers.

## Intended users

- System engineers

## Procedure

1. To deploy the application to a server instance, run the `deploy` subcommand of the `asadmin` utility command.

   ```
   asadmin deploy --target server_instance_name_or_cluster_name
   file_path_of_application
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   When this command is run, the result is displayed as follows:

   ```
   Application deployed with name application_name.
   Command deploy executed successfully.
   ```

2. Repeat step 1 for each application that you want to deploy.

## Postrequisites

- Confirming the running state of applications.

### Related topics

- 4.8.4 Confirming the operational status of applications

## 4.8.4 Confirming the operational status of applications

To confirm the operational status of applications, display the list of application by running the `list-applications` subcommand of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The relevant applications have been deployed.

### Intended users

- System engineers

### Procedure

1. To display a list of applications deployed to the server instance, run the `list-applications` subcommand of the `asadmin` utility command with the `--long` option.

```
asadmin list-applications --long=true server_instance_name_or_cluster_name
```

   - In a configuration where only one Java EE server is deployed, the server instance name for the *server_instance_name_or_cluster_name*.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for *server_instance_name_or_cluster_name*.

   When this command is run, the results are displayed as follows: Verify that the status of all applications that are deployed is `enabled`.

```
NAME                    TYPE        STATUS
application_name  <ear, web>     enabled
application_name  <web>          enabled
Command list-applications executed successfully.
```

## 4.9 Verifying system-configuration information

This section describes the procedure for verifying the system-configuration information. The information includes the following details: setup of the host on which Application Server is configured, setup of each server on Application Server, the settings for each server, and so on. Verify this information in either the files that are saved by using commands or in the pane that is displayed in the Administration Console.

## 4.9.1 Confirming the system settings information

The system settings information indicates information such as the configuration information about the host on which Application Server is configured, configuration information about the servers within Application Server, and the setting values of each server. This topic describes the following: the types of system setting information and how to confirm the information, and the system configuration after setting up the Application Server after connecting to the database server, and after deploying applications.

### Types of system settings information and how to confirm it

The types of system settings information include the following:

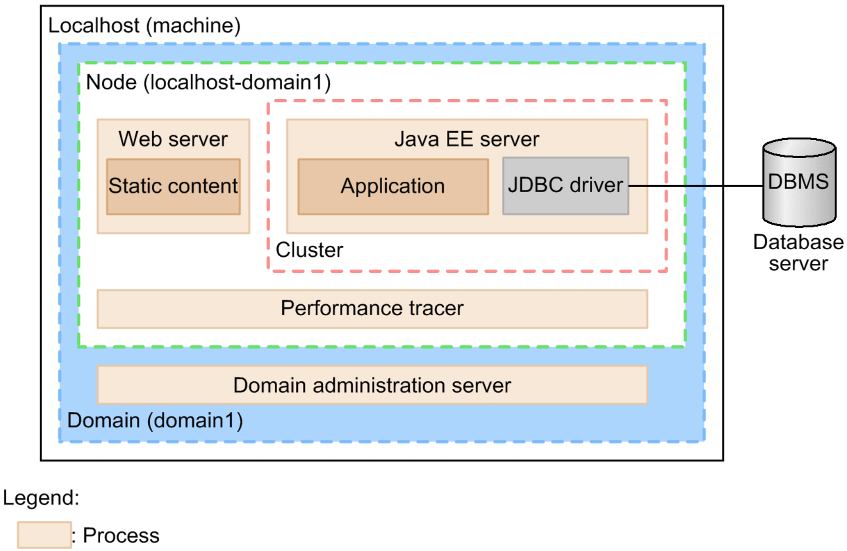
- Names and setting values for the system properties, other properties, and definition items for files that are specified on the server.
- Running status and the configuration information of the server.
- Configuration information for the domain administration server (DAS) and the server, and basic information such as name, type, and node.

This information is verified using the command screen or Administration Console. The settings information of the web server is verified using the server template. If commands are used, you can save the settings information in a file.

### System configuration after the deployment of applications

The following figure displays the system configuration when the tasks that are listed have been completed after the installation of Application Server:

1. Set up Application Server.
2. Establish a connection to the database server.
3. Deploy applications.



## 4.9.2 Confirming system settings information by using commands

To confirm the system settings information by using commands, generate a list file using the subcommands that start with list- or the `get` subcommands of the `asadmin` utility command. Verify that the contents of the generated file match the intended setting. To refer to the standard properties and the settings of the extended properties, use the `get` subcommand. Verify the information that is configured in the template except for the properties that are set by the server template.

### Prerequisites

- Application Server has been set up.

### Intended users

- System engineers

### Procedure

1. To save a list of all the Java EE servers in the domain, along with detailed information about the host names and port numbers in a file, run the `list-instances` subcommand of the `asadmin` utility command.

```
asadmin list-instances --long=true
>> file_path_of_output_file
```

2. To save a list of all the performance tracers in the domain, along with detailed information about the host names and process IDs in a file, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs --long=true
>> file_path_of_output_file
```

3. To save a list of all the web servers in the domain, along with detailed information about the host names and process IDs in a file, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers --long=true >> File_path_of_output_file
```

4. To save a list of all the clusters in the domain in a file, run the `list-clusters` subcommand of the `asadmin` utility command.

```
asadmin list-clusters >> file_path_of_output_file
```

5. To save a list of all the dependency relations in the domain, along with detailed information about the relation types, sources, and destinations in a file, run the `list-relations` subcommand of the `asadmin` utility command.

```
asadmin list-relations --long=true >> file_path_of_output_file
```

6. To save the setup information in a file, run the subcommands, which start with list-, of the `asadmin` utility command.

   (Example)

   To save a list of Java VM options and system properties in a file, run the `list-jvm-options` and `list-system-properties` subcommands.

```
asadmin list-jvm-options --target server_instance_name_or_cluster_name
>> file_path_of_output_file
asadmin list-jvm-options --target configuration_name >> file_path_of_output_file
asadmin list-system-properties server_instance_name_or_cluster_name
>> file_path_of_output_file
asadmin list-system-properties Configuration_name >> file_path_of_output_file
```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for *server_instance_name_or_cluster_name*.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for *server_instance_name_or_cluster_name*.

7. To save the setup information of each server in a file, run the `get` subcommand of the `asadmin` utility command.

```
asadmin get "*" >> file_path_of_output_file
```

8. Verify whether the contents of the files that are saved in steps 1 through 7 are the same as the system configuration information.

**Reference note**

Refer to the configuration values of all the standard and extended properties by using the `get` subcommand. For the information that is set by using a server template and not by the standard or extended properties, you must verify the server template.

## 4.9.3 Verifying the settings information of the web server from the server template

Using the contents of the server template, verify the settings information of the web server.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

## Intended users

- System engineers

## Procedure

1. To determine the configuration name of the web server, run the `get` subcommand of the `asadmin` utility command.

   ```
   asadmin get hitachi-webservers.hitachi-webserver.web_server_name.
   hitachi-webserver-config-ref
   ```

2. To acquire the file path of the server template, run the `get` subcommand of the `asadmin` utility command specifying the configuration name of the web server acquired in step 1.

   ```
   asadmin get hitachi-webserver-configs.hitachi-webserver-config.
   configuration_name_of_the_web_server.hitachi-manage-info.template-path
   ```

   The `${com.sun.aas.instanceRoot}` to be included in the file path that was acquired in step 2, displays *`installation_directory_for_Application_Server`*`/javaee/glassfish/domains/`*`domain_name`*.

3. Open the server template that is available in the file path acquired in step 2, and verify that the contents match the settings information.

## 4.9.4 Verifying the system settings information by using the Administration Console

To verify the system settings information in the Administration Console, use the **Configuration** tab. Verify whether the contents of the screen have been set as required.

### Prerequisites

- Application Server has been set up.

### Intended users

- System engineers

### Procedure

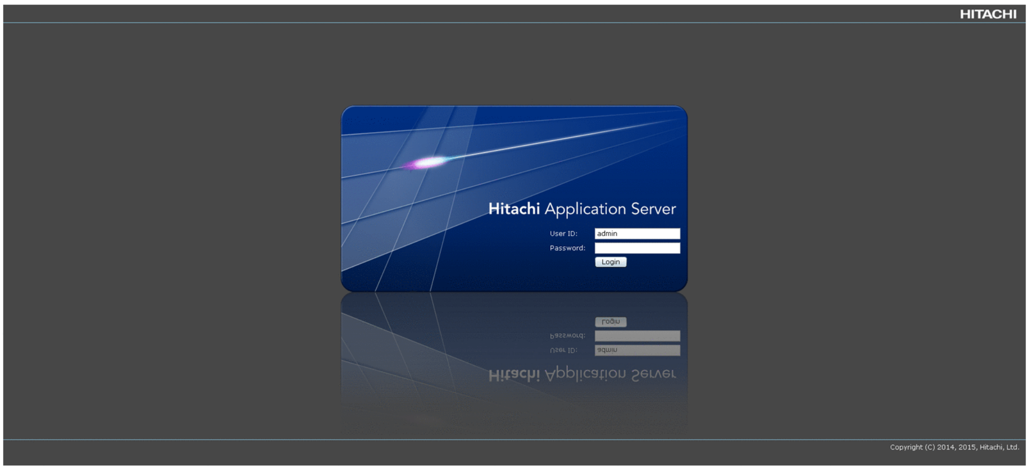
1. Start the web browser and enter the following URL to start Administration Console:

   ```
   http://IP_address_of_the_domain_administration_server:
   HTTP_port_number_of_the_domain_administration_server/admin/
   ```

   > **Reference note**
   >
   > The default value of *IP_address_of_the_domain_administration_server* is `127.0.0.1`, and the default value of *HTTP_port_number_of_the_domain_administration_server* is `8080`. When both values are left as the default, you can start Administration Console from the Start menu or the start screen of the terminal.

However, if you use the built-in Administrator account in Windows Server 2012, Windows Server 2012 R2, or Windows 8, the Modern UI version of Internet Explorer is unable to start for security reasons, and an error message prompting you to sign in again using a different account might be displayed. In this case, from the **Tools** menu of Internet Explorer, select **Internet Options**. Then in the **Programs** tab, select **Always in Internet Explorer on the desktop** for **Choose how you open links**, and then restart Administration Console.

2. Input the user ID in the **User ID** text box and the password in the **Password** text box, and then click the **Login** button.


HITACHI
Hitachi Application Server
Login
Copyright (C) 2014, 2015, Hitachi, Ltd.

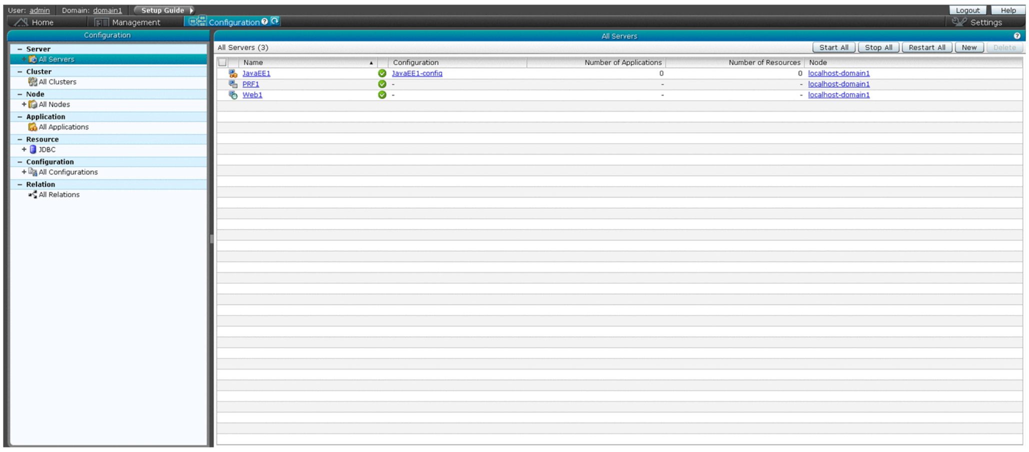
User ID: admin

Password:

## Reference note

The default value for the user ID is `admin`, and there is no default value for the password.

3. In the Administration Console, click the **Configuration** tab.



All Servers (3)

| Name | Configuration | Number of Applications | Number of Resources | Node |
|---|---|---|---|---|
| JavaEE1 | JavaEE1-config | 0 | 0 | localhost-domain1 |
| PRF1 | - | - | - | localhost-domain1 |
| Web1 | - | - | - | localhost-domain1 |

4. To verify the system settings information of the Java EE server, performance tracer, and web server, do the following:

   a. In the **Configuration** tab, click the **All Servers** in the navigation pane.

   b. In the **All Servers** pane, in the **Name** column, click the server name (link) for which you want to verify the system settings information.

   c. Verify whether the system settings information of the server is as set in the **General** tab.

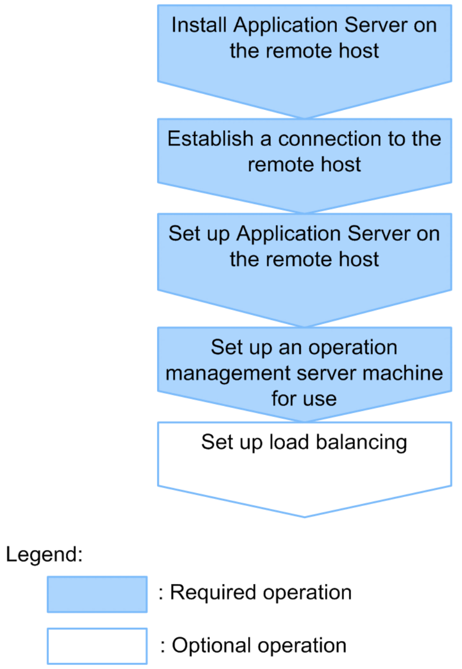
5. To verify the settings information of the cluster, do the following:
    a. In the **Configuration** tab, click the **All Clusters** node in the navigation pane.
    b. In the **All Clusters** pane, in the **Name** column, click the cluster name (link) for which you want to verify the system settings information.
    c. Verify whether the system settings information of the cluster is as set in the **General** tab.
6. To verify the settings information of the dependency relations, do the following:
    a. In the **Configuration** tab, click the **All Relations** node in the navigation pane.
    b. Verify whether the system settings information of the relationship between logical servers is as set in the **Relation** pane.

## 4.10 Configuring Application Server on a remote host

This section describes how to set up a domain administration server for use as an operation management server machine, separate from Application Server, and how to build a system in a cluster configuration. A "cluster configuration" is a configuration for managing multiple Java EE servers by grouping. If Application Server is configured in a cluster configuration, then a request can be distributed to multiple Application Servers, thus maintaining the reliability and availability of the system. Either a software or hardware load balancer is used for load balancing.

### 4.10.1 Workflow for configuring an application execution environment on a remote host

This section describes the processes for configuring an operation management server machine and a system in a cluster configuration, and also describes the workflow for these processes. A "*cluster configuration*" is a configuration for managing multiple Java EE servers by grouping. To configure an operation management server machine and a system in a cluster configuration, you must install and set up Application Server on a remote host. Additionally, you must set up the local host for use as an operation-management server machine.


Install Application Server on the remote host
Establish a connection to the remote host
Set up Application Server on the remote host
Set up an operation management server machine for use
Set up load balancing
Legend:
: Required operation
: Optional operation

#### Related topics

- 4.10.2 Installing Application Server on a remote host
- 4.10.3 Connecting to a remote host
- 4.10.4 Setting up Application Server on a remote host
- 4.10.5 Settings for using the operation management server machine
- 5.1 System environment settings

## 4.10.2 Installing Application Server on a remote host

To install Application Server on a new remote host, start the Program Product Installer from the provided media, and then select the program to install from the program product installation window. After the installation is complete, set the environment settings.

### Prerequisites

- The system engineer has root privileges (super user).
- Remote host is running.
- A prerequisite operating system (OS) and patches are installed.

### Intended users

- System engineers

### Procedure

1. Ensure that the packages listed in the following table have been applied.

Table 4-5: Package to be applied for each OS

| Package Name | Red Hat Enterprise Linux 5 | Red Hat Enterprise Linux Server 6 |
|---|---|---|
| compat-libstdc++-296 (i386) | Y | -- |
| compat-libstdc++-33 (i386) | Y | -- |
| compat-libstdc++-33 (i686) | -- | Y |
| compat-libstdc++-33 (x86_64) | Y | Y |
| coreutils (x86_64) | Y | Y |
| findutils (x86_64) | Y | Y |
| gdb (x86_64) | Y | Y |
| glibc (i686) | Y | Y |
| glibc (i686) 2.5-24 or later | Y[#1] | -- |
| glibc (x86_64) | -- | Y |
| glibc (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-common (x86_64) | -- | Y |
| glibc-common (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-devel (i386) 2.5-24 or later | Y[#1] | -- |
| glibc-devel (i686) | -- | Y |
| glibc-devel (x86_64) | -- | Y |
| glibc-devel (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-headers (x86_64) | -- | Y |
| glibc-headers (x86_64) 2.5-24 or later | Y[#1] | -- |
| glibc-utils (x86_64) | -- | Y |

| Package Name | Red Hat Enterprise Linux 5 | Red Hat Enterprise Linux Server 6 |
|---|---|---|
| glibc-utils (x86_64) 2.5-24 or later | Y#1 | -- |
| gzip (x86_64) | Y | Y |
| ksh (x86_64) | -- | Y |
| libgcc (i386) | Y | -- |
| libgcc (i686) | -- | Y |
| libstdc++ (i386) | Y | -- |
| libstdc++ (i686) | -- | Y |
| lksctp-tools (x86_64) | Y | Y |
| ncompress (x86_64) | Y | Y#2 |
| ncurses (x86_64) | Y | Y |
| net-tools (x86_64) | Y | Y |
| nscd (x86_64) | -- | Y |
| nscd (x86_64) 2.5-24 or later | Y#1 | -- |
| nss-softokn-freebl (i686) | Y | Y |
| procps (x86_64) | Y | Y |
| rpm (x86_64) | Y | Y |
| sysstat (x86_64) | Y | Y |
| tar (x86_64) | Y | Y |

Legend

Y: Apply this package

--: Not applicable

#1

These packages might not be included in the installation media, depending on the version of Red Hat Enterprise Linux 5 that is being used.

#2

If Red Hat Enterprise Linux Server 6.2 is used, ncompress (x86_64) version 4.2.4-54.el6_2.1 or later needs to be applied.

To verify whether a package is installed, execute the `rpm` command. The following example displays both the command execution and execution result of the `rpm` command:

Example of execution:

```
#rpm -q --qf  '%{NAME}-%{ARCH}\n'  ncompress
```

Example of execution result:

```
ncompress-x86_64
```

This example displays that the 64 bit ncompress package was installed.

If the message `Package` *`package_name`* `is not installed` is displayed, this indicates that the package was not installed. In such a case, install the package displayed in *package_name*. During the installation, also install related packages as required.

2. Verify whether the language type of the Program Product Installer matches the language on the terminal where you will perform the installation. If the languages do not match, make them the same.

3. Insert the provided media of the product into the corresponding drive.

4. Run the `mount` command to mount the file system of the corresponding media.

   In Linux, use the following code:

   ```
   mount -r -o mode=0544 /dev/cdrom
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   Note that *special_file_name_of_device* and *mount_directory_name_of_filesystem_for_corresponding_media* vary depending on the operating system (OS), hardware, and environment.

5. Start the setup program from the media of the product.

   In Linux, use the following code:

   ```
   mount_directory_name_of_filesystem_for_corresponding/x64lin/setup -m
   mount_directory_name_of_filesystem_for_corresponding_media
   ```

   **Important note**

   The details of the directory name and file name of the corresponding media and how to view this information, might differ depending on the computer environment. Use the `ls` command to verify this information, and enter the displayed file name as is.

   The setup program installs the Program Product Installer and program required to start resident processes automatically, on the hard disk. The Program Product Installer starts automatically.

6. On the main menu of the Program Product Installer, press the **I** key.

7. In the program product installation window, move the cursor over the name of the product to be installed (`Hitachi Application Server`), and then press **Spacebar** key.

   A (`<@>`) icon, which indicates the selection state appears on the left side of the selected program.

8. To select and install the programs that will compose the product, move the cursor to the programs that you do not want to install, and then press the **Spacebar** key.

   The `[@]` icon appears on the left side of programs that are optional. When you press the **Spacebar** key, the icon within `[]` changes, and those programs will not be installed.

   Note that the `@` icon is displayed on the left side of programs that are mandatory.

9. Ensure that the `<@>` icon is displayed on the left side of the product to be installed and that `[@]` or `@` is displayed on the left side of the programs to be installed. Then, press the **I** key.

   A message `Install PP? (y: install, n: cancel)==>` appears at the bottom of the window.

10. Press the **y** or **Y** key.

    A message `Enter the installation path ==>`*`default_installation_destination`* appears at the bottom of the window.

    If you press the **Ctrl**+**n** or **Ctrl**+**N**, then the installation will stop and you will return to the program product installation window.

11. Verify the *default_installation_destination* value and change the installation destination directory as required. Then, press the **Enter** key.

    You can specify only alphanumeric characters, hyphens (`-`), and underscores (_) in the installation destination.

    If the specified installation directory does not exist, a message confirming the creation of the directory appears.

    ```
    The specified path does not exist. Do you want to create the path?
    Specified path: installation_destination
    ```

    Verify the *installation_destination* and then press the **y** or **Y** key.

    A message `Enter a display name ==> `*`default_display_name`* appears at the bottom of the window.

    To reconfigure the installation destination, press the **Ctrl**+**B** keys.

12. Verify the *default_display_name*, change the display name as required, and then press the **Enter** key.

    A maximum of 40 characters can be specified for the display name. You can only specify alphanumeric characters, hyphens (`-`), and underscores (_).

    The message asking whether you want to continue with the installation with the specified display name and installation destination is displayed in the window.

    ```
    This program product will be installed at the specified path,
    with the specified display name.
    Do you want to continue?
    Specified display name: display_name
    Specified path        : installation_directory
    ```

    To change the display name or installation destination, press the **n** or **N** key. You will be returned to step 10.

13. Verify the display name and installation destination, and then press the **y** or **Y** key.

    The installation starts.

14. When a message indicating the end of the installation appears, press the **Q** key.

15. On the main menu of the Program Product Installer, press the **L** key.

    The program product list window displays a list of the installed products.

16. Verify whether the product was installed successfully, and then press the **Q** key.

17. On the main menu of the Program Product Installer, press the **Q** key.

    The Program Product Installer closes, and the installation of Application Server is complete.

18. Set the Time Zone (TZ) environment variable in the configuration file (`asenv.conf`).

    The configuration file (`asenv.conf`) is stored in *`installation_directory_for_Java_EE_Server`*`/glassfish/config/asenv.conf`. The following is a definition example of a configuration file (`asenv.conf`):

    ```
    TZ=JST-9
    ```

19. Allow the IP address to be resolved from the local host name by specifying an appropriate IP address assigned for the network interface for the local host name in the `/etc/hosts` file.

20. Repeat steps 1 to 19 for each remote host that needs to be configured as a request distribution destination.

### Postrequisites

- Set up Application Server to connect to a remote host.

#### Related topics

- 4.10.3 Connecting to a remote host

## 4.10.3 Connecting to a remote host

To configure a cluster, you must configure the local host and remote host to connect with SSH. On the local host, you can establish the SSH connection using the `setup-ssh` subcommand of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up on the localhost.
- The remote host is running.
- Installation of Application Server is complete on the remote host.
- The remote host and local host can be connected using SSH.

### Intended users

- System engineers

### Procedure

1. To resolve the name of the host with the DAS (local host) on a remote host, edit the host file.

2. On the local host, create a password file.

   ```
   AS_ADMIN_SSHPASSWORD=remote_host_password
   ```

3. To create the code key (SSL key), run the `setup-ssh` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin --passwordfile password_file_path setup-ssh
   --sshuser remote_host_SSH_user_name remote_host_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command setup-ssh executed successfully.
   ```

4. To set an alias for the password, run the `create-password-alias` subcommand of the `asadmin` utility command on the local host.

   ```
   asadmin create-password-alias password_alias_name
   ```

   Specify the password of the remote host when it is requested.

   When this command is run, the result is displayed as follows:

   ```
   Command create-password-alias executed successfully.
   ```

5. Restart the domain administration server.

   ```
   asadmin restart-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command restart-domain executed successfully.
   ```

6. Edit the password file available on the local host created in Step 2.

   ```
   AS_ADMIN_SSHPASSWORD=${ALIAS=password_alias_name}
   ```

7. For each remote host to be configured as a distribution location of a request, repeat the procedure from steps 1 through 6 on each remote host.

### Postrequisites

- Setting up Application Server on the remote host.

### Related topics

- 4.10.4 Setting up Application Server on a remote host

## 4.10.4 Setting up Application Server on a remote host

To set up Application Server on a remote host, use the `create-node-ssh` subcommand to create a node for the remote host. After you create the node, configure the performance tracer and Java EE server (server instance) on the remote host by using the `create-prf` and `create-instance` subcommands respectively. If you plan to use a hardware load balancer, use the `create-webserver` subcommand to configure the web server on the remote host. If you plan to use a software load balancer, use the `delete-webserver` subcommand to delete the web server on the localhost. Run these subcommands on the localhost where the domain administration server (DAS) is running.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up on the localhost.
- Application Server has been installed on the remote host.
- The remote host and localhost have been connected by using SSH.

## Intended users

- System engineers

## Procedure

1. To add a node to the remote host, run the `create-node-ssh` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin --user domain_administration_server_user_name
   --passwordfile password_file_path
   create-node-ssh --nodehost remote_host_address
   --installdir installation_directory_path_for_Application_Server/
   javaee_absolute_path
   --sshuser remote_host_SSH_user_name
   --sshkeyfile ~/.ssh/id_rsa remote_host_node_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command create-node-ssh executed successfully.
   ```

2. To set up the performance tracer on the remote host, run the `create-prf` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin create-prf --node remote_host_node_name
   performance_tracer_name_in_remote_host
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command create-prf executed successfully.
   ```

3. To set up the Java EE server on the remote host, run the `create-instance` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin create-instance --node remote_host_node_name
   --prf performance_tracer_name_in_remote_host
   --cluster cluster_name server_instance_name_in_remote_host
   ```

   In the *cluster_name*, specify the name of the cluster that was specified when setting up Application Server on the localhost.

   When this command is run, the result is displayed as follows:

   ```
   Command create-instance executed successfully.
   ```

4. If you plan to use a hardware load balancer, set up the web server on the remote host.

   a. To set up the web server on the remote host, run the `create-webserver` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin create-webserver --node remote_host_node_name
   --prf performance_tracer_name web_server_name_of_remote_host
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command create-webserver executed successfully.
   ```

b. When specifying the extended properties of the web server that was created on the localhost, to specify the extended properties of the web server on the remote host, run the `set` subcommand of the `asadmin` utility command on the localhost.

> **Reference note**
>
> In the `create-webserver` subcommand, which is run in the previous step, the server template settings of the localhost are applied to the web server that was created on the remote host. However, the settings of the extended properties are not applied.

```
asadmin set hitachi-webservers.hitachi-webserver.
web_server_name_in_remote_host.property.extended_property_name=value
```

When this command is run, the result is displayed as follows:

```
hitachi-webservers.hitachi-webserver.web_server_name_in_remote_host.property.
extended_property_name=value
Command set executed successfully.
```

c. To associate the Java EE server (server instance), which is used as the redirector destination of the request received by the web server, run the `create-relation` subcommand of the `asadmin` utility command on the localhost.

```
asadmin create-relation --relationtype redirect
--from web_server_name_in_remote_host
--to server_instance_name_in_remote_host
--properties property_name_of_dependency_relation = value
dependency_relation_name_in_remote_host
```

While setting up the redirection-relation association, in *property_name_of_dependency_relation* = *value*, specify `path` and `network-listener`.

Example:

`path=/:network-listener=http-listener-1`

When this command is run, the result is displayed as follows:

```
Command create-relation executed successfully.
```

d. If the static content is deployed on the web server of the localhost, store the static content in the document root directory of the web server of the remote host.
The web server document root directory of the remote host is `installation_directory_for_Application_Server_in_remote_host/javaee/glassfish/nodes/remote_host_node_name/web_server_name_in_remote_host/root/htdocs`.

5. Repeat steps 1 to 4 for each remote host that is used as the distribution destination of requests.

6. When Web Server is used as a software load balancer, delete the web server of the localhost.

a. For planned termination of the web server of the localhost, run the `stop-webserver` subcommand of the `asadmin` utility command, on the localhost.

```
asadmin stop-webserver --graceful true web_server_name_in_localhost
```

When this command is run, the result is displayed as follows:

```
Command stop-webserver executed successfully.
```

b. To delete associations of redirector relations that are not required, run the `delete-relation` subcommand of the `asadmin` utility command, on the localhost.

```
asadmin delete-relation dependency_relation_name_in_local_host
```

Because the web server of the destination that receives the request becomes a software load balancer, the web server created on the localhost will not be required anymore. For *dependency_relation_name_in_local_host*, specify the relation dependency that were created when Application Server was set up on the localhost.

When this command is run, the result is displayed as follows:

```
Command delete-relation executed successfully.
```

c. To delete unnecessary web servers, run the `delete-webserver` subcommand of the `asadmin` utility command on the localhost.

```
asadmin delete-webserver web_server_name_in_localhost
```

In *web_server_name_in_localhost*, specify the name of the web server created while setting up Application Server on the localhost.

When this command is run, the result is displayed as follows:

```
Command delete-webserver executed successfully.
```

### Postrequisites

- Specifying settings for using the operation management server

### Related topics

- 4.10.5 Settings for using the operation management server machine

## 4.10.5 Settings for using the operation management server machine

The operation management server machine is the machine on which only the domain administration server (DAS) operates. To use the operation management server machine, delete all other servers that make up Application Server from the local host, so that only the DAS is running on it. To delete each server from Application Server, run the `delete-instance`, `delete-webserver`, and `delete-prf` subcommands of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Setup of Application Server is complete on both the remote host and localhost.
- The remote host and localhost can be connected by using SSH.

### Intended users

- System engineers

## Procedure

1. To stop the Java EE server (server instance) on the local host, run the `stop-instance` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin stop-instance server_instance_name_in_localhost.
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-instance executed successfully.
   ```

2. To stop the performance tracer on the localhost, run the `stop-prf` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin stop-prf performance_tracer_name_in_localhost.
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-prf executed successfully.
   ```

3. To use a hardware load balancer, delete the connection that is established with the localhost Application Server by running the `delete-relation` subcommand of the `asadmin` utility command on the local host.

   ```
   asadmin delete-relation dependency_relation_name_in_localhost
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command delete-relation executed successfully.
   ```

4. To delete the Java EE server on the local host, run the `delete-instance` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin delete-instance server_instance_name_in_localhost
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command delete-instance executed successfully.
   ```

5. To use a hardware load balancer, delete the web server on the localhost.

   a. To perform a planned termination of the web server, run the `stop-webserver` subcommand of the `asadmin` utility command on the local host.

   ```
   asadmin stop-webserver --graceful true web_server_name_in_localhost
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-webserver executed successfully.
   ```

   b. To delete the web server on the localhost, run the `delete-webserver` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin delete-webserver web_server_name_in_localhost
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command delete-webserver executed successfully.
   ```

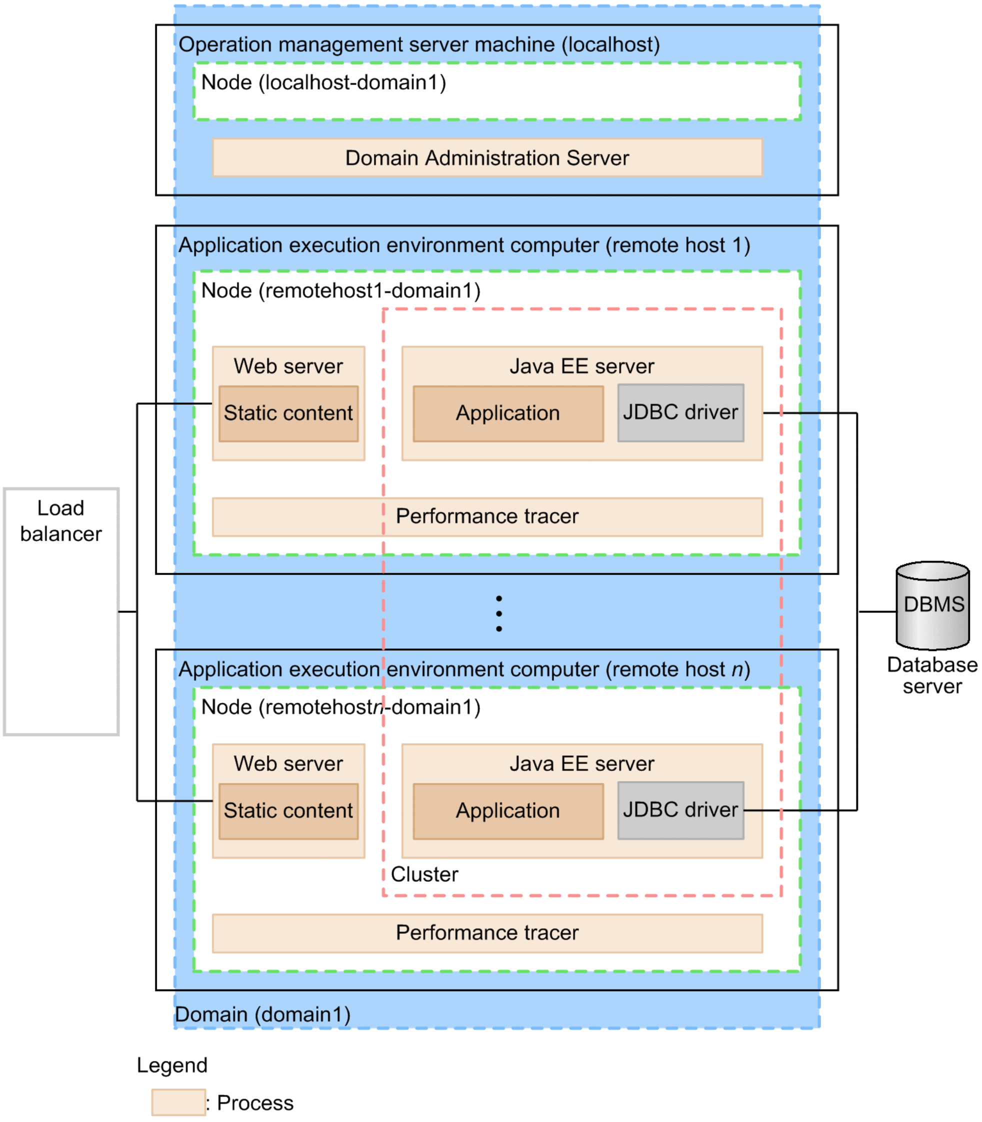
6. To delete the performance tracer on the local host, run the `delete-prf` subcommand of the `asadmin` utility command on the localhost.

```
asadmin delete-prf performance_tracer_name_in_localhost
```

When this command is run, the result is displayed as follows:

```
Command delete-prf executed successfully.
```

The following example displays the configuration.



## 4.11 Deleting and uninstalling Application Server

This section describes the procedure to delete or uninstall Application Server.

### 4.11.1 Deleting Application Server

To delete Application Server, stop each server, and then run the `delete-relation`, `delete-instance`, `delete-webserver`, and `delete-prf` subcommands of the `asadmin` utility command. Each command deletes the dependency relations, Java EE server, web server, and performance tracer. Before deleting Application Server, be sure to back up the environment information.

#### Prerequisites

- Application Server has been set up.

#### Intended users

- System engineers

#### Procedure

1. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

```
asadmin stop-servers
```

When this command is run, the result is displayed as follows:

```
Command stop-servers executed successfully.
```

2. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

```
performance_tracer_name not running
Command list-prfs executed successfully.
```

3. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

```
server_instance_name host_name port_number process_ID cluster_name not running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

4. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

   ```
   web_server_name not running
   Command list-webservers executed successfully.
   ```

5. If required, back up the environment information as follows:

   a. To stop the domain administration server (DAS), run the `stop-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-domain executed successfully.
   ```

   b. To back up the environment information, run the `backup-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin backup-domain --backupdir destination_directory_of_backup_file
   domain_name
   ```

   Verify the backup file that is created.

   c. To start the DAS, run the `start-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin start-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-domain executed successfully.
   ```

6. Back up all management files (server templates) except the domain directory.

7. To delete the dependency relation, run the `delete-relation` subcommand of the `asadmin` utility command.

   ```
   asadmin delete-relation dependency_relation_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command delete-relation executed successfully.
   ```

8. To delete the Java EE server (server instance), run the `delete-instance` subcommand of the `asadmin` utility command.

   ```
   asadmin delete-instance server_instance_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command delete-instance executed successfully.
   ```

9. To delete the web server, run the `delete-webserver` subcommand of the `asadmin` utility command.

   ```
   asadmin delete-webserver web_server_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command delete-webserver executed successfully.
   ```

10. To delete the performance tracer, run the `delete-prf` subcommand of the `asadmin` utility command.

    ```
    asadmin delete-prf performance_tracer_name
    ```

    When this command is run, the result is displayed as follows:

    ```
    Command delete-prf executed successfully.
    ```

11. To delete the node, run the `delete-node-ssh` subcommands of the `asadmin` utility command.

    ```
    asadmin delete-node-ssh node_name
    ```

    When this command is run, the result is displayed as follows:

    ```
    Command delete-node-ssh executed successfully.
    ```

## 4.11.2 Uninstalling Application Server

To uninstall Application Server, start the Program Product Installer and select the program that you want to uninstall in the program product installation window.

### Prerequisites

- The system engineer has root privileges (super user).
- Application Server has been set up.

### Intended users

- System engineers

### Procedure

1. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

2. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

```
performance_tracer_name not running
Command list-prfs executed successfully.
```

3. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

```
server_instance_name host_name port_number process_ID cluster_name not running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

4. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

```
web_server_name not running
Command list-webservers executed successfully.
```

5. To back up the environmental information, if required, run the `backup-domain` subcommand of the `asadmin` utility command.

```
asadmin backup-domain --backupdir backup_file_save_location_directory
domain_name
```

Verify the backup file that is displayed.

6. Back up the files that are managed somewhere other than the domain directory (for example, server templates).

7. To stop a domain, run the `stop-domain` subcommand of the `asadmin` utility command.

```
asadmin stop-domain domain_name
```

When this command is run, the result is displayed as follows:

```
Command stop-domain executed successfully.
```

8. To start the Program Product Installer, run the following command:

For Linux, use the following code:

```
/etc/hitachi_x64setup
```

9. On the main menu of the Program Product Installer, press the **D** key.

10. To select and uninstall the product (Application Server), in the program product uninstallation window, move the cursor to the product to be uninstalled, and then press the **Spacebar** key.

    An icon that indicates the selection state `<@>` appears on the left side of the product to be uninstalled.

    An icon that indicates the selection state ( `[@]` or `@`) appears on the left side of the programs that compose the product to be uninstalled.

    If multiple instances of the same product are installed, then verify the display name (**DisplayName**) of the product to be uninstalled.

11. To select and uninstall the programs that compose the product, in the program product uninstallation window, move the cursor to the programs to be uninstalled, and then press the **Spacebar** key.

    An icon appears on the left side of the programs you can uninstall. When you press the **Spacebar** key, the `[@]` icon changes, and that program will be uninstalled.

12. Ensure that the `<@>` icon is displayed on the left side of the product to be uninstalled and then, press the **D** key.

    A message `Delete PP? (y: delete, n:cancel)==>` appears at the bottom of the window.

13. Press the **y** or **Y** key.

    The uninstallation starts.

    If you press the **n** or **N** key, then the uninstallation will stop.

14. When a message indicating the end of the uninstallation appears, press the **Q** key.

15. On the main menu of the Program Product Installer, press the **Q** key.

    The Program Product Installer closes, and the uninstallation of Application Server is complete.

# 5 Specifying system environment settings

This chapter describes system environment settings that can be specified when needed. Network, load balancing, and operation automation settings can be specified.

# 5.1 System environment settings

Network, load balancing, and operation automation settings are system environment settings that need to be specified. This section covers points to note when specifying network settings, settings for the different types of load balancing (hardware load balancers, and software load balancers), and the automation of (creating a job for) operations by using JP1/AJS3.

## Points to note when specifying network settings

### Important note

Disable the firewall idle communication watch functionality because the functionality might lower communication speeds between Application Server and the database when the functionality is enabled.

If the idle communication watch functionality is enabled, communication speeds might be lower than when connections reserved on the connection pool are used. This is because database connections reserved on the connection pool are disconnected by the idle communication watch functionality, and a new connection is required every time Application Server operates the database.

## Load balancing settings

- When using a hardware load balancer

  Specify the setting for request distribution to Application Server and the setting for session maintenance on the hardware load balancer. For details about how to specify the settings, see the manual of the hardware load balancer in use.

  - For configurations where the domain administration server and the runtime environment are set on different hosts

    Specify the request distribution settings only for Application Server on the created remote host. Do not specify request distribution settings for the domain administration server host.

  - For configurations where the domain administration server and the runtime environment are set on the same host

    Specify request distribution settings for the domain administration server host.

- When using a software load balancer

  Configure load balancing by creating a web server on a remote host node and creating a dependency relation between the local host and the cluster on the remote host.

## Automation of operations by using JP1/AJS3

Define the processes of starting and stopping Application Server , checking operation status, and collecting operation information and failure information as jobs, and then register them to JP1/AJS3. You also need to create a batch program to check operation status, and collect operation information and failure information.

### Related topics

- 5.2 Setting the software load balancer
- 5.3 Automating operations by using JP1/AJS3

## 5.2 Setting the software load balancer

This section describes the settings needed to use Web Server as a software load balancer for request load balancing. Create a node on a remote host (`create-node-ssh` subcommand), and then create a web server on the node (`create-webserver` subcommand). After remote host preparations are complete, create a dependency relation between the local host and the cluster created on the remote host (`create-relation` subcommand).

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been installed on the remote host for a software load balancer.
- Settings to connect to the remote host for a software load balancer are complete.

### Intended users

- System engineers

### Procedure

1. To create a node on the remote host, run the `create-node-ssh` subcommand of the `asadmin` utility command.

```
asadmin --user domain_administrative_user --passwordfile password_file_path
create-node-ssh --nodehost remote_host_IP_address
--sshuser remote_host_account_name --sshkeyfile ~/.ssh/id_rsa
--installdir installation_directory_path_for_Application_Server/
javaee_absolute_path name_of_node_to_be_created
```

When this command is run, the result is displayed as follows:

```
Command create-node-ssh executed successfully.
```

2. To create a web server on the node created on the remote host, run the `create-webserver` subcommand of the `asadmin` utility command.

```
asadmin create-webserver --node created_node_name
name_of_web_server_to_be_created
```

When this command is run, the result is displayed as follows:

```
Command create-webserver executed successfully.
```

3. To create a dependency relation, run the `create-relation` subcommand of the `asadmin` utility command.

```
asadmin create-relation --relationtype redirect
--from created_web_server_name
--to name_of_cluster_on_which_dependency_relation_was_created
--properties path=/:network-listener=http-listener-1
dependency_relation_name_to_be_created
```

When this command is run, the result is displayed as follows:

```
Command create-relation executed successfully.
```

4. When static contents are kept on the web server to increase performance, store static contents on the root document directory of the web server.

   The web server root document directory is `installation_directory_for_Application_Server/javaee/glassfish/nodes/node_name/web_server_name/root/htdocs`.

## 5.3 Automating operations by using JP1/AJS3

Define the processes of starting and stopping Application Server, checking operation status, and collecting operation information and failure information as jobs, and then register them to JP1/AJS3.

### Prerequisites

- A JP1/AJS3 environment has been prepared.

### Intended users

- System engineers

### Procedure

1. Define the process of starting Application Server as a job in JP1/AJS3.

   a. Define the commands used to start the domain administration server, performance tracer, server instance, and web server as a job.

   b. Register the created job to JP1/AJS3.

2. Define the process of stopping Application Server as a job in JP1/AJS3.

   a. Define the commands used to stop the web server, server instance, performance tracer, and domain administration server as a job.

   b. Register the created job to JP1/AJS3.

3. Define the process of checking the operation status of Application Server as a job in JP1/AJS3.

   a. Create a batch program to check the operation status.

   b. Register the created batch program as a job to JP1/AJS3.

4. Define the process of collecting operation information regarding Application Server as a job in JP1/AJS3.

   a. Create a batch program to collect operation information files.

   b. Register the created batch program as a job to JP1/AJS3.

5. Define the process of collecting failure information regarding Application Server as a job in JP1/AJS3.

   a. Create a batch program to collect failure information files.

   b. Register the created batch program as a job to JP1/AJS3.

### Related topics

- 4.6.6 Setting up batch collection of troubleshooting materials
- 7.2.1 Starting your system with a command
- 7.2.2 Stopping your system with a command
- 7.3.1 Checking the operating status of Application Server with a command
- 7.3.2 Checking the status of connection to a database server with a command
- 7.3.3 Checking the operating status of an application with a command
- 8.9 Verifying the state of use of the system
- 8.10 Checking the operating status of the system

- 9.1 Troubleshooting data output by Application Server

# 6 Settings for achieving high reliability

There are settings for achieving high reliability. This chapter explains the settings for error detection functions (process monitoring and message monitoring) and the settings for improving security (reverse proxy and SSL).

# 6.1 Settings for error detection

This section explains process monitoring and message monitoring, which are both required to detect system errors.

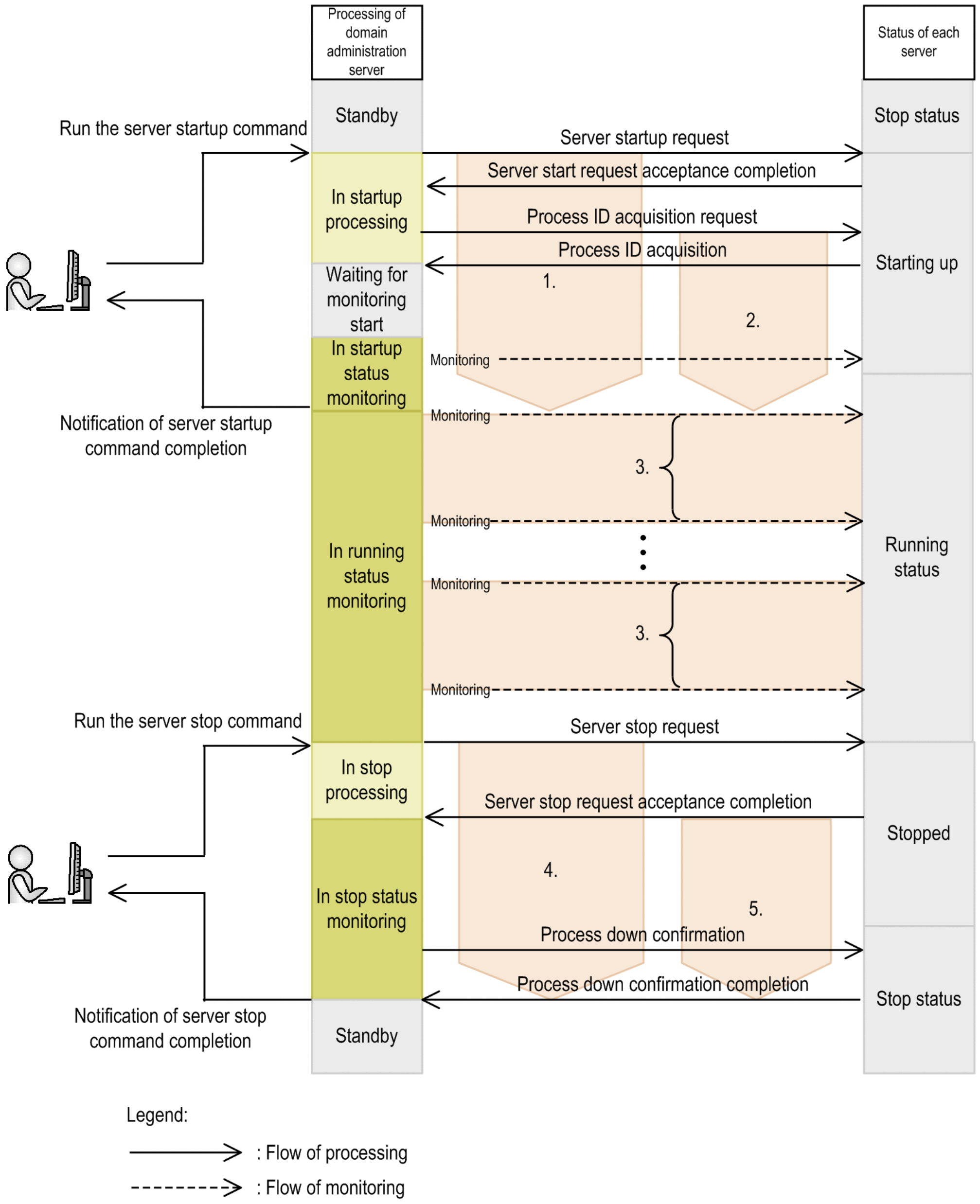
## 6.1.1 Process monitoring

For processes that are started on a domain administration server, the startup status, running status, and termination are automatically monitored. For the other processes to be monitored, a monitoring application or monitoring system needs to be used.

Quick and safe recovery from any error occurring on Application Server requires monitoring of all processes. All the Application Server processes can be monitored by using both the process monitoring function of the domain administration server and a monitoring application or monitoring system. Note that the domain administration server itself must also be monitored. Therefore, this section also describes how to do this.

### Process monitoring function of the domain administration server

When a domain administration server is running, its process monitoring functionality monitors any processes that are running on the Java EE server, web server, and performance tracer. Therefore, if you use a generic process monitoring product or monitoring system to monitor processes, monitor only the domain administration server rather than monitoring individual processes. The following figure illustrates the sections where the user can set any values for the timeouts and intervals in the flow of a process monitored by the process monitoring function of the domain administration server.


Processing of domain administration server
Status of each server
Standby
Stop status
Run the server startup command
Server startup request
In startup processing
Server start request acceptance completion
Process ID acquisition request
Process ID acquisition
Starting up
1.
2.
Waiting for monitoring start
In startup status monitoring
Monitoring
Monitoring
Notification of server startup command completion
3.
Monitoring
In running status monitoring
Running status
Monitoring
3.
Monitoring
Run the server stop command
Server stop request
In stop processing
Server stop request acceptance completion
Stopped
4.
5.
In stop status monitoring
Process down confirmation
Process down confirmation completion
Stop status
Notification of server stop command completion
Standby
Legend:
: Flow of processing
: Flow of monitoring

The following table describes the setting items in the figure. You can set any length of time for each setting item by using a parameter.

| Number in the figure | Monitored process | Description of the setting item | Parameter |
|---|---|---|---|
| 1 | • Performance tracer<br>• Web server | A startup timeout period | • Performance tracer:<br>`hitachi-prf-configs.hitachi-prf-config.`*configuration_name_of_the_performance_tracer*`.hitachi-manage-info.start-timeout-in-seconds`<br>• Web server:<br>`hitachi-webserver-configs.hitachi-webserver-config.`*configuration_n* |
| 2 | Java EE server | | |

| Number in the figure | Monitored process | Description of the setting item | Parameter |
|---|---|---|---|
| | | | *ame_of_the_web_server*.hitachi-manage-info.start-timeout-in-seconds<br>- Java EE server:<br>configs.config.*configuration_name_of_the_Java_EE_server*.hitachi-manage-info.start-timeout-in-seconds |
| 3 | - Performance tracer<br>- Web server<br>- Java EE server | The interval at which to monitor the running status of the server This section monitors server errors such as service stoppages resulting from process failures or hangs. The running status of the server process is periodically checked at the preset interval until server error monitoring ends.<br>A monitoring process is periodically generated to check the running status for each monitored process. Because this processing consumes CPU resources, increase the monitoring interval if online processing performance is affected. | - Performance tracer:<br>hitachi-prf-configs.hitachi-prf-config.*configuration_name_of_the_performance_tracer*.hitachi-manage-info.running-watch-interval-in-seconds<br>- Web server:<br>hitachi-webserver-configs.hitachi-webserver-config.*configuration_name_of_the_web_server*.hitachi-manage-info.running-watch-interval-in-seconds<br>- Java EE server:<br>configs.config.*configuration_name_of_the_Java_EE_server*.hitachi-manage-info.running-watch-interval-in-seconds |
| 4 | - Performance tracer<br>- Web server | A stop timeout period | - Performance tracer:<br>hitachi-prf-configs.hitachi-prf-config.*configuration_name_of_the_performance_tracer*.hitachi-manage-info.stop-timeout-in-seconds<br>- Web server:<br>hitachi-webserver-configs.hitachi-webserver-config.*configuration_name_of_the_web_server*.hitachi-manage-info.stop-timeout-in-seconds<br>- Java EE server:<br>configs.config.*configuration_name_of_the_Java_EE_server*.hitachi- |
| 5 | Java EE server | | |

| Number in the figure | Monitored process | Description of the setting item | Parameter |
|---|---|---|---|
| | | | `manage-info.stop-timeout-in-seconds` |

## Process monitorability

The following table shows the processes to be monitored. This table also shows whether the processes can be monitored by the domain administration server, or by generic monitoring applications or systems. Processes that cannot be monitored by the domain administration server can be monitored by other monitoring applications or monitoring systems if necessary.

| Component | Process to be monitored | | | | Domain administration server | Generic monitoring applications or systems |
|---|---|---|---|---|---|---|
| Application client | java | | | | N | Y |
| Web server | httpsd | | | | Y# | Y |
| | --- | rotatelogs2 | | | N | Y |
| | --- | rotatelogs | | | N | Y |
| | --- | httpsd | | | N | Y |
| Java EE server | java | | | | Y# | Y |
| Performance tracer | cprfd | | | | Y# | Y |
| Domain administration server | *service_name* | | | | N | Y |
| | --- | cmd | | | | Y |
| | --- | --- | java | | | Y |
| | --- | --- | --- | java | | Y |

(Legend)

Y: Can monitor the process.

N: Cannot monitor the process.

#:

If the process is directly started without using the `asadmin` utility command, the process cannot be monitored by the domain administration server.

## Monitoring the domain administration server and Java EE server

The process names of the domain administration server and Java EE server are `java`. When monitoring these processes, distinguish them from other java processes by including options in the command. Command line examples are shown below.

Example of monitoring the domain administration server process

```
ps auxww | grep "\-domainname domain1"
```

Example of monitoring the Java EE server process

```
ps auxww | grep "\-instancedir .*/localhost-domain1/JavaEE1"
```

## 6.1.2 Message monitoring

Application Server outputs a specific message to a log file when an error occurs. You can use and set up a monitoring application or monitoring system to monitor messages that are output, and to notify error occurrences when messages are output.

Messages are output to the log files shown in the following table. The table also shows keys that can be used for monitoring. Set the monitoring application or monitoring system to monitor these keys in log files.

| Log file to which messages are output | | | Key usable for monitoring |
|---|---|---|---|
| Type | Storage location | File format/file switchover type | |
| Web server error log | `installation_directory_for_Application_Server/javaee/logs/nodes/node_name/server_instance_name/error.[n]` | WRAP2#1/ new file creation type#2 | • `emerg`<br>• `alert`<br>• `crit`<br>• `error` |
| Java EE-server server instance message log | `installation_directory_for_Application_Server/javaee/logs/nodes/node_name/server_instance_name/je_message[n].log` | SEQ2#3 | • `KDKDXXXXX-E`#4<br>• `EMERGENCY`<br>• `ALERT`<br>• `SEVERE`<br>• `Transaction rolled back due to time out.` |
| Domain administration server message log | `installation_directory_for_Application_Server/javaee/logs/domains/domain_name/das_message[n].log` | SEQ2#3 | • `KDKDXXXXX-E`#4<br>• `EMERGENCY`<br>• `ALERT`<br>• `SEVERE` |

#1:

File format for the JP1/Base log file trapping function. With this format, when log data is written to the end of the last log file, logging continues with the first log file, deleting the existing data in the file.

#2:

Switchover type of monitoring-target files. The file to be monitored is switched in sequence within some files. Not all of these files exist when monitoring starts.

#3:

File format for the JP1/Base log file trapping function. With this format, a new log file is created with the same name and logging continues with the new log file after the existing log file is saved with a different name or deleted.

#4:

The *XXXXX* part indicates a unique five-digit message number managed by the program that outputs the message.

### Related topics

- 9.1 Troubleshooting data output by Application Server

## 6.2 Settings for improving security

This section explains how to set up reverse proxies and SSL for the web server in order to improve system security. Note that this setup method can be used only when the web server is directly started without using the operation management server.

### 6.2.1 Setting up reverse proxies

To improve security, set up the web server as a reverse proxy by specifying the parameters related to reverse proxies in the `httpsd.conf` and `reverse_proxy.conf` files.

#### Prerequisites

- System security requirements have been decided.

#### Intended users

- System engineers

#### Procedure

1. Install Application Server on the host to be used as a reverse proxy.
2. Specify the parameters related to reverse proxies in the `httpsd.conf` and `reverse_proxy.conf` files.
   - `httpsd.conf` file definition example:
     ```
     Include "installation_directory_for_Application_Server/httpsd/conf/
     reverse_proxy.conf"
     ```
   - `reverse_proxy.conf` file definition example:
     ```
     LoadModule proxy_module modules/mod_proxy.so
     LoadModule proxy_http_module modules/mod_proxy_http.so
     ProxyPass path_name back_end_server_URL key=value
     ProxyPassReverse path_name URL
     HWSProxyPassReverseCookie path_name
     ```

### 6.2.2 Setting up SSL

To improve security, set up the web server as an SSL accelerator. Specify the parameters related to SSL accelerators in the `httpsd.conf` file, and use the web server commands (`hwskeygen` and `hwscertutil reqgen`) to create a private key and certificate signing request (CSR) for the web server. Store the private key and certificate to complete the setup.

#### Prerequisites

- System security requirements have been decided.

#### Intended users

- System engineers

## Procedure

1. Install the web server on the host to be used as an SSL accelerator.

2. Specify the parameters related to SSL accelerators in the `httpsd.conf` file.

   - `httpsd.conf` file definition example:

     ```
     SSLEnable
     SSLCertificateFile "installation_directory_for_Application_Server/httpsd/conf/ssl/server/httpsd.pem"
     SSLCertificateKeyFile "installation_directory_for_Application_Server/httpsd/conf/ssl/server/httpsdkey.pem"
     ```

3. To create a private key for the web server, run the `hwskeygen` command .

   - `hwskeygen` command specification example

     ```
     hwskeygen -rand any_file_name -out private_key_file_name -bits private_key_bit_length
     ```

4. To create a certificate signing request (CSR), run the `hwscertutil reqgen` command.

   - `hwscertutil reqgen` command specification example

     ```
     hwscertutil reqgen -sign signature_algorithm -key private_key_file_name -out CSR_file_name
     ```

5. Request the certification authority (CA) to issue a certificate for the web server, and obtain the certificate.

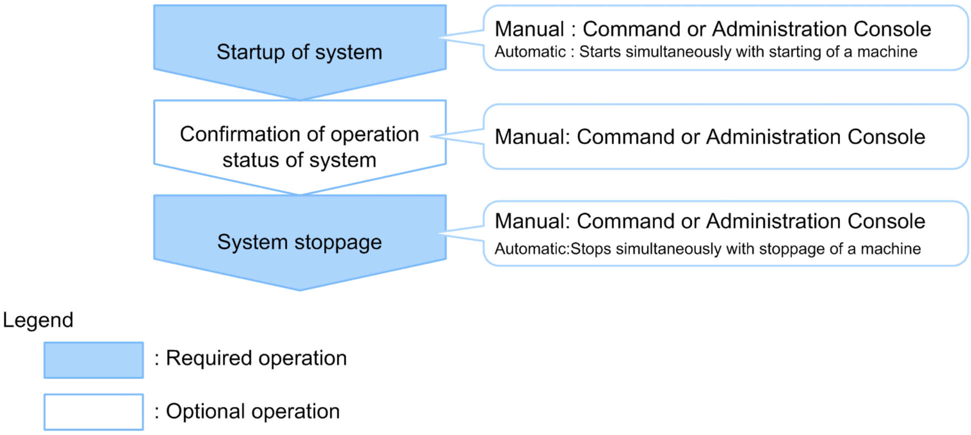
6. Store the private key and certificate in the location specified in the `httpsd.conf` file.

# 7 Tasks involved in normal operation

Tasks involved in normal operation, which you need to carry out daily, are described below. In normal operation, you use commands or Administration Console to start and stop your system and check the operating status of your system. You can start and stop your system with a machine at the same time.

# 7.1 Flow of tasks in normal operation

In normal operation, you start and stop your system as well as check the operating status of your system. To keep your system operating in a stable state, checking the operating status of your system involves checking the operating status of Application Server and applications as appropriate. To start and stop your system, and to check the operating status of your system, use a command or Administration Console. If you use commands to perform tasks for normal operations, you must make sure that you have the administrator authority. You can start and stop your system with a machine at the same time. The flow of tasks in normal operation is shown below.


Startup of system
Manual : Command or Administration Console
Automatic : Starts simultaneously with starting of a machine
Confirmation of operation status of system
Manual: Command or Administration Console
System stoppage
Manual: Command or Administration Console
Automatic:Stops simultaneously with stoppage of a machine
Legend
: Required operation
: Optional operation

## Related topics

- 7.2.1 Starting your system with a command
- 7.2.2 Stopping your system with a command
- 7.3.1 Checking the operating status of Application Server with a command
- 7.3.2 Checking the status of connection to a database server with a command
- 7.3.3 Checking the operating status of an application with a command
- 7.4.1 Logging into Administration Console
- 7.4.2 Starting your system with Administration Console
- 7.4.3 Stopping your system with Administration Console
- 7.5.1 Checking the operating status of Application Server with Administration Console
- 7.5.2 Checking the status of connection to the database server with Administration Console
- 7.5.3 Checking the operating status of applications with Administration Console
- 7.6.1 Starting your system with a machine at the same time
- 7.6.2 Stopping your system with a machine at the same time

## 7.2 Starting and stopping the system with a command

How to start or stop the system with a command is described below.

### 7.2.1 Starting your system with a command

To start your system with a command, use the `start-domain` subcommand of the `asadmin` utility command to start the domain administration server, use its `start-servers` subcommand to start Application Server, and then, use its `enable` subcommand to start application. If a hardware-based load balancer is used, open the connection to the load balancer last.

#### Prerequisites

- The domain administration server (DAS) is not running.
- Application Server is not running.
- No application is running.

#### Intended users

- System operators

#### Procedure

1. To start the domain administration server, run the `start-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin start-domain
   ```

   When this command is run, the result is displayed as follows.

   ```
   Command start-domain executed successfully.
   ```

2. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin start-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-servers executed successfully.
   ```

3. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

   ```
   performance_tracer_name running
   Command list-prfs executed successfully.
   ```

4. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

```
server_instance_name host_name port_number process_ID cluster_name running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

5. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

```
web_server_name   running
Command list-webservers executed successfully.
```

6. To start the application, run the `enable` subcommand of the `asadmin` utility command.

```
asadmin enable --target server_instance_name_or_cluster_name application_name
```

- In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
- In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

7. To display a list of applications, run the `list-applications` subcommand of the `asadmin` utility command.

```
asadmin list-applications --long=true server_instance_name_or_cluster_name
```

- In a configuration where only one Java EE server is deployed, specify the server instance name.
- In a cluster configuration where more than one Java EE server is deployed, specify the cluster name.

When this command is run, the result is displayed as follows. Ensure that the statuses of applications are `enabled`.

```
NAME                    TYPE        STATUS
application_name  <ear, web>  enabled
application_name   <web>       enabled
Command list-applications executed successfully.
```

8. For each application, specify the URL to start the application in a web browser and check that the started application can be accessed.

```
http://server_instance's_IP_address:server_instance's_port_number/
started_application_path
```

9. If you are using a hardware load balancer, unblock it.
   For more information about unblocking, refer to the hardware load balancer manual.

## 7.2.2 Stopping your system with a command

To stop your system with a command, use the `disable` subcommand of the `asadmin` utility command to stop the applications, use its `stop-servers` subcommand to stop Application Server, and then, use its `stop-domain` subcommand to stop the domain administration server. If you are using a hardware load balancer, block it first. If you are using a software load balancer, stop the web servers first.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- An application is running.

### Intended users

- System operators

### Procedure

1. If you are using a hardware load balancer, then block it.
   For more information about blocking, refer to the hardware load balancer manual.

2. If you are using a software load balancer, run the `stop-webserver` subcommand of the `asadmin` utility command with the `--graceful` option specified to perform a planned termination of the web server.

   ```
   asadmin stop-webserver --graceful true web_server_name
   ```

   A planned termination of a web server occurs after completing processing of all requests currently accepted.
   When this command is run, the result is displayed as follows:

   ```
   Command stop-webserver executed successfully.
   ```

3. If you are using a software load balancer, run the `list-webservers` subcommand of the `asadmin` utility command to display a list of web servers.

   ```
   asadmin list-webservers
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.
   *web_server_name*

   ```
      not running
   Command list-webservers executed successfully.
   ```

4. To stop the application, run the `disable` subcommand of the `asadmin` utility command.

   ```
   asadmin disable --target server_instance_name_or_cluster_name application_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

> **Important note**
>
> When an application is undeployed, its instance is released from memory, but when an application is disabled, its instance is not released. If many applications are disabled, more memory will be in use, easily leading to an out of memory error.

When this command is run, the result is displayed as follows:

```
Command disable executed successfully.
```

5. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

```
asadmin stop-servers
```

When this command is run, the result is displayed as follows:

```
Command stop-servers executed successfully.
```

6. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

```
performance_tracer_name not running
Command list-prfs executed successfully.
```

7. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

```
server_instance_name host_name port_number process_ID cluster_name not running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

8. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command. If you are using a software load balancer, you can skip this step.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

```
web_server_name   not running
Command list-webservers executed successfully.
```

9. To stop the domain administration server, run the `stop-domain` subcommand of the `asadmin` utility command.

```
asadmin stop-domain
```

When this command is run, the result is displayed as follows:

```
Command stop-domain executed successfully.
```

# 7.3 Checking the operating status of your system with a command

To keep your system operating in a stable state, check the operating status of Application Server and applications as appropriate. How to check the operating status of your system with a command is described below.

## 7.3.1 Checking the operating status of Application Server with a command

To check the operating status of Application Server with a command, execute the `list-prfs`, `list-instances`, and `list-webservers` subcommands of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.

### Intended users

- System operators

### Procedure

1. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs
```

When this command is run, the result is displayed as follows. If the server instance is running, the status is displayed as `running`, and if it is not running, the status is displayed as `not running`.

```
performance_tracer_name running
Command list-prfs executed successfully.
```

2. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. If the server instance is running, the status is displayed as `running`, and if it is not running, the status is displayed as `not running`.

```
server_instance_name host_name port_number process_ID cluster_name not running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

3. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. If the server instance is running, the status is displayed as `running`, and if it is not running, the status is displayed as `not running`.

```
web_server_name  running
Command list-webservers executed successfully.
```

## 7.3.2 Checking the status of connection to a database server with a command

To check the status of connection from a server instance to a database server, execute the `ping-connection-pool` subcommand of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- The connection from the server instance to the database server is set up.

### Intended users

- System operators

### Procedure

1. To verify whether the server instance can connect to the database server, run the `ping-connection-pool` subcommand of the `asadmin` utility command.

   ```
   asadmin ping-connection-pool --target server_instance_name connection_pool_ID
   ```

   When this command is run, the result is displayed as follows. If the server instance can connect to the database server, the command will end successfully.

   ```
   Command ping-connection-pool executed successfully.
   ```

## 7.3.3 Checking the operating status of an application with a command

To check the operating status of a application on a server instance with a command, execute the `list-applications` subcommand of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.

### Intended users

- System operators

### Procedure

1. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

```
server_instance_name host_name port_number process_ID cluster_name running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

2. To display a list of applications on the server instance, run the `list-applications` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-applications --long=true server_instance_name_or_cluster_name
```

- In a configuration where only one Java EE server is deployed, specify the server instance name.
- In a cluster configuration where more than one Java EE server is deployed, specify the cluster name.

When this command is run, the result is displayed as follows. Check `STATUS` for the operating status of the application. If the application is enabled, the operating status is displayed `enabled`, and if it is disabled, the operating status is displayed as `disabled`.

```
NAME                   TYPE        STATUS
application_name  <ear, web>  enabled
application_name  <web>       enabled
Command list-applications executed successfully.
```

# 7.4 Starting and stopping the system with Administration Console

How to log into Administration Console and how to start and stop the system with Administration Console are described below.

## 7.4.1 Logging into Administration Console

To log into Administration Console, use a web browser to start Administration Console, and enter your user ID and password.

### Prerequisites

- The domain administration server (DAS) is running.

### Intended users

- System operators

### Procedure

1. Start the web browser and enter the following URL to start Administration Console:

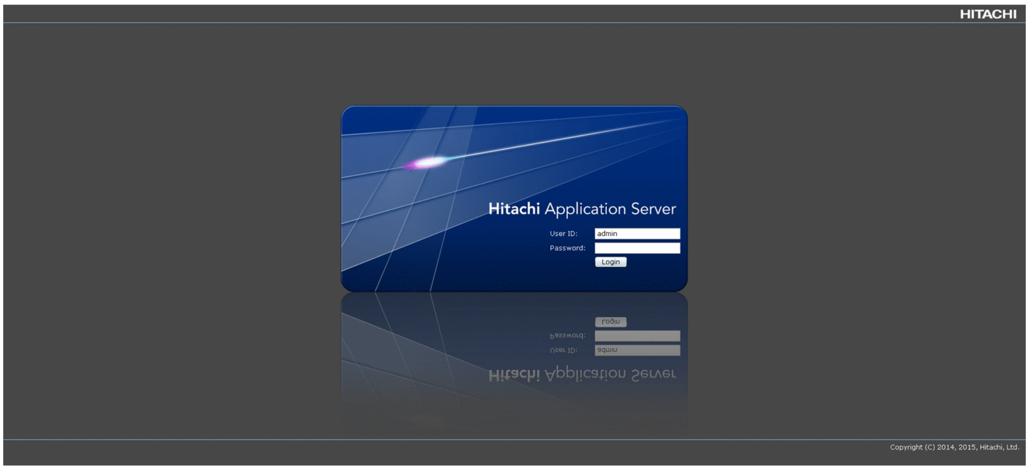
```
http://IP_address_of_the_domain_administration_server:
HTTP_port_number_of_the_domain_administration_server/admin/
```

**Reference note**

The default value of *IP_address_of_the_domain_administration_server* is `127.0.0.1`, and the default value of *HTTP_port_number_of_the_domain_administration_server* is `8080`. When both values are left as the default, you can start Administration Console from the Start menu or the start screen of the terminal.

However, if you use the built-in Administrator account in Windows Server 2012, Windows Server 2012 R2, or Windows 8, the Modern UI version of Internet Explorer is unable to start for security reasons, and an error message prompting you to sign in again using a different account might be displayed. In this case, from the **Tools** menu of Internet Explorer, select **Internet Options**. Then in the **Programs** tab, select **Always in Internet Explorer on the desktop** for **Choose how you open links**, and then restart Administration Console.

2. Input the user ID in the **User ID** text box and the password in the **Password** text box, and then click the **Login** button.



HITACHI
Hitachi Application Server
User ID: admin
Password:
Login
Copyright (C) 2014, 2015, Hitachi, Ltd.

### Reference note

The default value for the user ID is `admin`, and there is no default value for the password.

## 7.4.2 Starting your system with Administration Console

To start your system with Administration Console, use the **All Servers** pane of the **Management** tab to start Application Server, and the **All Applications** pane to start the applications.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is not running.
- No application is running.
- Administration Console is logged in.

### Intended users

- System operators

### Procedure

1. Start Application Server.

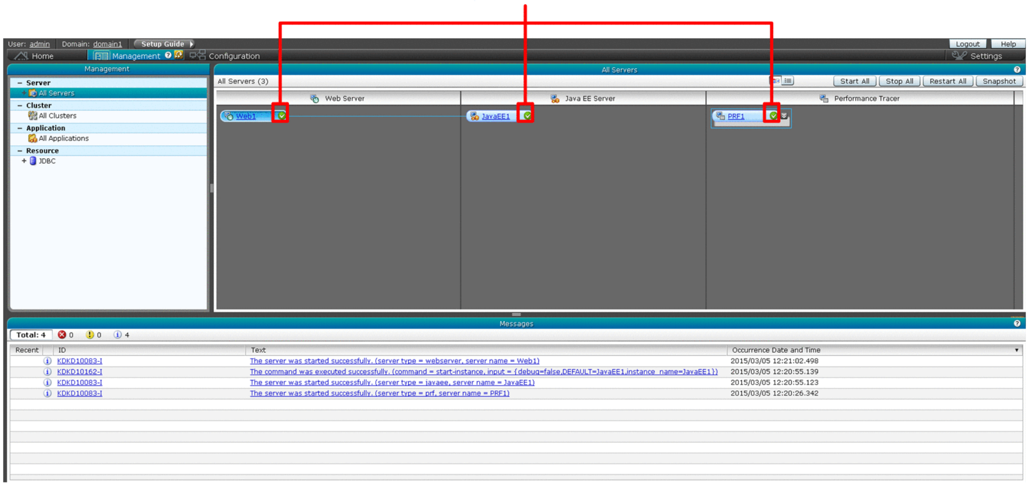
   a. Open the **Management** tab of Administration Console, and select **All Servers** from the tree in the navigation pane.

   b. On the **All Servers** pane, click the **Start All** button.

   c. On the **Start Server** dialog box, confirm the servers that will be started, and click the **OK** button.

   On the **All Servers** pane, check that all servers are running (the icon ✓ indicating running is displayed).

Confirm whether the icon ✓ which shows the operation is displayed



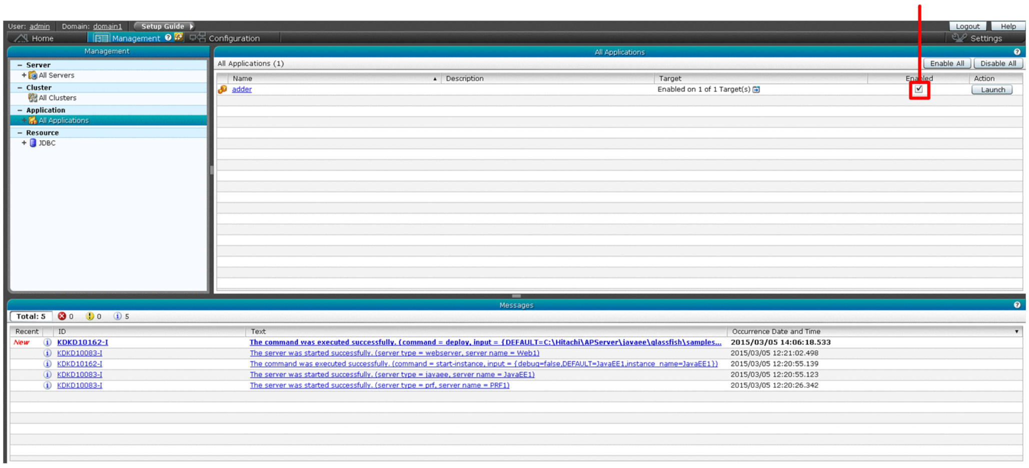
Total: 4 ✖ 0 ! 0 ⓘ 4

| Recent | ID | Text | Occurrence Date and Time |
|---|---|---|---|
| ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 12:21:02.498 |
| ⓘ | KDKD10162-I | The command was executed successfully. (command = start-instance, input = {debug=false,DEFAULT=JavaEE1,instance_name=JavaEE1}) | 2015/03/05 12:20:55.139 |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = javaee, server name = JavaEE1) | 2015/03/05 12:20:55.123 |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = prf, server name = PRF1) | 2015/03/05 12:20:26.342 |

2. Start the applications.
   a. Select **All Applications** from the tree in the navigation pane.
   b. On the **All Applications** pane, click the **Enable All** button.
   c. On the **Enable Application** dialog box, confirm the applications that will be started, and click the **OK** button.

   On the **All Applications** pane, check that all applications are running (**Enabled** is checked).

Confirm that there is a check in Enabled column



| Name | Description | Target | Enabled | Action |
|---|---|---|---|---|
| adder | | Enabled on 1 of 1 Target(s) | ☑ | Launch |

Total: 5 ✖ 0 ! 0 ⓘ 5

| Recent | ID | Text | Occurrence Date and Time |
|---|---|---|---|
| New ⓘ | **KDKD10162-I** | **The command was executed successfully. (command = deploy, input = {DEFAULT=C:\Hitachi\APServer\javaee\glassfish\samples...** | **2015/03/05 14:06:18.533** |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 12:21:02.498 |
| ⓘ | KDKD10162-I | The command was executed successfully. (command = start-instance, input = {debug=false,DEFAULT=JavaEE1,instance_name=JavaEE1}) | 2015/03/05 12:20:55.139 |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = javaee, server name = JavaEE1) | 2015/03/05 12:20:55.123 |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = prf, server name = PRF1) | 2015/03/05 12:20:26.342 |

3. For each application, specify the URL to start the application in a web browser and check that the started application can be accessed.

```
http://server_instance's_IP_address:server_instance's_port_number/
started_application_path
```

## 7.4.3 Stopping your system with Administration Console

To stop your system with Administration Console, use the **All Applications** pane of the **Management** tab to stop the applications, and **All Servers** pane to stop Application Server.

### Prerequisites

- The domain administration server (DAS) is running.

- Application Server is running.
- An application is running.
- Administration Console is logged in.

## Intended users

- System operators

## Procedure

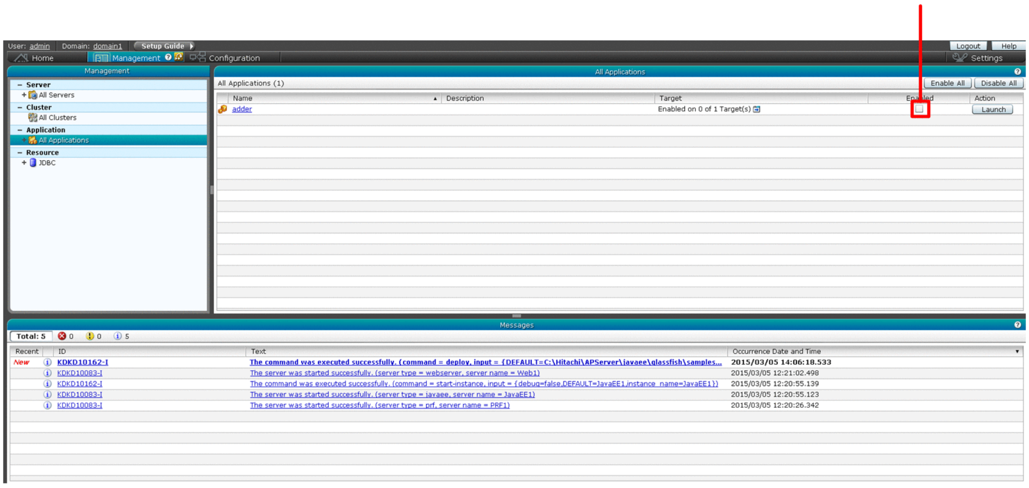
1. Stop the applications.

    a. Open the **Management** tab of Administration Console, and select **All Applications** from the tree in the navigation pane.

    b. On the **All Applications** pane, click the **Disable All** button.

    c. On the **Disable Application** dialog box, confirm the applications that will be stopped, and click the **OK** button.

    On the **All Applications** pane, check that no application is running (**Enabled** is not checked).



Confirm that there is no check in Enabled column

All Applications (1)

| | Name | Description | Target | Enabled | Action |
|---|---|---|---|---|---|
| | adder | | Enabled on 0 of 1 Target(s) | □ | Launch |

Total: 5 ⊗ 0 ⚠ 0 ⓘ 5

| Recent | | ID | Text | Occurrence Date and Time |
|---|---|---|---|---|
| New | ⓘ | KDKD10162-I | The command was executed successfully. (command = deploy, input = {DEFAULT=C:\Hitachi\APServer\javaee\glassfish\samples... | 2015/03/05 14:06:18.533 |
| | ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 12:21:02.498 |
| | ⓘ | KDKD10162-I | The command was executed successfully. (command = start-instance, input = {debug=false,DEFAULT=JavaEE1,instance_name=JavaEE1}) | 2015/03/05 12:20:55.139 |
| | ⓘ | KDKD10083-I | The server was started successfully. (server type = javaee, server name = JavaEE1) | 2015/03/05 12:20:55.123 |
| | ⓘ | KDKD10083-I | The server was started successfully. (server type = prf, server name = PRF1) | 2015/03/05 12:20:26.342 |

2. Stop Application Server.

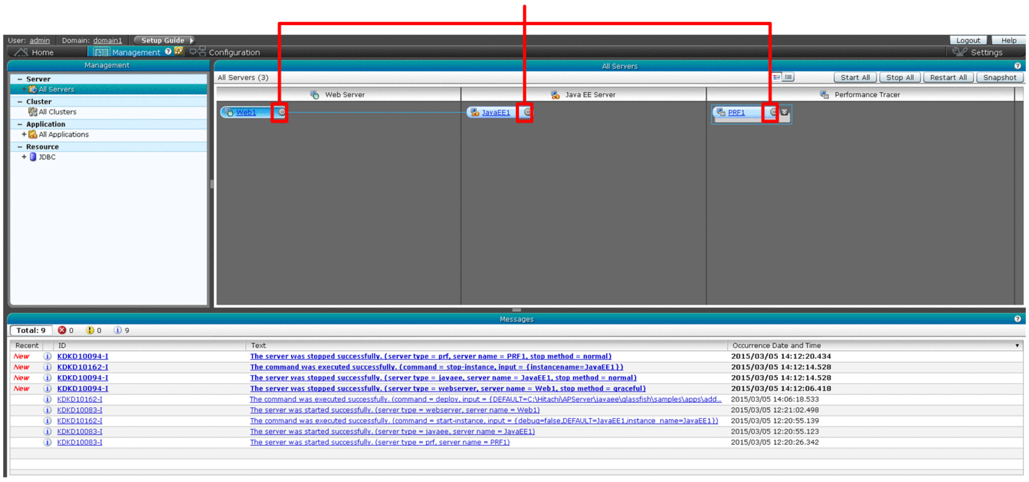
    a. Select **All Servers** from the tree in the navigation pane.

    b. On the **All Servers** pane, click the **Stop All** button.

    c. On the **Stop Server** dialog box, confirm the servers that will be stopped, and click the **OK** button.

    On the **All Servers** pane, check that all servers are not running (the icon ⏻ indicating not running is displayed).

Confirm whether the icon ⏻ which indicates stop is displayed



Total: 9 ✖ 0 ! 0 ⓘ 9

| Recent |  | ID | Text | Occurrence Date and Time |
|---|---|---|---|---|
| New | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = prf, server name = PRF1, stop method = normal) | 2015/03/05 14:12:20.434 |
| New | ⓘ | KDKD10162-I | The command was executed successfully. (command = stop-instance, input = {instancename=JavaEE1}) | 2015/03/05 14:12:14.528 |
| New | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = javaee, server name = JavaEE1, stop method = normal) | 2015/03/05 14:12:14.528 |
| New | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = webserver, server name = Web1, stop method = graceful) | 2015/03/05 14:12:06.418 |
|  | ⓘ | KDKD10162-I | The command was executed successfully. (command = deploy, input = {DEFAULT=C:\Hitachi\APServer\javaee\glassfish\samples\apps\add.. | 2015/03/05 14:06:18.533 |
|  | ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 12:21:02.498 |
|  | ⓘ | KDKD10162-I | The command was executed successfully. (command = start-instance, input = {debug=false,DEFAULT=JavaEE1,instance_name=JavaEE1}) | 2015/03/05 12:20:55.139 |
|  | ⓘ | KDKD10083-I | The server was started successfully. (server type = javaee, server name = JavaEE1) | 2015/03/05 12:20:55.123 |
|  | ⓘ | KDKD10083-I | The server was started successfully. (server type = prf, server name = PRF1) | 2015/03/05 12:20:26.342 |

# 7.5 Checking the operating status of the system with Administration Console

To keep the system operating stably, check the operating status of Application Server and applications as appropriate. How to check the operating status of the system with Administration Console is described below.

## 7.5.1 Checking the operating status of Application Server with Administration Console

To check the operating status of Application Server with Administration Console, use the **System Status** pane of the **Home** tab.

### Prerequisites

- The domain administration server (DAS) is running.
- Administration Console is logged in.

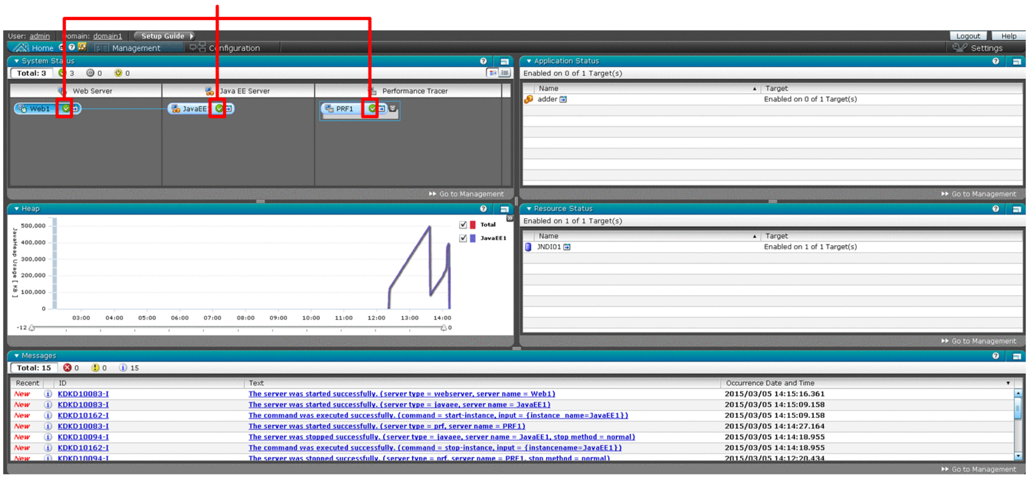
### Intended users

- System operators

### Procedure

1. Open the **Home** tab of Administration Console.

   On the **System Status** pane, check the operating status of Application Server with its icon indicating the status ( ✓ indicating running and ⏻ indicating not running).



Confirmation of icon which indicates the status

Application Status

Enabled on 0 of 1 Target(s)

| Name | Target |
|---|---|
| adder | Enabled on 0 of 1 Target(s) |

Resource Status

Enabled on 1 of 1 Target(s)

| Name | Target |
|---|---|
| JNDI01 | Enabled on 1 of 1 Target(s) |

Messages

Total: 15 ☒ 0 ! 0 ⓘ 15

| Recent | | ID | Text | Occurrence Date and Time |
|---|---|---|---|---|
| New | ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 14:15:16.361 |
| New | ⓘ | KDKD10083-I | The server was started successfully. (server type = javaee, server name = JavaEE1) | 2015/03/05 14:15:09.158 |
| New | ⓘ | KDKD10162-I | The command was executed successfully. (command = start-instance, input = {instance_name=JavaEE1}) | 2015/03/05 14:15:09.158 |
| New | ⓘ | KDKD10083-I | The server was started successfully. (server type = prf, server name = PRF1) | 2015/03/05 14:14:27.164 |
| New | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = javaee, server name = JavaEE1, stop method = normal) | 2015/03/05 14:14:18.955 |
| New | ⓘ | KDKD10162-I | The command was executed successfully. (command = stop-instance, input = {instancename=JavaEE1}) | 2015/03/05 14:14:18.955 |
| New | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = prf, server name = PRF1, stop method = normal) | 2015/03/05 14:12:20.434 |

## 7.5.2 Checking the status of connection to the database server with Administration Console

To check the status of connection from the server instance to the database server with Administration Console, use the JDBC resource **Target** pane of the **Management** tab.

## Prerequisites

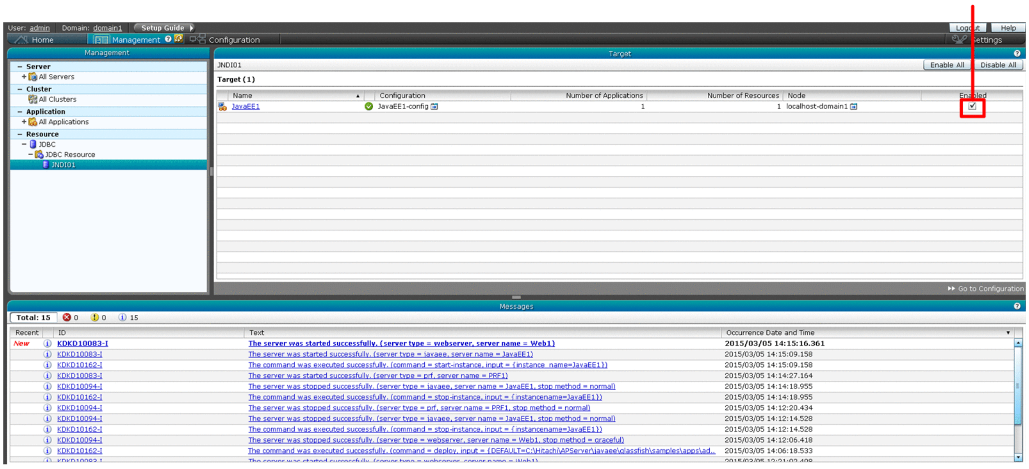
- The domain administration server (DAS) is running.
- Application Server is running.
- The connection from the server instance to the database server is set up.
- Administration Console is logged in.

## Intended users

- System operators

## Procedure

1. Open the **Management** tab of Administration Console, and click the JDBC resource name link from the tree in the navigation pane.
   On the JDBC resource **Target** pane, check whether the JDBC resource is enabled (**Enabled** is checked).



Confirm the existence of check in Enabled column

JNDI01

Target (1)

| Name | Configuration | Number of Applications | Number of Resources | Node | Enabled |
|---|---|---|---|---|---|
| JavaEE1 | JavaEE1-config | 1 | 1 | localhost-domain1 | ☑ |

Total: 15 ☒ 0 ⚠ 0 ⓘ 15

| Recent | | ID | Text | Occurrence Date and Time |
|---|---|---|---|---|
| New | ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 14:15:16.361 |
| | ⓘ | KDKD10083-I | The server was started successfully. (server type = javaee, server name = JavaEE1) | 2015/03/05 14:15:09.158 |
| | ⓘ | KDKD10162-I | The command was executed successfully. (command = start-instance, input = {instance_name=JavaEE1}) | 2015/03/05 14:15:09.158 |
| | ⓘ | KDKD10083-I | The server was started successfully. (server type = prf, server name = PRF1) | 2015/03/05 14:14:27.164 |
| | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = javaee, server name = JavaEE1, stop method = normal) | 2015/03/05 14:14:18.955 |
| | ⓘ | KDKD10162-I | The command was executed successfully. (command = stop-instance, input = {instancename=JavaEE1}) | 2015/03/05 14:14:18.955 |
| | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = prf, server name = PRF1, stop method = normal) | 2015/03/05 14:12:20.434 |
| | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = javaee, server name = JavaEE1, stop method = normal) | 2015/03/05 14:12:14.528 |
| | ⓘ | KDKD10162-I | The command was executed successfully. (command = stop-instance, input = {instancename=JavaEE1}) | 2015/03/05 14:12:14.528 |
| | ⓘ | KDKD10094-I | The server was stopped successfully. (server type = webserver, server name = Web1, stop method = graceful) | 2015/03/05 14:12:06.418 |
| | ⓘ | KDKD10162-I | The command was executed successfully. (command = deploy, input = {DEFAULT=C:\Hitachi\APServer\javaee\glassfish\samples\apps\ad.. | 2015/03/05 14:06:18.533 |
| | ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 12:21:02.408 |

## 7.5.3 Checking the operating status of applications with Administration Console

To check the operating status of applications with Administration Console, use the applications **Target** pane of the **Management** tab.

## Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.
- Administration Console is logged in.

## Intended users

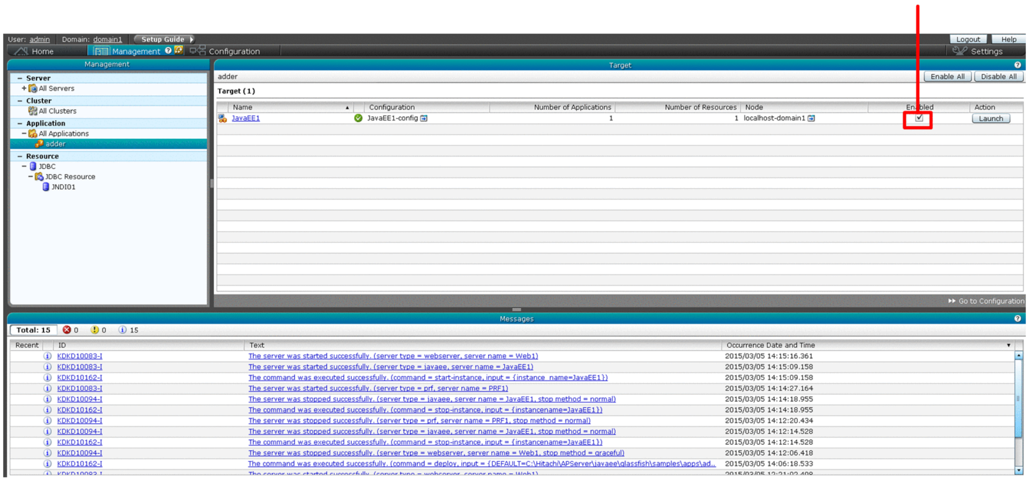
- System operators

# Procedure

1. Open the **Management** tab of Administration Console and click the application name link from the tree in the navigation pane.

   On the application **Target** pane, check whether the application is enabled (**Enabled** is checked).

Confirm the existence of check in Enabled column



adder

Target (1)

| Name | Configuration | Number of Applications | Number of Resources | Node | Enabled | Action |
|---|---|---|---|---|---|---|
| JavaEE1 | JavaEE1-config | 1 | 1 | localhost-domain1 | ☑ | Launch |

Total: 15 ⊗ 0 ! 0 ⓘ 15

| Recent | ID | Text | Occurrence Date and Time |
|---|---|---|---|
| ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 14:15:16.361 |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = javaee, server name = JavaEE1) | 2015/03/05 14:15:09.158 |
| ⓘ | KDKD10162-I | The command was executed successfully. (command = start-instance, input = {instance_name=JavaEE1}) | 2015/03/05 14:15:09.158 |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = prf, server name = PRF1) | 2015/03/05 14:14:27.164 |
| ⓘ | KDKD10094-I | The server was stopped successfully. (server type = javaee, server name = JavaEE1, stop method = normal) | 2015/03/05 14:14:18.955 |
| ⓘ | KDKD10162-I | The command was executed successfully. (command = stop-instance, input = {instancename=JavaEE1}) | 2015/03/05 14:14:18.955 |
| ⓘ | KDKD10094-I | The server was stopped successfully. (server type = prf, server name = PRF1, stop method = normal) | 2015/03/05 14:12:20.434 |
| ⓘ | KDKD10094-I | The server was stopped successfully. (server type = javaee, server name = JavaEE1, stop method = normal) | 2015/03/05 14:12:14.528 |
| ⓘ | KDKD10162-I | The command was executed successfully. (command = stop-instance, input = {instancename=JavaEE1}) | 2015/03/05 14:12:14.528 |
| ⓘ | KDKD10094-I | The server was stopped successfully. (server type = webserver, server name = Web1, stop method = graceful) | 2015/03/05 14:12:06.418 |
| ⓘ | KDKD10162-I | The command was executed successfully. (command = deploy, input = {DEFAULT=C:\Hitachi\APServer\javaee\glassfish\samples\apps\ad.. | 2015/03/05 14:06:18.533 |
| ⓘ | KDKD10083-I | The server was started successfully. (server type = webserver, server name = Web1) | 2015/03/05 13:21:02.498 |

## 7.6 Starting and stopping your system with a machine at the same time

How to start or stop your system with a machine at the same time is described below.

### 7.6.1 Starting your system with a machine at the same time

To start your system with a machine at the same time, prepare a script to specify the start processing of your system. The script will contain the `start-domain` subcommand and `start-servers` subcommand of the `asadmin` utility command.

#### Intended users

- System engineers

#### Procedure

1. Prepare a script which contains the `start-domain` subcommand and `start-servers` subcommand of the `asadmin` utility command.

2. Register the prepared script (or its symbolic link) to UNIX as a rc script.

3. Restart the machine to check whether your system is started with the machine at the same time.

4. Check the execution result.
   If your system is not started with the machine at the same time, check the contents of the script and the script registration for an error.

### 7.6.2 Stopping your system with a machine at the same time

To stop your system with a machine at the same time, prepare a script to specify the stop processing of your system. The script should contain the `stop-servers` subcommand and `stop-domain` subcommand of the `asadmin` utility command.

#### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is not running.

#### Intended user

- System engineers

#### Procedure

1. Prepare a script which contains the `stop-servers` subcommand and `stop-domain` subcommand of the `asadmin` utility command.

2. Register the prepared script (or its symbolic link) to UNIX as a rc script.

3. Shut down the machine to check whether your system is stopped with the machine at the same time.

4. Check the execution result.

   If your system is not stopped with the machine at the same time, check the contents of the script and the script registration for an error.

# 8 About the maintenance operations

This chapter describes the maintenance operations required when the running state or system configuration is changed. For such Application Server maintenance operations, do the following: change the environment definitions to match the running state or system configuration, replace applications, and back up and restore the environment information. To perform maintenance operations, use commands or Administration Console.

# 8.1 Overview of the tasks for maintenance operations

Maintenance operations are performed to maintain the system. These operations include tasks corresponding to changes in the execution status of the system or to system configuration changes. This section provides an overview of the tasks for maintenance operations, and includes the purpose of the tasks, the related work items, and methods.

## Lists of tasks for maintenance operations

The following table shows the purpose of the tasks for maintenance operations and their related work items.

| No. | Purpose of the task | Work items |
|---|---|---|
| 1 | You can change the environment definition of Application Server when:<br>• The execution status changed for a reason such as an increase in access numbers or application changes.<br>• You want to temporarily change the environment definition to handle security problems.<br>When you change an environment definition, stop the performance tracer, server instance, and web server in a batch before making any changes. After making the changes, start the servers that configure Application Server in a batch. | Use the `set` subcommand to change the Application Server settings. |
| 2 | | Use the server template to change the Web server settings. |
| 3 | | Use the `create-jvm-options` subcommand to change the Java VM options. |
| 4 | | Change the environment variables that are applied to the processing of the `asadmin` utility command. |
| 5 | You can replace the application that is running in Application Server to add a functionality to the application, or to take care of the defects. | Replace the applications. |
| 6 | If the configuration of a network is changed (for example, for regular maintenance), you can change the IP address or the host name of computer on which Application Server is installed. | Change the IP address and the host name. |
| 7 | You can back up the environment information of Application Server, when you change the environment definition and keep the setting before changing. In addition, you can restore environment information that was backed up in cases such as when you failed to change the environment information. | Back up the environment information. |
| 8 | | Restore the environment information. |
| 9 | You can apply revision patches and revised versions when revision patches are released for Application Server, or when the system undergoes regular maintenance. | Apply revision patches and revised versions. |
| 10 | You can check the usage status (such as access logs) and operational status (execution information file) of the system to revise the machine configuration and its environment definition. | Check the usage status of the system. |
| 11 | | Check the operational status of the system. |
| 12 | You can configure the same settings as another instance of Application Server by adding Application Server to a new computer (new server) when an increase in work load is required or to make preparations for an expected increase in work load. | Scale out the system. |
| 13 | You can apply an updated version of Application Server. | Upgrade Application Server. |

## Methods for performing maintenance operations

Maintenance operations can be performed by using either of the methods below. This manual gives the procedure for performing maintenance operations by using commands.

- Commands

  Follow the procedures in this manual. Execute commands by using the administrator authority.
- Administration Console

Log in to the Administration Console and perform operations by using the GUI.

## Related topics

- 8.2.1 Changing Application Server settings by using the set subcommand
- 8.2.2 Changing web server settings by using server templates
- 8.2.3 Changing Java VM options by using the create-jvm-options subcommand
- 8.2.4 Changing the environment variable to be applied to the process of the asadmin utility command
- 8.3 Replacing applications
- 8.4 Changing the IP address and host name
- 8.5 Backing up the environment information
- 8.6 Restoring environment information
- 8.7 Applying revision patches and installing revised versions
- 8.9 Verifying the state of use of the system
- 8.10 Checking the operating status of the system
- 8.11 Scaling out the system
- 8.12 Upgrading Application Server

# 8.2 Changing the environment definition of Application Server

As changes to the environment definition of Application Server, you can change Application Server settings, web server settings, and Java VM option settings.

## 8.2.1 Changing Application Server settings by using the set subcommand

To change Application Server settings, run the `set` subcommand of the `asadmin` utility command to change the setting values of Application Server.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### Procedure

1. If you are using a hardware load balancer, then block it.
   For more information about blocking, refer to the hardware load balancer manual.
   If the performance tracer, server instance, and web server are not stopped, go to step 2.
   If the performance tracer, server instance, and web server are already stopped, omit steps 1 to 5.

2. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

3. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

   ```
   performance_tracer_name not running
   Command list-prfs executed successfully.
   ```

4. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

```
server_instance_name host_name port_number process_ID cluster_name not running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

5. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

```
web_server_name not running
Command list-webservers executed successfully.
```

6. To view the settings of Application Server, run the `get` subcommand of the `asadmin` utility command.

```
asadmin get "*"
```

In the execution results of the command, verify the identifier of the setting to be changed and the value before the change.
If a value is specified for multiple parameters, the specified value is treated as follows:

For performance tracer-related parameters:

When parameter values that start with both `hitachi-prf.hitachi-prf.`*`performance_tracer_name`*`.` and `hitachi-prf-configs.hitachi-prf-config.`*`configuration_name_of_the_performance_tracer`* are set, the value of the parameter that starts with `hitachi-prf.hitachi-prf.`*`performance_tracer_name`*`.` takes effect.

For web server-related parameters

When parameter values that start with both `hitachi-webservers.hitachi-webserver.`*`web_server_name`* and `hitachi-webserver-configs.hitachi-webserver-config.`*`configuration_name_of_the_web_server`* are set, the value of the parameter that starts with `hitachi-webservers.hitachi-webserver.`*`web_server_name`* takes effect.

For a server instance-related parameter

When parameter values that start with both `servers.server.`*`Java_EE_server_name`* and `configs.config.`*`configuration_name_of_the_Java_EE_server`* are set, the value of the parameter that starts with `servers.server.`*`Java_EE_server_name`* takes effect.

7. Specify the value of the identifier to be changed, and then run the `set` subcommand of the `asadmin` utility command.

```
asadmin set target_identifier_to_be_changed=value
```

**Reference note**

When you change values other than standard properties (directives) of the web server, use the server template to change the web server settings.

When this command is run, the result is displayed as follows:

```
Command set executed successfully.
```

8. To verify the settings of Application Server after the change, run the `get` subcommand of the `asadmin` utility command.

   ```
   asadmin get "*"
   ```

   Verify that the value that was specified for the identifier by running the `set` subcommand in step 7 is reflected in the execution result of the command.

   If you are finished making changes, go to step 9.

   If you want to change the following environment definitions, define each environment setting without starting the servers that configure Application Server.

   - Web server settings
   - Java VM options
   - Environment variables to be applied to the processing of the `asadmin` utility command

9. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin start-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-servers executed successfully.
   ```

10. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

    ```
    asadmin list-prfs
    ```

    When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

    ```
    performance_tracer_name running
    Command list-prfs executed successfully.
    ```

11. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

    ```
    asadmin list-instances --long=true
    ```

    When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

    ```
    server_instance_name host_name port_number process_ID cluster_name running
    Command list-instances executed successfully.
    ```

    - *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

12. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

    ```
    asadmin list-webservers
    ```

    When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

```
web_server_name   running
Command list-webservers executed successfully.
```

13. If you are using a hardware load balancer, unblock it.

    For more information about blocking, refer to the hardware load balancer manual.

### Related topics

- 8.2.2 Changing web server settings by using server templates
- 8.2.3 Changing Java VM options by using the create-jvm-options subcommand
- 8.2.4 Changing the environment variable to be applied to the process of the asadmin utility command

## 8.2.2 Changing web server settings by using server templates

To change the web server settings (other than standard properties), use server templates, which contain the settings required to run the web server. When changing the web server settings, either set extended properties or enter a directive directly into the server template. If you edit a server template, we recommend setting the extended properties.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### Storage location and file names of the server templates

The file names of the server templates are as follows:

- `httpsd.conf@linux.vtl` (Linux)

  The server template for web server basic settings. This template contains basic settings other than those for request transmission to web server and load balancing.
- `reverse_proxy.conf@.vtl`

  The server template for the settings for web server request transmission. This template contains the settings for when the server instance of the request transmission destination is in a non-cluster configuration.
- `proxy_balancer.conf@.vtl`

  The server template for the settings for web server load balancing. This template contains the settings for when the server instance of the request transmission destination is in a cluster configuration.

When the domain is started for the first time, files in the server template are deployed to the following location: *`installation_directory_for_Application_Server`*`/javaee/glassfish/domains/`*`domain_name`*`/server_templates/webserver/conf`.

### Editing the server template

The server template can be edited as follows:

- Use VTL syntax to set the extension properties.

  Specify the VTL syntax in the server template, and then set up the web server by processing the extended property values of the `set` subcommand of the `asadmin` utility command.
- Directly specify directives.

  Set the web server by entering directives directly into the server template.

By using VTL syntax to specify extended properties, you will be able to change the web server settings by using the `set` subcommand of the `asadmin` utility command, and verify the changes by using the `get` subcommand.

When the `create-webserver` and `start-webserver` subcommands of the `asadmin` utility command are run, the server template is read by the DAS. The settings are then applied to the definition file that is read by the web server.

## Procedure

1. If you are using a hardware load balancer, then block it.

   For more information about blocking, refer to the hardware load balancer manual.

   If the performance tracer, server instance, and web server are not stopped, go to step 2.

   If the performance tracer, server instance, and web server are already stopped, omit steps 1 to 5.

2. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

3. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

   ```
   performance_tracer_name not running
   Command list-prfs executed successfully.
   ```

4. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

   ```
   server_instance_name host_name port_number process_ID cluster_name not running
   Command list-instances executed successfully.
   ```

   - *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

5. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

```
web_server_name not running
Command list-webservers executed successfully.
```

6. To set extended properties by using VTL syntax, do the following:

   The following procedure uses the `ProxyPreserveHost` directive of `reverse_proxy.conf` as an example.

   a. Using a text editor, open the server template file (`reverse_proxy.conf@.vtl`), and then specify the VTL syntax of extended properties.

   To specify extended properties, add `ex_` at the beginning of the directive name.

   For example, for the `ProxyPreserveHost` directive, extended properties can be set by using `ex_ProxyPreserveHost.value`.

   ```
   ProxyPreserveHost ${property.ex_ProxyPreserveHost.value}
   ```

   b. Save the server template file.

   c. To specify `On` for the value of the `ex_ProxyPreserveHost.value` extended property of the web server settings, run the `set` subcommand of the `asadmin` utility command.

   ```
   asadmin set
   hitachi-webservers.hitachi-webserver.web_server_name.
   property.ex_ProxyPreserveHost=On
   ```

   When this command is run, the result is displayed as follows:

   ```
   hitachi-webservers.hitachi-webserver.web_server_name.
   property.ex_ProxyPreserveHost=On
   Command set executed successfully.
   ```

   d. Run the `get` subcommand of the `asadmin` utility command to verify whether the value of the `ex_ProxyPreserveHost.value` extended property is the same as intended.

   ```
   asadmin get hitachi-webservers.hitachi-webserver.web_server_name.
   property.ex_ProxyPreserveHost
   ```

   When this command is run, the result is displayed as follows:

   ```
   hitachi-webservers.hitachi-webserver.web_server_name.
   property.ex_ProxyPreserveHost=On
   Command get executed successfully.
   ```

7. When specifying a directive directly, do the following:

   a. Open the server template file by using a text editor, and then directly specify the Web Server directive.

   To specify the directive as a comment, start the line with a hash mark (`#`) followed by a half-width space.

   b. Save the server template file.

   If you are finished making changes, go to step 8.

   If you want to change the following environment definitions, define each environment setting without starting the servers that configure Application Server.

   Application Server settings

   Java VM options

   Environment variables to be applied to the processing of the `asadmin` utility command

8. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

```
asadmin start-servers
```

When this command is run, the result is displayed as follows:

```
Command start-servers executed successfully.
```

9. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

```
performance_tracer_name running
Command list-prfs executed successfully.
```

10. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

```
server_instance_name host_name port_number process_ID cluster_name running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

11. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

```
web_server_name   running
Command list-webservers executed successfully.
```

12. If you are using a hardware load balancer, unblock it.
For more information about unblocking, refer to the hardware load balancer manual.

### Related topics

- 8.2.1 Changing Application Server settings by using the set subcommand
- 8.2.3 Changing Java VM options by using the create-jvm-options subcommand
- 8.2.4 Changing the environment variable to be applied to the process of the asadmin utility command

## 8.2.3 Changing Java VM options by using the create-jvm-options subcommand

Java VM options can be specified for a server instance and for the Domain Administration Server (DAS). To change a Java VM option, which has already been specified, use the `delete-jvm-options` subcommand of the `asadmin` utility command to delete the existing Java VM option, and then use the `create-jvm-options` subcommand to specify a new Java VM option.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server has been set up.

### Intended users

- System engineers

### Procedure

1. If you are using a hardware load balancer, then block it.
   For more information about blocking, refer to the hardware load balancer manual.
   If the performance tracer, server instance, and web server are not stopped, go to step 2.
   If the performance tracer, server instance, and web server are already stopped, omit steps 1 to 5.

2. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

```
asadmin stop-servers
```

   When this command is run, the result is displayed as follows:

```
Command stop-servers executed successfully.
```

3. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs
```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

```
performance_tracer_name not running
Command list-prfs executed successfully.
```

4. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

```
server_instance_name host_name port_number process_ID cluster_name not running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

5. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

   ```
   web_server_name not running
   Command list-webservers executed successfully.
   ```

6. To display a list of options for all server instances in the cluster, run the `list-jvm-options` subcommand of the `asadmin` utility command.

   ```
   asadmin list-jvm-options --target server_instance_name_or_cluster_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

7. When you change a Java VM option from the options listed in step 6, to delete the earlier option, run the `delete-jvm-options` subcommand of the `asadmin` utility command.

   ```
   asadmin delete-jvm-options --target server_instance_name_or_cluster_name
   [option_name[=value][:option_name[=value]]...]
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.

     In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.
   - When you specify multiple options, separate them with a colon (`:`).

   ### Important note

   If an option name or value contains symbols, the backslash (`\`) must be used as an escape character for the symbols in the name or value.

   For example, to specify `-XX:MaxMetaspaceSize=192m`, insert an escape character before the colon (`:`) by using double backslashes (`\\`), and specify the text as `-XX\\:MaxMetaspaceSize=192m`.

   When this command is run, the result is displayed as follows:

   ```
   Deleted n option(s)
   Command delete-jvm-options executed successfully.
   ```

   *n* indicates the number of options that are specified.

8. To specify a value for the Java memory, such as the Java heap for all the server instances, run the `create-jvm-options` subcommand of the `asadmin` utility command.

   ```
   asadmin create-jvm-options --target server_instance_name_or_cluster_name
   [option_name[=value][:option_name[=value]]...]
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.

     In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   **Important note**

   - When you change the value of an option that has already been specified, use the `delete-jvm-options` subcommand to delete the earlier option. If the specified option is not deleted, then a warning is displayed informing you that the same option is registered multiple times. When this warning is displayed, verify the specified option using the `list-jvm-options` subcommand and then delete the options that are not required.
   - The DAS runs SystemGC every hour and a server instance runs SystemGC every 24 hours. Use the `sun.rmi.dgc.server.gcInterval` and `sun.rmi.dgc.client.gcInterval` system properties settings to change the execution-interval of the SystemGC process. If the occurrence of the GC processes do not reduce even when you extend the occurrence interval of FullGC by changing the values of these system properties, then the Java heap might be insufficient. In this case, you might be able to improve the occurrence interval of FullGC by tuning the Java heap.

   When this command is run, the result is displayed as follows:

   ```
   Created n option(s)
   Command create-jvm-options executed successfully.
   ```

   *n* indicates the number of options that are specified.

9. To display a list of Java VM options for all server instances, run the `list-jvm-options` subcommand of the `asadmin` utility command.

   ```
   asadmin list-jvm-options --target server_instance_name_or_cluster_name
   ```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   Verify and confirm that the value of the option specified in step 8 has changed.

10. To display a list of Java VM options for the domain administration server, run the `list-jvm-options` subcommand of the `asadmin` utility command.

    ```
    asadmin list-jvm-options
    ```

11. When you change a Java VM option from the options listed in step 10, to delete the earlier option, run the `delete-jvm-options` subcommand of the `asadmin` utility command.

    ```
    asadmin delete-jvm-options [option_name[=value]
    [:option_name[=value]]...]
    ```

    When this command is run, the result is displayed as follows:

    ```
    Deleted n option(s)
    Command delete-jvm-options executed successfully.
    ```

    *n* indicates the number of options that are specified.

12. To specify a value for the Java memory, such as the Java heap for the DAS, run the `create-jvm-options` subcommand of the `asadmin` utility command.

    ```
    asadmin create-jvm-options [option_name[=value]
    [:option_name[=value]]...]
    ```

    You can specify the value for `option_name[= value]`, such as `-Xms1024m -Xmx1024m`.

    **Important note**

    Adjust the Java heap size of the DAS, based on the archive file size of the application that is deployed. Depending on the archive file size, the Java heap size of the DAS may be full and this may lead to insufficient memory.

    Additionally, when an inappropriate value (an extremely small or large value) is specified for the Java heap size of the DAS, then the DAS might not start and you might have to reconfigure the domain.

    To prevent a case where memory might be insufficient, it is recommended that you run the `backup-domain` command in advance to back up the domain.

    When this command is run, the result is displayed as follows:

    ```
    Created n option(s)
    Command create-jvm-options executed successfully.
    ```

    *n* indicates the number of options that are specified.

13. To list the options of the Java VM options for DAS, run the `list-jvm-options` subcommand of the `asadmin` utility command.

    ```
    asadmin list-jvm-options
    ```

    Verify that the value of the option specified in step 12 has changed.

14. To specify the options (except Java memory-related options) for the server instance and DAS, repeat steps 6 to 13.

    If you are finished making changes, go to step 15.

    If you want to change the following environment definitions, define each environment setting without starting the servers that configure Application Server.

    - Application Server settings
    - Web server settings
    - Environment variables to be applied to the processing of the `asadmin` utility command

15. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

```
asadmin start-servers
```

When this command is run, the result is displayed as follows:

```
Command start-servers executed successfully.
```

16. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

```
performance_tracer_name running
Command list-prfs executed successfully.
```

17. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

```
server_instance_name host_name port_number process_ID cluster_name running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

18. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

```
web_server_name   running
Command list-webservers executed successfully.
```

19. If you are using a hardware load balancer, unblock it.

For more information about unblocking, refer to the hardware load balancer manual.

## Related topics

- 8.2.1 Changing Application Server settings by using the set subcommand
- 8.2.2 Changing web server settings by using server templates
- 8.2.4 Changing the environment variable to be applied to the process of the asadmin utility command

## 8.2.4 Changing the environment variable to be applied to the process of the asadmin utility command

To change the environment variable to be applied to the process of the `asadmin` utility command, edit the environment definition file of the Java EE server (`asenv.conf`). For the environment variable to be applied to the process of the `asadmin` utility command, specify values for Java memory, such as the Java heap; the log for the `asadmin` utility command; and other information. For example, if many applications are deployed or many files are included in applications, and a memory shortage thereby occurs when the Java EE server starts, change the size of the Java heap to be applied to the process of the `asadmin` utility command.

### Prerequisites

- The domain administration server (DAS) is running.
- Application Server is running.

### Intended users

- System engineers

### Procedure

1. If you are using a hardware load balancer, then block it.
   For more information about blocking, refer to the hardware load balancer manual.
   If the performance tracer, server instance, and web server are not stopped, go to step 2.
   If the performance tracer, server instance, and web server are already stopped, omit steps 1 to 5.

2. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

3. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

   ```
   performance_tracer_name not running
   Command list-prfs executed successfully.
   ```

4. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

```
server_instance_name host_name port_number process_ID cluster_name not running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

5. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

   ```
   web_server_name not running
   Command list-webservers executed successfully.
   ```

6. Edit the environment variable definition file of Java EE Server (`asenv.conf`), and change the environment variable to be applied to the process of the `asadmin` utility command.

   Change the environment variable for the remote host and the localhost in a cluster configuration that contains more than one Java EE server.

   Environment variable definition file of Java EE Server

   *`installation_directory_for_Application_Server`*`/javaee/glassfish/config/asenv.conf`

   Example edits

   If you want to change the maximum size of the Java heap, set a value for the environment variable HJES_ASADMIN_JVM_OPTIONS.

   ```
   HJES_ASADMIN_JVM_OPTIONS=-Xmx256m
   ```

   If you are finished making changes, go to step 7.

   If you want to change the following environment definitions, define each environment setting without starting the servers that configure Application Server.

   - Application Server settings
   - Web server settings
   - Java VM options

7. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin start-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-servers executed successfully.
   ```

8. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

   ```
   performance_tracer_name running
   Command list-prfs executed successfully.
   ```

9. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

```
server_instance_name host_name port_number process_ID cluster_name running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

10. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

```
web_server_name   running
Command list-webservers executed successfully.
```

11. If you are using a hardware load balancer, unblock it.

For more information about unblocking, refer to the hardware load balancer manual.

## Related topics

- 8.2.1 Changing Application Server settings by using the set subcommand
- 8.2.2 Changing web server settings by using server templates
- 8.2.3 Changing Java VM options by using the create-jvm-options subcommand

## 8.3 Replacing applications

You can replace the applications running on Application Server to add functionality or correct a defect. To replace an application, stop the Application Servers that are running, undeploy the application to be replaced by using the `undeploy` subcommand of the `asadmin` utility, and then deploy the updated application by using the `deploy` subcommand of the `asadmin` utility. If a hardware load balancer is in used, then block it first.

### Prerequisites

- The domain administration server (DAS) is running.
- A server instance is running.
- Applications are running on the server instance.
- An application is being used.

### Intended users

- System engineers

### Procedure

1. If you are using a hardware load balancer, then block it.
   For more information about blocking, refer to the hardware load balancer manual.

2. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

   ```
   performance_tracer_name not running
   Command list-prfs executed successfully.
   ```

3. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

   ```
   server_instance_name host_name port_number process_ID cluster_name not running
   Command list-instances executed successfully.
   ```

   - *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

4. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

```
web_server_name not running
Command list-webservers executed successfully.
```

5. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

```
web_server_name not running
Command list-webservers executed successfully.
```

6. To display a list of the applications that have been deployed in order to check application names, run the `list-applications` subcommand of the `asadmin` utility command by specifying the server instance or cluster.

```
asadmin list-applications --long=true server_instance_name_or_cluster_name
```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for *server_instance_name_or_cluster_name*.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for *server_instance_name_or_cluster_name*.

   When this command is run, the result is displayed as follows. Verify the application name in the list.

```
NAME                     TYPE               STATUS
application_name    <ear, web, ejb>     enabled
application_name    <ear, web, ejb>     enabled
Command list-application executed successfully.
```

7. To undeploy applications you want to replace, run the `undeploy` subcommand of the `asadmin` utility command by specifying the application names checked in Step 6 and the server instance or cluster.

```
asadmin undeploy --target server_instance_name_or_cluster_name application_name
```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.
   - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

   When this command is run, the result is displayed as follows. Ensure that the applications to be replaced were undeployed successfully.

```
Command undeploy executed successfully.
```

8. Repeat step 7 for all applications to be replaced.

9. To deploy replaced applications to the server instance, run the `deploy` subcommand of the `asadmin` utility command by specifying the server instance or cluster.

```
asadmin deploy --target server_instance_name_or_cluster_name
file_path_of_application
```

   - In a configuration where only one Java EE server is deployed, specify the server instance name for the `--target` option.

- In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for the `--target` option.

When this command is run, the result is displayed as follows. Ensure that the application was successfully deployed.

```
Application deployed with name application_name.
Command deploy executed successfully.
```

10. Repeat step 9 for all updated applications to be replaced.

> **Reference note**
>
> In steps 6 to 10, you are replacing applications by undeploying and then deploying the applications. Alternatively, you can replace applications by redeploying applications.

11. Replace the static content stored in the `docroot` directory in the lower layer of the domain directory with the new content.

12. Replace the static content stored in the document root directory of the web server (*`installation_directory_for_Application_Server`*`/javaee/glassfish/nodes/`*`node_name`*`/`*`web_server_name`*`/root/htdocs`) with the new content.

13. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

```
asadmin list-prfs
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

```
performance_tracer_name running
Command list-prfs executed successfully.
```

14. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

```
asadmin list-instances --long=true
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

```
server_instance_name host_name port_number process_ID cluster_name running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

15. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

```
web_server_name   running
Command list-webservers executed successfully.
```

16. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

    ```
    asadmin list-webservers
    ```

    When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

    ```
    web_server_name   running
    Command list-webservers executed successfully.
    ```

17. To display a list of applications that have been deployed in the cluster in order to check the state of the replaced applications, run the `list-applications` subcommand of the `asadmin` utility command by specifying the server instance or cluster.

    To display whether an application is enabled, run the `list-applications` subcommand by specifying the `--long` option.

    ```
    asadmin list-applications --long=true server_instance_name_or_cluster_name
    ```

    - In a configuration where only one Java EE server is deployed, specify the server instance name for *server_instance_name_or_cluster_name*.
    - In a cluster configuration where more than one Java EE server is deployed, specify the cluster name for *server_instance_name_or_cluster_name*.

    When this command is run, the result is displayed as follows. The name of the application is displayed in `NAME`, and the type of application is displayed in `TYPE`. Verify that the statuses of all the applications are `enabled`.

    ```
    NAME                   TYPE         STATUS
    application_name   <ear, web>   enabled
    application_name   <web>        enabled
    Command list-applications executed successfully.
    ```

18. If you are using a hardware load balancer, unblock it so that the applications can be accessed from outside the server.

    For more information about unblocking, refer to the hardware load balancer manual.

# 8.4 Changing the IP address and host name

If the configuration of a network is changed (for example, for regular maintenance), you can change the IP address or the host name of a computer on which Application Server is installed. To change the IP address or host name, stop Application Server that is running in advance. If you change the IP address or host name of the domain administration server (DAS), you also need to stop the DAS. After that, change the IP address and host name of the target host by following the operating system (OS) procedure. In addition, update the node configuration information by using the `update-node-config` and `update-node-ssh` subcommands of the `asadmin` utility command.

## Prerequisites

- Application Server is running.
- The domain administration server (DAS) is running.

## Intended users

- System engineers

## Procedure

1. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

2. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

   ```
   performance_tracer_name not running
   Command list-prfs executed successfully.
   ```

3. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

   ```
   server_instance_name host_name port_number process_ID cluster_name not running
   Command list-instances executed successfully.
   ```

   - *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

4. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

```
web_server_name not running
Command list-webservers executed successfully.
```

5. If you change the IP address or the host name of the DAS, do the following:

   Go to step 6 when you change the IP address or host name of the remote host.

   a. To stop the DAS, run the `stop-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-domain executed successfully.
   ```

   b. Change the IP address or host name of the DAS host by following the operating system (OS) procedure.

   Restart the computer, if required.

   c. To start the DAS, run the `start-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin start-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-domain executed successfully.
   ```

   d. To display a list of nodes, run the `list-nodes` subcommand of the `asadmin` utility command.

   ```
   asadmin list-nodes
   ```

   When this command is run, the result is displayed as follows:

   ```
   node_name CONFIG IP_address_or_host name
   node_name SSH IP_address_or_host name
   Command list-nodes executed successfully.
   ```

   e. If the node and the DAS are on the same host, run the `update-node-config` subcommand of the `asadmin` utility command to update the node configuration information.

   This operation is not required if a node, such as the default node (`localhost-domain1`), is registered in the localhost, because the host name can be resolved in the localhost.

   ```
   asadmin update-node-config --nodehost changed_IP_address_or_host_name node_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command update-node-config executed successfully.
   ```

   f. If the node is on the remote host (if a node whose type is `SSH` is shown in the list of nodes that was checked in sub-step d), configure the settings for the remote host to resolve the host name of the DAS (localhost).

   Edit the `hosts` file on the remote host.

   g. If a node is configured on the remote host, edit the `das.properties` configuration file in the remote host.

   Specify the new IP address or host name for the parameter (`agent.das.host`) in the configuration file (`das.properties`). The configuration file (`das.properties`) is stored in the following location:

   *`installation_directory_for_Java_EE_Server`*`/glassfish/nodes/`*`node_name`*`/agent/config/das.properties.`

6. If you change the IP address or host name of a remote host, do the following:

   a. Change the IP address or host name of the remote host by following the operating system (OS) procedure. Restart the computer, if required.

   b. To display a list of nodes, run the `list-nodes` subcommand of the `asadmin` utility command.

   ```
   asadmin list-nodes
   ```

   When this command is run, the result is displayed as follows:

   ```
   node_name CONFIG IP_address_or_host name
   node_name SSH IP_address_or_host name
   Command list-nodes executed successfully.
   ```

   c. To update the node configuration information, run the `update-node-ssh` subcommand of the `asadmin` utility command for the remote host node (nodes whose type shown in the list of nodes that was checked in step b is `SSH`).

   ```
   asadmin --user user_name_for_domain_administration_server --passwordfile
   password_file_path
   update-node-ssh --nodehost changed_IP_address_or_host_name node_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command update-node-ssh executed successfully.
   ```

7. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin start-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-servers executed successfully.
   ```

8. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

   ```
   performance_tracer_name running
   Command list-prfs executed successfully.
   ```

9. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

   ```
   server_instance_name host_name port_number process_ID cluster_name running
   Command list-instances executed successfully.
   ```

   - *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

10. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

    ```
    asadmin list-webservers
    ```

    When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

    ```
    web_server_name   running
    Command list-webservers executed successfully.
    ```

# 8.5 Backing up the environment information

You can back up the environment information of Application Server, for example to change the environment definition and keep both the settings before and after the change. To back up environment information, create a backup file based on a file managed in the domain directory by using the `backup-domain` subcommand of the `asadmin` utility command. Additionally, copy and back up the files (mainly server templates) that are managed outside the domain directory.

## Prerequisites

- The domain administration server (DAS) is stopped.
- Application Server is stopped.

## Intended users

- System engineers or system operators

## Procedure

1. To back up the environment information of Application Server in the domain, run the `backup-domain` subcommand of the `asadmin` utility command.

```
asadmin backup-domain --backupdir
path_of_the_directory_where_the_backup_files_are_saved domain_name
```

   When this command is run, the result is displayed as follows:

```
Command backup-domain executed successfully.
```

2. Check the directory that was specified when the `backup-domain` subcommand was run, and ensure that the files were backed up.

3. Back up the files (mainly server templates) that are managed outside the domain directory by copying them to a directory used for backups.

# 8.6 Restoring environment information

To restore environment information of Application Server, restore the environment information of a backup file to the domain directory by using the `restore-domain` subcommand of the `asadmin` utility command. Additionally, copy the files that were managed outside the domain directory, such as server templates, to the original store directory.

## Prerequisites

- The domain administration server (DAS) is stopped.
- Application Server is stopped.

## Intended users

- Systems engineers or systems operators

## Procedure

1. To restore environment information of Application Server to the domain directory by using backup files, run the `restore-domain` subcommand of the `asadmin` utility command.

```
asadmin restore-domain--backupdir
path_of_the_directory_where_the_backup_files_are_saved
name_of_domain_where_environment_information_will_be_restored
```

   When this command is run, the result is displayed as follows:

```
Command restore-domain executed successfully.
```

2. Verify the domain directory of the domain that was specified when the `restore-domain` subcommand was run, and ensure that the environment information was restored.

3. Copy the files (such as server templates) that were stored in the backup directory to their original directory.

# 8.7 Applying revision patches and installing revised versions

You can install and apply revision patches and revised versions when revision patches are released for Application Server, or when the system undergoes regular maintenance. To apply a revision patch or install a revised version, stop instances of Application Server and the domain administration server (DAS) that are running, and then install the patch or revised version by using the provided media of the product.

## Prerequisites

- Application Server is running.
- The domain administration server (DAS) is running.

## Intended users

- System engineers

## Procedure

1. Obtain media for the revision patch or revised version for Application Server.

   Verify the software-support service website or the CD-ROM of the revision patch to see whether a revision patch has been released. If one has been released, get the media from either the website or the CD-ROM.

2. To stop all performance tracers, server instances, and web servers at the same time, execute the `stop-servers` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-servers
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-servers executed successfully.
   ```

3. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

   ```
   asadmin list-prfs
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `not running`.

   ```
   performance_tracer_name not running
   Command list-prfs executed successfully.
   ```

4. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

   ```
   asadmin list-instances --long=true
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `not running`.

   ```
   server_instance_name host_name port_number process_ID cluster_name not running
   Command list-instances executed successfully.
   ```

   - *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

5. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

   ```
   asadmin list-webservers
   ```

   When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `not running`.

   ```
   web_server_name not running
   Command list-webservers executed successfully.
   ```

6. To stop the DAS, run the `stop-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin stop-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command stop-domain executed successfully.
   ```

7. Install the revision patch or the revised version by using the provided media of the product.

   For more information about how to apply patches, see the `RELEASE.TXT` file attached to the revision patch.

8. In the **Completing the setup** dialog box, ensure that the revision patch or revised version is installed.

9. To start the DAS, run the `start-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin start-domain
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command start-domain executed successfully.
   ```

10. To start Application Server at once, run the `start-servers` subcommand of the `asadmin` utility command.

    ```
    asadmin start-servers
    ```

    When this command is run, the result is displayed as follows:

    ```
    Command start-servers executed successfully.
    ```

11. To display a list of performance tracers, run the `list-prfs` subcommand of the `asadmin` utility command.

    ```
    asadmin list-prfs
    ```

    When this command is run, the result is displayed as follows. Ensure that the statuses of the performance tracers are `running`.

    ```
    performance_tracer_name running
    Command list-prfs executed successfully.
    ```

12. To display a list of server instances, run the `list-instances` subcommand of the `asadmin` utility command with the `--long` option specified.

    ```
    asadmin list-instances --long=true
    ```

    When this command is run, the result is displayed as follows. Ensure that the statuses of the server instances are `running`.

```
server_instance_name host_name port_number process_ID cluster_name running
Command list-instances executed successfully.
```

- *cluster_name* is displayed only for cluster configurations that contain multiple Java EE servers.

13. To display a list of web servers, run the `list-webservers` subcommand of the `asadmin` utility command.

```
asadmin list-webservers
```

When this command is run, the result is displayed as follows. Ensure that the statuses of the web servers are `running`.

```
web_server_name   running
Command list-webservers executed successfully.
```

## Related topics

- 8.8 Error messages while applying a revision patch

# 8.8 Error messages while applying a revision patch

This section describes the meanings of and actions required for the messages that are used to report errors when a revision patch is applied in UNIX

## Error message meanings and actions required

The following table provides information about the messages, their meanings, and actions required for each message, which is displayed when a standard error occurs during the application of a revision patch.

| Message ID | Message | Meaning | Action |
|---|---|---|---|
| KESP5100-E | usage: %s [-D *log_file*] | The argument specified for the `UPDATE` option is invalid. | Verify the argument and then try again. |
| KESP5101-E | An attempt to run the pre-processing program for the update has failed (details code = %d). Read the product documentation and follow the instructions. | The pre-processing failed. | Contact the help desk. |
| KESP5102-E | An attempt to execute the post-processing program for the update has failed (details code = %d). Read the product documentation and follow the instructions. | The post-processing failed. | Contact the help desk. |
| KESP5103-E | The module replacement program has failed (details code = %d). Read the product documentation and follow the instructions. | An attempt to replace the module failed. | Contact the help desk. |
| KESP5104-E | Application of the software update failed (details code = %d). | MODPATCH failed. | Check the log file. |
| KESP5105-E | You must be a super user to run this program. | The processing failed because the user does not have root privileges. | Try again as a user with root privileges. |
| KESP5106-E | This patch is for the %s platform. The patch cannot be applied to the %s platform. | The patch was applied on an unsupported platform. | Verify the target platform and the value defined in `[!confirm]platform` in the product information file (`PRODUCT.INI`). |
| KESP5107-E | Sufficient memory could not be allocated. Please retry. | The amount of memory is insufficient. | Ensure that enough memory is available to apply the revision patch, and then perform the installation again. |
| KESP5108-E | Insufficient space in the hard disk. Increase the space in the hard disk and then retry the operation. | The amount of space on the disk is insufficient. | Ensure that sufficient disk space is available to apply the revision patch, and then perform the installation again. |
| KESP5109-E | An attempt to modify the file pplistd (%s) has failed. Please retry the operation. | An attempt to analyze information while the product installation information (`pplistd`) was being updated, failed. | Ensure that the format of the product installation information file (`pplistd`) is correct. |

| Message ID | Message | Meaning | Action |
|---|---|---|---|
| KESP5110-E | The file (%s) was not found. Retrieve the patch file again. | A file required for the update does not exist. | Ensure that the displayed file is archived in the revision patch. |
| KESP5111-E | The file pplistd was not found (%s). | The product installation information file (`pplistd`) does not exist. | Verify whether the product installation information file (`pplistd`) exists. |
| KESP5112-E | The installation path does not exist (%s). | The installation directory does not exist. | Verify whether the installation directory exists. |
| KESP5113-E | The check program for the update has failed (details code = %d). | An attempt to run the checker program, failed. | Try again in debug mode, and then verify the debug log. |
| KESP5114-E | An attempt to modify the history file has failed (%s). | An attempt to update the history file, failed. | Ensure that that the directory name and displayed file name are not the same. |
| KESP5115-E | The format of the file (%s) is invalid. | The format of the product information file (`PRODUCT.INI`) or product installation information file (`pplistd`) is invalid. | Verify the format of the product information file (`PRODUCT.INI`) or product installation information file (`pplistd`). |
| KESP5116-E | [%s] %s is too long. | A value defined in the product information file (`PRODUCT.INI`) exceeds the maximum length. | Verify the values defined in the product information file (`PRODUCT.INI`). |
| KESP5117-E | [%s] %s has not been written. | An entry required for the product information file (`PRODUCT.INI`) does not exist. | Ensure that the product information file (PRODUCT.INI) includes the necessary entries. |
| KESP5118-E | The specified platform key in the [!confirm] section is invalid. | The value defined for `platform` in the product information file (`PRODUCT.INI`) is incorrect. | Ensure that the value defined for `platform` in the product information file (`PRODUCT.INI`) is one of the following: `HPUX`, `SOLARIS`, `DIGITAL`, `LINUX`, `AIX`, `IPLINUX`, `IPFHPUX`, or `X64LIN`. |
| KESP5119-E | The specified installtype key in the [!common] section is invalid. | The value defined for `installtype` in the product information file (`PRODUCT.INI`) is invalid. | Ensure that the value of `installtype` defined in the product information file (`PRODUCT.INI`) is `HIPPINST`. |
| KESP5120-E | The version of the file (%s) is not 1.0. | The version of the file is not 1.0. | Ensure that the file version is 1.0. |
| KESP5121-E | The installtype (%s) is not "%s". | The installation type is invalid. | Ensure that the installation type is `HIPPINST`. |
| KESP5122-E | The file PRODUCT.INI is invalid. | The format of the product information file (`PRODUCT.INI`) is invalid. | Verify the file format of the product information file (`PRODUCT.INI`). |
| KESP5123-E | The file (%s) cannot be opened. | Unable to open the file. | Ensure that the displayed file is not in use. Additionally, check the amount of free disk space. |
| KESP5124-E | The %s processing has failed. | The displayed processing failed. | Check the log file. |
| KESP5125-E | The %s system call has failed. | An attempt to execute the displayed system call, failed. | Verify the log file. |
| KESP5126-E | The specified logfilemode key in the [!updater] section is invalid. | The value defined for `logfilemode` in the product information file (`PRODUCT.INI`) is invalid. | Ensure that the value of `logfilemode` defined in the product information file (`PRODUCT.INI`) is `OVERWRITE` or `APPEND`. |
| KESP5129-E | The specified unpatch key in the [!modpatch] section is invalid. | The value defined for `unpatch` in the product information file (`PRODUCT.INI`) is invalid. | Ensure that the value of `unpatch` defined in the product information file (`PRODUCT.INI`) is `SPKUTIL.TAR`. |

| Message ID | Message | Meaning | Action |
|---|---|---|---|
| KESP5130-E | The specified unpatchversion key in the [!modpatch] section is invalid. | The value defined for `unpatchversion` in the product information file (`PRODUCT.INI`) is invalid. | Ensure that the value of `unpatchversion` defined in the product information file (`PRODUCT.INI`) is `0303`. |
| KESP5131-E | [%s]%s is invalid. | The path definition defined in the product information file (`PRODUCT.INI`) is invalid. | Ensure that the path defined in the product information file (`PRODUCT.INI`) is correct. |
| KESP5132-E | The installation stopped because the version information of the selected product has been changed and is incorrect. | The version information of the installed product is incorrect. | A problem exists in the CD-ROM media. Contact the help desk. |
| KESP5133-E | The update stopped because the version information of the installed product has been changed and is incorrect. | The version information for the installed product is not correct. | A problem exists in the installed product. Contact the help desk. |
| KESP5134-E | An attempt to start %s failed. | An attempt to start the optional update check, failed. | Ensure that the optional update check DLL is the correct DLL. |
| KESP5135-E | The rollback has failed. Please rerun UPDATE. | The rollback failed. | Run the UPDATE again. |
| KESP5255-E | A fatal error has occurred. | A fatal error occurred. | Verify the log file. |

## 8.9 Verifying the state of use of the system

When revising the system configuration or environment definitions, you can verify and analyze information that provides the state of use of the system, such as work distribution and the number of requests, by verifying the access logs.

### Prerequisites

- Application Server is set up.
- Applications are running in the server instance.

### Intended users

- System engineers

### Procedure

1. View the access logs and verify the state of use of the system.
   The files to be viewed are listed below. An *x* in the file name represents a variable.
   - Work distribution (File name: `*installation_directory_for_Application Server*/javaee/logs/nodes/*node_name*/*web_server_name*/access.*x*`)
   - Number of requests per second (File name: `*installation_directory_for_Application Server*/javaee/logs/nodes/*node_name*/*web_server_name*/access.*x*`)

   Note that in the file where you can verify the number of requests per second, static content requests processed by the web servers are included in addition to the requests processed by the Java EE servers. Make a note of this point when verifying or analyzing the state of use of the system.

# 8.10 Checking the operating status of the system

To analyze the current state of the system in order to change the system configuration or environment definition, you need to verify the operating status of the system by referring to the operation information file. The operation information gathering functionality of Application Server regularly collects system operating information about the Java EE server, and then outputs this information to the operation information file. You can use this file to check and analyze information that indicates the operating status of the system.

## Prerequisites

- Application Server is set up.
- Applications are running on the server instance.

## Intended users

- System engineers

## Procedure

1. From the directory where the operation information file is output, get the files for the period to be verified.

   The default output destination directory is: `*installation_directory_for_Application_Server*/javaee/logs/nodes/*node_name*/*server_instance_name*/statistics`.

2. Check the operating information file to view the operating status of the system.

   The operation information file and items to be verified are displayed as follows. The file name includes the date and time that the file was output in the *`YYYYMMDDhhmm`* format. In addition, the numeric time zone offset from GMT (`+0900` for Japan Standard Time) is set to the *`TZ`* parameter.

   - The Java VM memory extension operation information file (File name: `JVMMemoryExtensionsStatistics_`*`server_instance_name`*`_`*`YYYYMMDDhhmmTZ`*`.csv`)

| Item to check | Attribute name | Explanation |
|---|---|---|
| `ThreadBlockedCount` | `ThreadBlockedCount.Count` | The number of threads in the block state for monitoring locks in the Java VM |
| `EdenUsedMemorySize` | `EdenUsedMemorySize.Count` | The memory size of the Eden area that is currently being used |
| `EdenTotalMemorySize` | `EdenTotalMemorySize.Count` | The guaranteed memory size of the Eden area that can be used |
| `EdenMaxMemorySize` | `EdenMaxMemorySize.Count` | The maximum memory size of the Eden area that can be used |
| `SurvivorUsedMemorySize` | `SurvivorUsedMemorySize.Count` | The memory size of the Survivor area that is currently being used |
| `SurvivorTotalMemorySize` | `SurvivorTotalMemorySize.Count` | The guaranteed memory size of the Survivor area that can be used |
| `SurvivorMaxMemorySize` | `SurvivorMaxMemorySize.Count` | The maximum memory size of the Survivor area that can be used |
| `TenuredUsedMemorySize` | `TenuredUsedMemorySize.Count` | The memory size of the Tenured area that is currently being used |
| `TenuredTotalMemorySize` | `TenuredTotalMemorySize.Count` | The guaranteed memory size of the Tenured area that can be used |

| Item to check | Attribute name | Explanation |
|---|---|---|
| TenuredMaxMemorySize | TenuredMaxMemorySize.Count | The maximum memory size of the Tenured area that can be used |
| MetaspaceUsedMemorySize | MetaspaceUsedMemorySize.Count | The memory size of the Metaspace area that is currently being used |
| MetaspaceTotalMemorySize | MetaspaceTotalMemorySize.Count | The guaranteed memory size of the Metaspace area that can be used |
| MetaspaceMaxMemorySize | MetaspaceMaxMemorySize.Count | The maximum memory size of the Metaspace area that can be used |
| ExplicitHeapSize | ExplicitHeapSize.Count | The Explicit heap size |
| ExplicitMemoryBlockCount | ExplicitMemoryBlockCount.Count | The number of Explicit memory blocks in the Explicit heap area |
| ExplicitMemoryBlockMaxSize | ExplicitMemoryBlockMaxSize.Count | The maximum size of the Explicit memory blocks |
| HttpSessionExplicitMemoryBlockMaxSize | HttpSessionExplicitMemoryBlockMaxSize.Count | The maximum size of the Explicit memory blocks acquired in the HTTP session |
| HttpSessionExplicitMemoryBlockCount | HttpSessionExplicitMemoryBlockCount.Count | The number of Explicit memory blocks acquired in the HTTP session |
| ContainerExplicitHeapSize | ContainerExplicitHeapSize.Count | The Explicit heap size managed by the container, excluding the Explicit heap area acquired in the HTTP session |
| ApplicationExplicitHeapSize | ApplicationExplicitHeapSize.Count | The Explicit heap size managed by applications and the Java VM |

- Network connection queue statistics file (file name: `NetworkConnectionQueueStatistics_`*`server_instance_name`*`_`*`YYYYMMDDhhmmTZ`*`.csv`)

| Item to check | Attribute name | Explanation |
|---|---|---|
| CountOverflows | CountOverflows.Count | The cumulative total number of excessive connections that the connection queue could not contain |
| ExCountQueued | n/a | The number of connections in the queue |
|  | ExCountQueued.Current | The current value of the connection queue |
|  | ExCountQueued.HighWaterMark | The maximum number of connections that were stored in the connection queue after statistics were initialized until statistics were collected |
|  | ExCountQueued.LowWaterMark | The minimum number of connections that were stored in the connection queue after statistics were initialized until statistics were collected |

(Legend) n/a: Not applicable

- Network thread pool statistics file (file name: `NetworkThreadPoolStatistics_`*`server_instance_name`*`_`*`YYYYMMDDhhmmTZ`*`.csv`)

| Item to check | Attribute name | Explanation |
|---|---|---|
| ExCurrentThreadsBusy | n/a | The number of request-processing threads used in the listener thread pool that processes requests |

| Item to check | Attribute name | Explanation |
|---|---|---|
| | `ExCurrentThreadsBusy.Current` | The current value of the thread being executed |
| | `ExCurrentThreadsBusy.HighWaterMark` | The maximum number of threads that were concurrently executed after statistics were initialized until statistics were collected |
| | `ExCurrentThreadsBusy.LowWaterMark` | The minimum number of threads that were concurrently executed after statistics were initialized until statistics were collected |

(Legend) n/a: Not applicable

- Web session statistics file (file name: `WebSessionStatistics_`*`server_instance_name_YYYYMMDDhhmmTZ`*`.csv`)

| Item to check | Attribute name | Explanation |
|---|---|---|
| `ActiveSessions` | n/a | The number of active sessions |
| | `ActiveSessions.Current` | The current number of sessions |
| | `ActiveSessions.HighWaterMark` | The maximum number of sessions that existed after statistics were initialized until statistics were collected |
| | `ActiveSessions.LowWaterMark` | The minimum number of sessions that existed after statistics were initialized until statistics were collected |

(Legend) n/a: Not applicable

- JDBC connection pool statistics file (file name: `JDBCConnectionPoolStatistics_`*`server_instance_name_YYYYMMDDhhmmTZ`*`.csv`)

| Item to check | Attribute name | Explanation |
|---|---|---|
| `NumConnUsed` | n/a | The total number of connections being used |
| | `NumConnUsed.Current` | The current value of the connection being used |
| | `NumConnUsed.HighWaterMark` | The maximum number of connections that were used in the pool after statistics were initialized until statistics were collected |
| | `NumConnUsed.LowWaterMark` | The minimum number of connections that were used in the pool after statistics were initialized until statistics were collected |
| `NumConnFree` | n/a | The total number of connections in the pool when the sampling was last performed |
| | `NumConnUsed.Current` | The current value of the unused connection |
| | `NumConnUsed.HighWaterMark` | The maximum number of connections that were unused in the pool after statistics were initialized until statistics were collected |
| | `NumConnUsed.LowWaterMark` | The minimum number of connections that were unused in the pool after statistics were initialized until statistics were collected |

(Legend) n/a: Not applicable

# 8.11 Scaling out the system

You can scale out a system by adding Application Server to a new computer (new server) when an increase in work load is required or to make preparations for an expected increase in work load. To scale out a system, install Application Server on the new server, specify settings to connect the localhost of the new server to the remote hosts of the existing servers, and then set up Application Server.

## Prerequisites

- In the cluster configuration, the existing server instances are included in the cluster.
- The domain administration server (DAS) is running on an existing server.
- Application Server has been set up on an existing server.
- The new server is running.

## Intended users

- System engineers

## Procedure

1. Install Application Server on the new server.

2. To resolve the host name of the domain administration server (localhost), edit the hosts file on the remote host.

3. Create a password file on the localhost.

   ```
   AS_ADMIN_SSHPASSWORD=remote_host_password
   ```

4. To create an encryption key (SSL key), run the `setup-ssh` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin --passwordfile path_of_password_file setup-ssh
   --sshuser SSH_user_name_of_remote_host remote_host_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command setup-ssh executed successfully.
   ```

5. To set an alias for the password, run the `create-password-alias` subcommand of the `asadmin` utility command on the localhost.

   ```
   asadmin create-password-alias password_alias_name
   ```

   If a password is required, enter the password of the remote host.

   When this command is run, the result is displayed as follows:

   ```
   Command create-password-alias executed successfully.
   ```

6. Edit the password file that you created in step 3.

   ```
   AS_ADMIN_SSHPASSWORD=${ALIAS=password_alias_name}
   ```

7. Restart the domain administration server.

```
asadmin restart-domain
```

When this command is run, the result is displayed as follows:

```
Command restart-domain executed successfully.
```

8. To setup Application Server on the new server, run the `create-node-ssh` subcommand of the `asadmin` utility command to add a node on the remote host.

```
asadmin --user user_name_of_domain_administration_server
--passwordfile password_file_path
create-node-ssh --nodehost remote_host_name
--sshuser account_name_of_remote_host --sshkeyfile ~/.ssh/id_rsa
--installdir installation_path_for_Application_Server/javaee_absolute_path
node_name_of_remote_host
```

For *node_name_of_remote_host*, specify a name (any name to identify the node) for the node that will be added to the remote host.

When this command is run, the result is displayed as follows:

```
Command create-node-ssh executed successfully.
```

9. To setup Application Server on the new server, run the `create-prf` subcommand of the `asadmin` utility command to build the performance tracer.

```
asadmin create-prf --node node_name_of_remote_host performance_tracer_name
```

When this command is run, the result is displayed as follows:

```
Command create-prf executed successfully.
```

10. To setup Application Server on the new server, run the subcommand `create-instance` of the `asadmin` utility command to build the Java EE server (server instance).

```
asadmin create-instance --node node_name_of_remote_host
--prf performance_tracer_name
--cluster cluster_name server_instance_name
```

Specify the name of the cluster that was built in *cluster_name*.

When this command is run, the result is displayed as follows:

```
Command create-instance executed successfully.
```

11. When a hardware load balancer is used, run the `create-webserver` subcommand of the `asadmin` utility command to build a web server on the new server.

Run the operations mentioned in step 15 when a software load balancer is used.

```
asadmin create-webserver --node node_name_of_remote_host web_server_name
```

When this command is run, the result is displayed as follows:

```
Command create-webserver executed successfully.
```

12. When a hardware load balancer is used, run the subcommand `create-relation` of the `asadmin` utility command and set the dependency relations in the new server.

    ```
    asadmin create-relation --relationtype redirect
    --from web_server_name
    --to server_instance_name
    --properties
    property_name_of_dependency_relation=property_value_of_dependency_relation
    dependency_relation_name
    ```

    If you specify settings for the redirect-related relation, you must specify `path` and `network-listener` in the `--properties` option so that static content is processed by the web server, and requests for anything other than static content are processed by the Java EE server. The following is a specification example:

    Example:

    `path=/apserver/:network-listener=http-listener-1`

    For `path`, specify the URL path, starting with a slash (`/`). Do not specify a slash only (`path=/`). In this example, a request that does not contain `apserver` as the first part of the file path in the URL following the domain name (for example, `http://xxxxxxxxxxx/index.html`) is accessed as static content of a web server. A request containing `apserver` as the first part of the file path in the URL following the domain name (for example, `http://xxxxxxxxxxx/apserver/sample/index.jsp`) is redirected to the Java EE server. In this case, the request is redirected to the URL `http://yyyyyyyyyyy/sample/index.jsp` on the Java EE server.

    For `network-listener`, specify the network listener name of an HTTP or HTTPS listener on the Java EE server at the redirection destination. On the Java EE server, the default HTTP and HTTPS listeners are defined as `http-listener-1` and `http-listener-2`, respectively, both of which are network listeners.

    When this command is run, the result is displayed as follows:

    ```
    Command create-relation executed successfully.
    ```

13. When a hardware load balancer is used and when static content are placed in the web server (for efficiency), store the static content in the document root directory of the web browser.

    The document root directory of the web server is *`installation_directory_for_Application_Server`*`/javaee/glassfish/nodes/`*`node_name`*`/`*`web_server_name`*`/root/htdocs`.

14. When a hardware load balancer is used, set the distribution of requests for the hardware load balancer.

    For details on the settings to distribute requests, refer to the manual of the hardware load balancer that is being used.

15. When a software load balancer is used, run the `create-relation` subcommand of the `asadmin` utility command to set the dependency relations in the new server.

    ```
    asadmin create-relation --relationtype redirect
    --from web_server_name
    --to cluster_name
    --properties
    property_name_of_dependency_relation=property_value_of_dependency_relation
    dependency_relation_name
    ```

    If you specify settings for the redirect-related relation, you must specify `path` and `network-listener` in the `--properties` option so that static content is processed by the web server, and requests for anything other than static content are processed by the Java EE server. The following is a specification example:

    Example:

    `path=/apserver/:network-listener=http-listener-1`

For `path`, specify the URL path, starting with a slash (`/`). Do not specify a slash only (`path=/`). In this example, a request that does not contain `apserver` as the first part of the file path in the URL following the domain name (for example, `http://xxxxxxxxxxx/index.html`) is accessed as static content of a web server. A request containing `apserver` as the first part of the file path in the URL following the domain name (for example, `http://xxxxxxxxxxx/apserver/sample/index.jsp`) is redirected to the Java EE server. In this case, the request is redirected to the URL `http://yyyyyyyyyyy/sample/index.jsp` on the Java EE server.

For `network-listener`, specify the network listener name of an HTTP or HTTPS listener on the Java EE server at the redirection destination. On the Java EE server, the default HTTP and HTTPS listeners are defined as `http-listener-1` and `http-listener-2`, respectively, both of which are network listeners.

When this command is run, the result is displayed as follows:

```
Command create-relation executed successfully.
```

## 8.12 Upgrading Application Server

To upgrade Application Server , perform an upgrade operation and apply the domain of the older version to the newer version. To do this, create a backup file of the domain of the older version by using the `backup-domain` subcommand of the `asadmin` utility command, install the newer version, and then restore the domain to the new environment by using the `restore-domain` subcommand of the `asadmin` utility command. To upgrade the configuration of the domain administration server, execute the `start-domain` subcommand of the `asadmin` utility command on the restored domain.

### Prerequisites

- The domain of the older version exists.

### Intended users

- System engineers or system operators

### Procedure

1. Back up the environment of the older version.

   a. Stop all the servers and all the domain administration servers that are running in the environment of the older version.

   b. To back up the environment information of the older version , run the `backup-domain` subcommand of the `asadmin` utility command.

   ```
   asadmin backup-domain
   --backupdir path_of_the_directory_where_the_backup_files_are_saved domain_name
   ```

   When this command is run, the result is displayed as follows:

   ```
   Command  backup-domain executed successfully.
   ```

   c. Confirm that the following backup file has been output:

   ```
   path_of_the_directory_where_the_backup_files_are_saved/domain_name/
   domain_name_YYYY_MM_DD_vserial_number.zip
   ```

   The serial number is a number starting from 00001.

2. Install a newer version.

   a. Stop all the servers and all the domain administration servers that are running in the environment of the older version.

   b. Install a newer version.

   You can use the following types of installation:

   New installation: Install a new version on a machine other than the machine on which an old version is installed

   Multiple installations: Install a new version in another directory on the machine on which an old version is installed

   Overwrite installation: Install a new version in the same directory on the machine on which an old version is installed

   If you want to build a cluster configuration in which multiple Java EE servers are placed, perform the installation on each remote host.

3. Migrate the environment definitions that were backed up in the old version to the new version.

   This process is not required for overwrite installations.

   a. Set the environment variables that were added to the environment variable definition file (`asenv.conf`) of the Java EE Server in the old version to the same file in the new version.

      Environment variable definition file of the Java EE Server

      *`installation_directory`*`/javaee/glassfish/config/asenv.conf`

      For *installation_directory*, specify the installation directory for Application Server

   b. In the environment of the new version, run the `restore-domain` subcommand of the `asadmin` utility command to restore the environment information of the old version that was backed up to the new version.

      ```
      asadmin restore-domain
      --backupdir path_of_the_directory_where_the_backup_files_are_saved
      domain_name_whose_environment_information_will_be_restored
      ```

   c. Check the domain directory of the restored domain to make sure that the environment information was restored.

4. Upgrade the domain.

   a. To upgrade the domain, run the `start-domain` subcommand of the `asadmin` utility command in the environment of the new version, with the `--upgrade` option specified.

      ```
      asadmin start-domain --upgrade domain_name
      ```

      When this command is run, the result is displayed as follows:

      ```
      Command start-domain executed successfully.
      The DAS was stopped.
      ```

      At this time, check the console to make sure that no message in the SEVERE, ALERT, or EMERGENCY level is output.

   b. In the environment of the new version, delete the directory with the name `osgi-cache-`*`number`* that is output under the domain directory.

      When you upgrade the domain, the OSGi cache directory under the domain directory is renamed to `osgi-cache-`*`number`*. Delete this directory because it will not be used after the upgrade.

   c. To start the domain administration server, run the `start-domain` subcommand of the `asadmin` utility command.

      ```
      asadmin start-domain domain_name
      ```

      When this command is run, the result is displayed as follows:

      ```
      Command start-domain executed successfully.
      ```

   d. To change the installation directory of each node to the installation directory of the newer version, run the `update-node-config` subcommand or the `update-node-ssh` subcommand of the `asadmin` utility command.

      If the node is the local host:

      ```
      asadmin update-node-config --installdir installation_directory_for_new_version/
      javaee_absolute_path --nodedir node_directory node_name
      ```

      If the node is an SSH connection node:

```
asadmin update-node-ssh --installdir installation_directory_for_new_version/
javaee_absolute_path --nodedir node_directory node_name
```

If the node directory was changed from the default in the environment for the old version, specify the `--nodedir` option to also change the node directory.

If necessary, change the node directory as well.

When this command is run, the result is displayed as follows:

If the node is the local host:

```
Command update-node-config executed successfully.
```

If the node is an SSH connection node:

```
Command update-node-ssh executed successfully.
```

5. Test the server and the applications.
   a. Start the server on the domain and test the server and the applications.
   b. If the test finishes successfully, start using the newer version as the production environment.
6. If necessary, delete the environment of the older version.

   This step is not necessary if an overwrite installation was performed.

   If the environment of the older version was in a cluster configuration in which multiple Java EE servers were placed, perform this step for each of the remote hosts.

# 9 Using troubleshooting data

In the execution and development environments for applications, logs and traces are output as troubleshooting data. The information necessary for using troubleshooting data, such as the trace acquisition points for performance tracer and how to read the output troubleshooting data, is described below.

# 9.1 Troubleshooting data output by Application Server

Troubleshooting data output by Application Server includes message logs, performance analysis traces, and troubleshooting data specific to characteristics of each process. Collect troubleshooting data by using the system information collection functionality or manually, as necessary.

## What is the system information collection function?

The system information collection function collects all the information about a system configured with Application Server. If a failure occurs, you can use this function to collect a dump from when the failure occurred and collect a set of information (such as files, logs, and traces) that is necessary to identify the causes of the failure.

## Required troubleshooting data to be collected

The table below lists troubleshooting data that is required when Application Server is installed, when a failure occurs, and while the system is running. Troubleshooting data is collected by using the system information collection functionality. If information cannot be collected using this functionality, you must collect it manually.

| No. | An error occurs when | Content of troubleshooting data | Whether data can be collected by the system information collection functionality |
|---|---|---|---|
| 1 | Application Server is installed | Installation log | -- |
| 2 | An Application Server system is being constructed, and the system is running | Log trace of each process | Y |
| 3 | | Memory dump | Y |
| 4 | | Definition and configuration information (product version information, product configuration files, etc.) | Y |
| 5 | | Work directory contents | N |
| 6 | | Screenshot | N |
| 7 | | Standard error output | N |
| 8 | | OS log (syslog) | Y |
| 9 | | OS log (event log) | N |
| 10 | | OS statistics (CPU usage ratio, memory consumption, number of threads, etc.) | N |
| 11 | | OS status information (environment variables, results of commands including `netstat`, `ps` and `sar`) | Y |

Legend:

Y: Can be collected by using the system information collection functionality.

N: Cannot be collected by using the system information collection functionality.

## Categories of troubleshooting data output by Application Server

Troubleshooting data output by Application Server when an error occurs in a system where Application Server is used is categorized as below. Collect and investigate the data if necessary.

Message log

This is log information output by Java EE servers or web servers. You can use this information to check failure causes and the operating status.

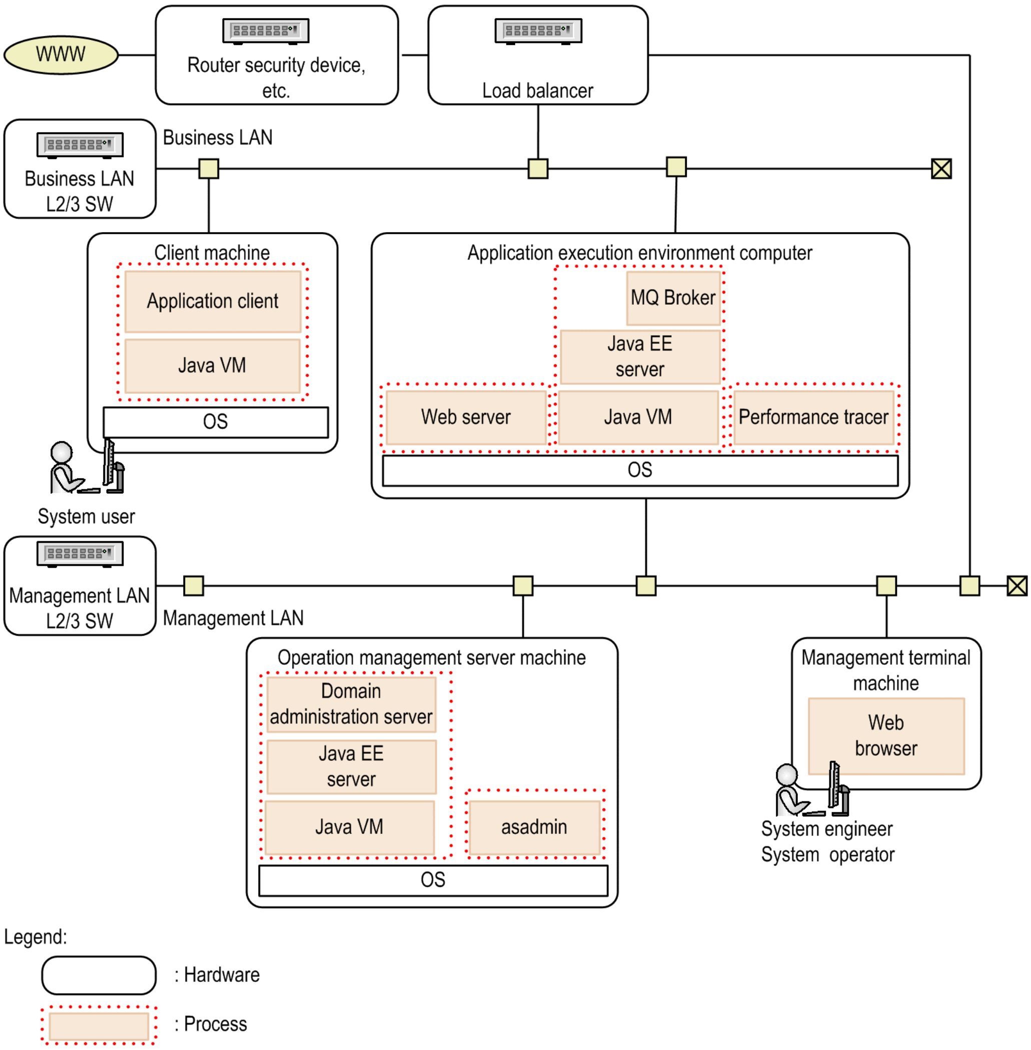
Performance analysis traces

This is trace information that can track the flow of processing requests across processes. You can use this information to analyze failures or performance.

You can track request processing by comparing performance analysis data, which has IDs output by Java EE servers, with the ID identifying a request output to the request log of a web server.

Troubleshooting data for each process

This is log data or trace information output to troubleshooting data according to the characteristics of each process. The following figure illustrates the composition of a process.



The following table lists the troubleshooting data according to process characteristics.

| No. | Process | Process characteristics and usage of data | Troubleshooting data |
|---|---|---|---|
| 1 | All processes | Run on the OS.<br>You can use information output by the OS as troubleshooting data. | • OS logs<br>• OS statistics<br>• OS status information<br>• Memory dump (core dump) |

| No. | Process | Process characteristics and usage of data | Troubleshooting data |
|---|---|---|---|
| 2 | Java processes:<br>- Java EE server<br>- Domain administration server<br>- asadmin | Run on Java VM.<br>For Java VM layers, in addition to general Java troubleshooting data, as troubleshooting data you can use Java log or thread dump information extended for the product. | - Java VM log<br>- Stack trace log<br>- Thread dump<br>- Error report file |
| 3 | Processes that handle requests:<br>- Web server<br>- Java EE server | Requests are processed across multiple processes.<br>You can check the flow of processing requests across multiple processes. | - Access log of web servers<br>- Request log of a web server<br>- Performance analysis trace |
| 4 | Java EE server | Use the explicit memory management functionality to manage session objects.<br>In the log you can check the occurrence of events of the explicit memory management functionality. | Explicit memory management functionality event log |
| 5 | | MQ Broker operates in a process.<br>In the log you can check MQ Broker messages. | MQ Broker message log |
| 6 | | Provides commands that are used during development of JSP or JAX-WS.<br>You can check error messages in the standard error output. | Standard error output |
| 7 | Java EE server (asadmin) | Starts a Java EE server.<br>You can check failure information before the Java EE server logger is initialized. | Process startup log |

## Troubleshooting data for domain administration servers

The following table lists the troubleshooting data for domain administration servers.

| No. | Log name | Output destination directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| 1 | Domain administration server message log# | *installation_directory_for_Application_Server*/javaee/logs/domains/*domain_name* | das_messagen.log | 00:00:00 | 16 megabytes | 8 | Specify the log in a parameter of the set-log-attributes subcommand of the |

| No. | Log name | Output destination directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| | | | | | | | asadmin utility command. |
| 2 | Domain administration server stack trace log# | *installation_directory_for_Application_Server*/javaee/logs/domains/*domain_name* | das_stacktrace*n*.log | 00:00:00 | 16 megabytes | 8 | Specify the log in a parameter of the set-log-attributes subcommand of the asadmin utility command. |
| 3 | Domain administration server Java VM log | *installation_directory_for_Application_Server*/javaee/logs/domains/*domain_name* | das_javavm*n*.log | 00:00:00 | 128 megabytes | 8 | Specify the log in a parameter of the create-jvm-options subcommand of the asadmin utility command. |
| 4 | Domain administration server thread dump | *installation_directory_for_Application_Server*/javaee/logs/domains/*domain_name*/config | javacore*process_id*.*date_and_time*.txt | • Output when users perform operations<br>• No limit on the size<br>• No deletion according to the number of hours or files | | | (No specification method) |
| 5 | Domain administration server error report file | *installation_directory_for_Application_Server*/javaee/logs/domains/*domain_name*/config | hs_err_pid*process_id*.log | • Output when a process terminates abnormally<br>• No limit on the size<br>• No deletion according to the number of hours or files | | | (No specification method) |

#:

Part of the stack trace data is output to a message log.

## Troubleshooting data for Java EE servers (server instances)

The following table lists troubleshooting data for Java EE servers (server instances).

| No. | Log name | Output destination directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| 1 | Server instance message log# | *installation_directory_for_Application_Server*/javaee/logs/nodes/*node_name*/*server_instance_name* | je_message*n*.log | 00:00:00 | 16 megabytes | 8 | Specify the log in a parameter of the set-log-attributes subcommand of the asadmin utility command. |
| 2 | Server instance stack trace log# | | je_stacktrace*n*.log | 00:00:00 | 16 megabytes | 8 | Specify the log in a parameter of the set-log-attributes subcommand of the asadmin utility command. |
| 3 | Server instance Java VM log | | je_javavm*n*.log | 00:00:00 | 128 megabytes | 8 | Specify the log in a parameter of the create-jvm-options subcommand of the asadmin |

| No. | Log name | Output destination directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| | | | | | | | utility command. |
| 4 | Server instance explicit memory management functionality event log | | je_eheap_event*n*.log | 00:00:00 | 128 megabytes | 8 | Specify the log in a parameter of the create-jvm-options subcommand of the asadmin utility command. |
| 5 | Server instance thread dump log | *installation_directory_for_Application_Server*/javaee/glassfish/nodes/*node_name*/*server_instance_name*/config | javacore*process_id*.*date_and_time*.txt | - Output when users perform operations<br>- No limit on the size<br>- No deletion according to the number of hours or files | | | (No specification method) |
| 6 | Server instance error report file | | hs_err_pid*process_id*.log | - Output when a process terminates abnormally<br>- No limit on the size<br>- No deletion according to the number of hours or files | | | (No specification method) |

#:

Part of the stack trace data is output to a message log.

## Troubleshooting data of Java EE servers (asadmin)

The following table lists the troubleshooting data for Java EE servers (asadmin).

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| 1 | asadmin message log# | *installation_directory_for_Application_ Server*/javaee/logs/commands/asadmin | asadmin_message*n*.log | 00:00:00 | 16 megabytes | 8 | Specify the log in the asenv environment variable. |
| 2 | asadmin stack trace log# | | asadmin_stacktrace*n*.log | 00:00:00 | 16 megabytes | 8 | Specify the log |

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| | | | | | | | in the `asenv` environment variable. |
| 3 | asadmin thread dump | *current_directory_when_command_was_executed* | `javacore`*`process_id.date_and_time`*`.txt` | • Output when users perform operations<br>• No limit on the size<br>• No deletion according to the number of hours or files | | | (No specification method) |
| 4 | asadmin error report file | | `hs_err_pid`*`process_id`*`.log` | • Output when a process terminates abnormally.<br>• No limit on the size<br>• No deletion according to the number of hours or files | | | (No specification method) |
| 5 | Log collected when the asadmin process was started | *`installation_directory_for_Application_ Server`*`/javaee/logs/commands/asadmin` | `asadmin_launch`*`n`*`.log` | 00:00:00 | 16 megabytes | 8 | Specify the log in the `asenv` environment variable. |

#:

Part of stack trace data is output to a message log.

## Troubleshooting data for Java EE servers (application clients)

The following table lists the troubleshooting data for Java EE servers (application clients).

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| 1 | Application client thread dump log | *current_directory* (the directory where the `appclient` command started) | `javacore`*`process_id.date_and_time`*`.txt` | • Output when users perform operations<br>• No limit on the size<br>• No deletion according to the number of hours or files. | | | (No specification method) |
| 2 | Application client error report file | | `hs_err_pid`*`process_id`*`.log` | • Output when a process terminates abnormally.<br>• No limit on the size<br>• No deletion according to the number of hours or files. | | | (No specification method) |

## Troubleshooting data for Java EE servers (MQ Broker)

The following table lists the troubleshooting data for Java EE servers (MQ Broker).

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| 1 | MQ Broker message log | *installation_directory_for_Application_Server*/javaee/nodes/*node_name*/*server_instance_name*/imq/instances/*MQ_instance_name*/log | log_*n*#.txt | 00:00:00 | 16 megabytes | 8 | Specify the log in the config.properties file. |

#:

*n* = 0 to *number_of_files* - 1

## Troubleshooting data for web servers

The following table lists the troubleshooting data for web servers.

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| 1 | Access log | *installation_directory_for_Application_Server*/javaee/logs/nodes/*node_name*/*web_server_name* | access.*n*#1 | 24 hours | 2 gigabytes (max.) | 8 | Specify the log in the TransferLog directive or the CustomLog directive. |
| 2 | Request log | | hwsrequest.*n*#1 | 24 hours | 2 gigabytes (max.) | 8 | Specify the log in the HWSRequestLog directive. |
| 3 | Error log | | error.*n*#1 | 24 hours | 2 megabytes (max.) | 8 | Specify the log in the ErrorLog directive. |
| 4 | Process ID log | | httpd.pid | -- | -- | -- | Specify the log in the PidFi |

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | How to specify the log |
|---|---|---|---|---|---|---|---|
| | | | | | | | le directive. |
| 5 | Internal trace | | hws.trclog.*n*#2 | -- | -- | 5 | Specify the log in the HWSTraceLogFile directive. |
| 6 | Shared memory ID log | | hws.trcid | -- | -- | -- | Specify the log in the HWSTraceIdFile directive. |

Legend:

--: Not applicable

#1:

*n* = *time_when_log_collection_started*

#2:

*n* = 01 to 05

## Troubleshooting data for performance tracer

The following table lists the troubleshooting data for performance tracer.

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | (No specification method) |
|---|---|---|---|---|---|---|---|
| 1 | Message log | $PRFSPOOL/log/*PRF_identifier* | prf_message*n*# | 00:00:00 | 10 megabytes | 8 | Follow the settings on the PRF identifier and the PRFSPOOL environment variable specified for the |

| No. | Log name | Output directory | File name | Wrap time (initial value) | Size (initial value) | Number of files (initial value) | (No specification method) |
|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  | process to output the log. |

#:

*n* = 01 to 32

# 9.2 Output formats of the log files

The rotation methods and the output formats of the log files that are output from each process are described below.

## 9.2.1 Log file rotation methods

The log data for each process is output to multiple files in rotation. The files to which log data is to be output are switched at a specified time or when the size of the output-destination file reaches a specified size. The log rotation methods are described below.

### Log rotation method for a Java EE server

If the log file to which log data is output is named `prf_message01`, when the first switch occurs, this file is renamed `prf_message02`. A new log file named `prf_message01` is created, and then log data is output to the newly created file. When the second switch occurs, `prf_message01` is renamed `prf_message02`, and `prf_message02` is renamed `prf_message03`. A new log file named `prf_message01` is created, and then log data is output to the newly created file. When the number of log files exceeds a specified number, a file that has the largest number is deleted.

### Log rotation method for a web server

You can specify how to split log files by using the rotatelogs or rotatelogs2 program. Specify the split method in the definition file of the web server.

rotatelogs program

The access log or error log can be output to multiple files by splitting the data at a specified time interval (for example, every 24 hours).

- Log split time interval

  Specify the amount of time for which data is to be collected to a single log file. After the specified time elapses, log data will be collected to a new file.

- Number of files

  Specify the number of files among which the output log data is to be split. When the number of files exceeds the specified number, the oldest file will be deleted.

rotatelogs2 program

The access log or error log can be split based on the size of the log file and then output to multiple files by using the wraparound method.

- Log file size

  Specify the maximum size (in kilobytes) of a log file. If the size of a file exceeds the maximum size when log data is output, the next file in the rotation is cleared, and then the log data is output to that file.

- Number of log files

  Specify the maximum number of log files to which data is to be output. When the size of a log file exceeds the maximum size, the log file is switched. In such a case, if the extension of the log file to which data was last output is equal to the maximum number of log files, when the log file is switched, data will be output to the first log file.

  Example: If `3` is specified for the number of log files

  If a log file is `errorlog.001`, log data is output sequentially from `errorlog.001` to `errorlog.003`. When the size of `errorlog.003` exceeds the maximum file size, is cleared, and then the remaining log data is output to `errorlog.001`.

### Log rotation method for performance tracer

The most recent file is always named `prf_message01`. When the first switch occurs, `prf_message01` is renamed `prf_message02`. When the second switch occurs, `prf_message01` is renamed `prf_message03`. (Note that `prf_message02` is not renamed `prf_message03`.)

### Log rotation method for the application development environment

Based on the file size, the log file to which log data is to be output is renamed.

## 9.2.2 Output formats of Java EE server logs

This section describes the output format of logs provided by Java EE server, and the output items.

### Logs provided by Java EE server

Java EE server provides the following types of logs:

- Message log
- Stack trace log
- Java VM log
- Event log of the explicit memory management functionality
- Thread dump log
- Process start log

The following describes the output format and of each log, and the output items:

### Message log

Output format

```
number date time application_name pid tid message_ID text
```

Output items

| Output items | Description |
|---|---|
| *number* | Trace code serial number (4 digits) |
| *date* | Trace collection date. Output in *yyyy/mm/dd* format. |
| *time* | Trace collection time. Output in the *hh:mm:ss.sss* format. |
| *application_name* | Application identification name.<br>Name that identifies an application. |
| *pid* | Process ID |
| *tid* | Thread ID.<br>ID for identifying a thread. |
| *message_ID* | Message ID.<br>ID for distinguishing messages. |
| *text* | Message text. |

## Stack trace log

Output format

This is the same as the message log.

Output items

This is the same as the message log.

## Java VM log

Extended verbosegc output functionality (when G1GC is not used)

Output format

```
[id] date (Skip Full:full_count, Copy:copy_count)
[gc_kind gc_info, gc_time secs][Eden:  eden_info]
[Survivor:  survivor_info]
[Tenured:  tenured_info][Metaspace:  metaspace_info]
[class space:  class_space_info] [cause:cause_info]
[User:  user_cpu secs]
[Sys:  system_cpu secs]
[IM:  jvm_alloc_size, mmap_total_size, malloc_total_size]
[TC:  thread_count]
[DOE:  doe_alloc_size, called_count]
[CCI:  cc_used_sizeK, cc_max_sizeK, cc_infoK]
```

Note:

There is no linefeed.

Output items

| Output items | Description |
|---|---|
| *id* | Identifier of the Java VM log file |
| *date* | Date and time when GC started. This item is not output if the `-XX:-HitachiVerboseGCPrintDate` option is specified |
| *full_count* | Number of times Full GC was skipped. This item is output if the `-XX:HitachiVerboseGCIntervalTime` option is specified. |
| *copy_count* | Number of times CopyGC was skipped. This item is output if the `-XX:HitachiVerboseGCIntervalTime` option is specified. |
| *gc_kind* | GC type (`Full GC` or `GC`) |
| *gc_info* | GC information (*`area_length_before_GC -> area-length-after-GC(area_size)`*<br>Example: `264K->0K(512K)` |
| *gc_time* | Elapsed time (in seconds) for GC |
| *Eden* | Type of the Eden area (`DefNew::Eden`). |
| *eden_info* | Memory information of the Eden area |
| *Survivor* | Type of the Survivor area (`DefNew::Survivor`). |
| *survivor_info* | Memory information of the Survivor area |
| *Tenured* | Type of the Tenured area (`Tenured`). |
| *tenured_info* | Memory information of the Tenured area |
| *metaspace_info* | Memory information (in kilobytes) of the Metaspace area |

| Output items | Description |
|---|---|
| *classspace_info* | CompressedClassSpace information (in kilobytes) |
| *cause_info* | Cause of GC |
| *user_cpu* | CPU time (in seconds) that the GC thread consumed in the user mode |
| *system_cpu* | CPU time (in seconds) that the GC thread consumed in the kernel mode |
| *jvm_alloc_size* | Size of the area currently in use, from among the areas being managed in the Java VM |
| *mmap_total_size* | Total size of the C heap area allocated for mmap (VirtualAlloc in Windows), from among the areas being managed in the Java VM. This item is output if the `-XX:+HitachiVerboseGCPrintJVMInternalMemory` option is specified. |
| *malloc_total_size* | Total size of the C heap area allocated for malloc, from among the areas being managed in the Java VM. This item is output if the `-XX:+HitachiVerboseGCPrintJVMInternalMemory` option is specified. |
| *thread_count* | The number of Java threads. This item is output if the `-XX:+HitachiVerboseGCPrintThreadCount` option is specified. |
| *doe_alloc_size* | Cumulative heap size allocated by invoking `java.io.File.deleteOnExit()`. This item is output if the `-XX:+HitachiVerboseGCPrintDeleteOnExit` option is specified. |
| *called_count* | Invocation count of `java.io.File.deleteOnExit()`. This item is output if the `-XX:+HitachiVerboseGCPrintDeleteOnExit` option is specified. |
| *cc_used_size* | Code cache area (in kilobytes) used when GC occurs. This item is output if the `-XX:+PrintCodeCacheInfo` option is specified. |
| *cc_max_size* | Maximum size (in kilobytes) of the code cache area. This item is output if the `-XX:+PrintCodeCacheInfo` option is specified. |
| *cc_info* | Maintenance information. This item is output if the `-XX:+PrintCodeCacheInfo` option is specified. |

## Extended verbosegc output functionality (when G1GC is used)

### Output format

```
[id]date[gc_kind gc_info, gc_time secs][Status:gc_status]
[G1GC::Eden:  eden_info][G1GC::Survivor:  survivor_info]
[G1GC::Tenured:  tenured_info][G1GC::Humongous:  humongous_info]
[G1GC::Free:  free_info][Metaspace:  metaspace_info]
[class space:  class_space_info] [cause:cause_info][RegionSize:  region_sizeK]
[Target:  target_time secs][Predicted:  predicted_time secs]
[TargetTenured:  target_sizeK][Reclaimable:  reclaimable_info]
[User:  user_cpu secs][Sys:  system_cpu secs]
[IM:  jvm_alloc_sizeK, mmap_total_sizeK, malloc_total_sizeK]
[TC:  thread_count][DOE:  doe_alloc_sizeK, called_count]
[CCI:  cc_used_sizeK, cc_max_sizeK, cc_infoK]
```

Note:

There is no linefeed.

### Output items

| Output items | Description |
|---|---|
| *id* | Identifier of the Java VM log file |
| *date* | Date and time when GC or CM started |
| *gc_kind* | Type of GC or CM. |

| Output items | Description |
|---|---|
| | `Full GC`, `Mixed GC`, `Young GC`, `Young GC(initial-mark)`, `CM Remark`, or `CM Cleanup` is output. |
| *gc_info* | Memory information (in kilobytes) of the Java heap area |
| *gc_time* | Elapsed time (in seconds) for GC |
| *gc_status* | GC status (`to exhausted` or `-`).<br>If *gc_kind* is `Young GC`, `Young GC(initial-mark)`, or `Mixed GC` - or `to exhausted` is output.<br>If *gc_kind* is other than `Young GC`, `Young GC(initial-mark)`, or `Mixed GC` - is output. |
| *eden_info* | Memory information (in kilobytes) of the Eden area |
| *survivor_info* | Memory information (in kilobytes) of the Survivor area |
| *tenured_info* | Memory information (in kilobytes) of the Tenured area |
| *humongous_info* | Memory information (in kilobytes) of the Humongous area |
| *free_info* | Memory information (in kilobytes) of the Free area |
| *metaspace_info* | Memory information (in kilobytes) of the Metaspace area |
| *classspace_info* | CompressedClassSpace information (in kilobytes) |
| *cause_info* | Cause of GC |
| *region_size* | Size of a region (in kilobytes) |
| *target_time sec* | Target time during which the application is stopped by GC (in seconds) |
| *predicted_time sec* | Duration (in seconds) that the Java VM predicted the application would be stopped by GC |
| *target_size* | Size of the GC-target Tenured area in Mixed GC |
| *reclaimable_info* | Predicted collection size information (in kilobytes) |
| *user_cpu* | CPU time (in seconds) that the GC thread consumed in the user mode. This item is not output if the `-XX:-HitachiVerboseGCCpuTime` option is specified.<br>If an attempt to obtain the CPU time failed, `unknown` is displayed. |
| *system_cpu* | CPU time (in seconds) that the GC thread consumed in the kernel mode. This item is not output if the `-XX:-HitachiVerboseGCCpuTime` option is specified.<br>If an attempt to obtain the CPU time failed, `unknown` is displayed. |

CSV format output (when G1GC is not used)

Output format

```
id,date,full_count,copy_count,gc_kind,gc_info,gc_time,eden_info,
survivor_info,tenured_info,metaspace_info,classspace_info,cause_info,user_cpu,
system_cpu,jvm_alloc_size,mmap_total_size,malloc_total_size,thread_count,
doe_alloc_size,called_count,cc_used_size,cc_max_size,cc_info
```

Note:

There is no linefeed.

Output items

| Output items | Description |
|---|---|
| *id* | Identifier of the Java VM log file |

| Output items | Description |
|---|---|
| *date* | Date and time when GC started |
| *full_count* | Number of times Full GC was skipped. (This item is output if the `-XX:HitachiVerboseGCIntervalTime` option is specified.) |
| *copy_count* | Number of times CopyGC was skipped. This item is output if the `-XX:HitachiVerboseGCIntervalTime` option is specified. |
| *gc_kind* | GC type (`Full GC` or `GC`). |
| *gc_info* | GC information (`(area_length_before_GC -> area-length-after-GC(area_size))`) (in kilobytes).<br>Example: `264K->0K(512K)` |
| *gc_time* | Elapsed time (in seconds) for GC |
| *eden_info* | Eden area information (in kilobytes) |
| *survivor_info* | Memory information (in kilobytes) of the Survivor area |
| *tenured_info* | Memory information (in kilobytes) of the Tenured area |
| *metaspace_info* | Memory information (in kilobytes) of the Metaspace area |
| *classspace_info* | CompressedClassSpace information (in kilobytes)<br>If the compressed object pointer functionality is disabled, `0` is output as dummy information. |
| *cause_info* | Cause number of GC. This item is not output if the `-XX:-HitachiVerboseGCPrintCause` option is specified. |
| *user_cpu* | CPU time (in seconds) that the GC thread consumed in the user mode.<br>If an attempt to obtain the CPU time failed, `unknown` is displayed. This item is not output if the `-XX:-HitachiVerboseGCCpuTime` option is specified. |
| *system_cpu* | CPU time (in seconds) that the GC thread consumed in the kernel mode.<br>If an attempt to obtain the CPU time failed, `unknown` is displayed. This item is not output if the `-XX:-HitachiVerboseGCCpuTime` option is specified. |
| *jvm_alloc_size* | Size of the area currently in use, from among the areas being managed in the Java VM |
| *mmap_total_size* | Total size of the C heap area allocated for mmap , from among the areas being managed in the Java VM. This item is not output if the `-XX:-HitachiVerboseGCPrintJVMInternalMemory` option is specified. |
| *malloc_total_size* | Total size of the C heap area allocated for malloc, from among the areas being managed in the Java VM. This item is not output if the `-XX:-HitachiVerboseGCPrintJVMInternalMemory` option is specified. |
| *thread_count* | The number of Java threads. This item is not output if the `-XX:-HitachiVerboseGCPrintJVMInternalMemory` option is specified. |
| *doe_alloc_size* | Cumulative heap size allocated by invoking `java.io.File.deleteOnExit()`. This item is not output if the `-XX:-HitachiVerboseGCPrintDeleteOnExit` option is specified. |
| *called_count* | Invocation count of `java.io.File.deleteOnExit()`. This item is not output if the `-XX:-HitachiVerboseGCPrintDeleteOnExit` option is specified. |
| *cc_used_size* | Code cache area (in kilobytes) used when GC occurs. This item is not output if the `-XX:-PrintCodeCacheInfo` option is specified. |
| *cc_max_size* | Maximum size of the code cache area (in kilobytes). This item is not output if the `-XX:-PrintCodeCacheInfo` option is specified. |
| *cc_info* | Maintenance information. This item is not output if the `-XX:-PrintCodeCacheInfo` option is specified. |

CSV format output (when G1GC is used)

Output format

```
id,date,gc_kind,gc_info,gc_time,gc_status,eden_info,survivor_info,
tenured_info,humongous_info,free_info,metaspace_info,
classspace_info,cause_info,region_size,target_time,predicted_time,
target_size,reclaimable_info,user_cpu,system_cpu,jvm_alloc_size,
mmap_total_size,malloc_total_size,thread_count,doe_alloc_size,
called_count,cc_used_size,cc_max_size,cc_info
```

Note:

There is no linefeed.

Output items

| Output items | Description |
|---|---|
| *id* | Identifier of the Java VM log file |
| *date* | Date and time when GC started |
| *gc_kind* | Type of GC or CM.<br>`Full GC`, `Mixed GC`, `Young GC`, `Young GC(initial-mark)`, `CM Remark`, or `CM Cleanup` is output. |
| *gc_info* | Memory information (in kilobytes) of the Java heap area |
| *gc_time* | Duration (in seconds) during which the application is stopped by GC |
| *gc_status* | GC status (`to exhausted` or `-`).<br>If gc_kind is `Young GC`, `Young GC(initial-mark)`, or `Mixed GC`<br>`-` or `to exhausted` is output.<br>If gc_kind is other than `Young GC`, `Young GC(initial-mark)`, or `Mixed GC`<br>`-` is output. |
| *eden_info* | Memory information (in kilobytes) of the Eden area |
| *survivor_info* | Memory information (in kilobytes) of the Survivor area |
| *tenured_info* | Memory information (in kilobytes) of the Tenured area |
| *humongous_info* | Memory information (in kilobytes) of the Humongous area |
| *free_info* | Memory information (in kilobytes) of the Free area |
| *metaspace_info* | Memory information (in kilobytes) of the Metaspace area |
| *classspace_info* | CompressedClassSpace information (in kilobytes) |
| *cause_info* | Cause of GC |
| *region_size* | Size of a region (in kilobytes) |
| *target_time* | Target time during which the application is stopped by GC (in seconds) |
| *predicted_time* | Duration (in seconds) that the Java VM predicted the application would be stopped by GC |
| *target_size* | Size of the GC-target Tenured area in Mixed GC. |
| *reclaimable_info* | Predicted collection size information (in kilobytes). |
| *user_cpu* | CPU time (in seconds) that all GC threads consumed in the user mode. This item is not output if the `-XX:-HitachiVerboseGCCpuTime` option is specified.<br>If an attempt to obtain the CPU time failed, `unknown` is displayed. |

| Output items | Description |
|---|---|
| *system_cpu* | CPU time (in seconds) that all GC threads consumed in the kernel mode. This item is not output if the -XX:-HitachiVerboseGCCpuTime option is specified.<br>If an attempt to obtain the CPU time failed, unknown is displayed. |

## Event log of the explicit memory management functionality

Output to the event log of the explicit memory management functionality is triggered by different events depending on the specified log output level.

There are three types of log output levels: normal, verbose, and debug. The following shows the output format for each level.

If the log output level is normal (Explicit heap usage status when GC occurred):

Output format

```
[ENS]ctime[EH:  EH_USED_BF->EH_USED_AF(EH_TOTAL/EH_MAX)]
[E/F/D:  AC_NUM/FL_NUM/DA_NUM][cause:CAUSE][CF:  CF_CNT]
```

Note:

There is no linefeed.

Output items

| Output items | Description |
|---|---|
| *ctime* | This indicates the date and time of the GC occurrence. This is the same time format as the logs output by the extended verbosegc functionality.<br>If the HitachiOutputMilliTime option is enabled, the time is output in milliseconds. |
| *EH_USED_BF* | Used size (in kilobytes) of the Explicit heap before GC is output |
| *EH_USED_AF* | Used size of the Explicit heap after GC |
| *EH_TOTAL* | Size (in kilobytes) of the allocated Explicit heap after GC |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is Enable after GC |
| *FL_NUM* | This output item is used for an extension. The output is always 0. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is Disable after GC |
| *CAUSE* | GC indicates that Copy GC triggered the output of this log. Full GC indicates Full GC triggered the output of this log. |
| *CF_CNT* | Failure count of initializations of Explicit memory blocks from the occurrence of the previous GC to the occurrence of the current GC. |

If the log output level is normal (processing to release Explicit memory blocks):

Output format

```
[ENS]ctime[EH:  EH_USED_BF->EH_USED_AF(EH_TOTAL/EH_MAX),
ELAPSED secs][E/F/D:  AC_NUM/FL_NUM/DA_NUM]
[DefNew::Eden:  ED_USED_BF->ED_USED_AF(ED_TOTAL)]
[DefNew::Survivor:  SV_USED_BF->SV_USED_AF(SV_TOTAL)]
[Tenured:  TN_USED_BF->TN_USED_AF(TN_TOTAL)][User:  USERCPU secs]
[Sys:  SYSCPU secs][cause:CAUSE]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when processing to release Explicit memory blocks occurred. This is the same time format as the logs output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EH_USED_BF* | Used size (in kilobytes) of the Explicit heap before the processing to release Explicit memory blocks |
| *EH_USED_AF* | Used size (in kilobytes) of the Explicit heap after the processing to release Explicit memory blocks |
| *EH_TOTAL* | Size (in kilobytes) of the allocated Explicit heap after the processing to release Explicit memory blocks |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *ELAPSED* | Time (in seconds) required for the processing to release Explicit memory blocks |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is `Enable` after execution of the processing to release Explicit memory blocks |
| *FL_NUM* | This output item is used for an extension and always shows `0`. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is `Disable` after execution of the processing to release Explicit memory blocks |
| *ED_USED_BF* | Used size (in kilobytes) of the Eden area before the processing to release Explicit memory blocks |
| *ED_USED_AF* | Used size (in kilobytes) of the Eden area after the processing to release Explicit memory blocks |
| *ED_TOTAL* | Allocated size (in kilobytes) of the Eden area after the processing to release Explicit memory blocks |
| *SV_USED_BF* | Used size (in kilobytes) of the Survivor area before the processing to release Explicit memory blocks |
| *SV_USED_AF* | Used size (in kilobytes) of the Survivor area after the processing to release Explicit memory blocks |
| *SV_TOTAL* | Allocated size (in kilobytes) of the Survivor area after the processing to release Explicit memory blocks |
| *TN_USED_BF* | Used size (in kilobytes) of the Tenured area before the processing to release Explicit memory blocks |
| *TN_USED_AF* | Used size (in kilobytes) of the Tenured area after the processing to release Explicit memory blocks |
| *TN_TOTAL* | Allocated size (in kilobytes) of the Tenured area after the processing to release Explicit memory blocks |
| *USERCPU* | User CPU time (in seconds) required for the processing to release Explicit memory blocks |
| *SYSCPU* | System CPU time (in seconds) required for the processing to release Explicit memory blocks |
| *CAUSE* | `Reclaim` is always output. This indicates that this log data was output by the release of Explicit memory blocks. |

If the log output level is `normal` (Java heap overflow during processing to release Explicit memory blocks):

Output format

```
[ENS]ctime[EH:  EH_USED_BF->EH_USED_AF(EM_TOTAL/EH_MAX),
ELAPSED secs][E/F/D:  AC_NUM/FL_NUM/DA_NUM]
[DefNew::Eden:  ED_USED_BF->ED_USED_AF(ED_TOTAL)]
[DefNew::Survivor:  SV_USED_BF->SV_USED_AF(SV_TOTAL)]
[Tenured:  TN_USED_BF->TN_USED_AF(TN_TOTAL)][User:  USERCPU secs]
[Sys:  SYSCPU secs][cause:CAUSE]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when processing to release Explicit memory blocks occurred. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EH_USED_BF* | Used size (in kilobytes) of the Explicit heap before the processing to release Explicit memory blocks |

| Output items | Description |
|---|---|
| *EH_USED_AF* | Used size of the Explicit heap after a Java heap overflow. When Java heap overflows, the processing to release Explicit memory blocks is not executed, and thus the value (in kilobytes) is always the same as *EH_USED_BF*. |
| *EH_TOTAL* | Size (in kilobytes) of the allocated Explicit heap after a Java heap overflow |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *ELAPSED* | Time (in seconds) from the start of the processing to release Explicit memory blocks until the Java heap overflow |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is `Enable` after a Java heap overflow |
| *FL_NUM* | This output item is used for an extension and always shows `0`. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is `Disable` after a Java heap overflow |
| *ED_USED_BF* | Used size (in kilobytes) of the Eden area before the processing to release Explicit memory blocks |
| *ED_USED_AF* | Used size (in kilobytes) of the Eden area after a Java heap overflow |
| *ED_TOTAL* | Allocated size (in kilobytes) of the Eden area after a Java heap overflow |
| *SV_USED_BF* | Used size (in kilobytes) of the Survivor area before the processing to release Explicit memory blocks |
| *SV_USED_AF* | Used size (in kilobytes) of the Survivor area after a Java heap overflow |
| *SV_TOTAL* | Allocated size (in kilobytes) of the Survivor area after a Java heap overflow |
| *TN_USED_BF* | Used size (in kilobytes) of the Tenured area before the processing to release Explicit memory blocks |
| *TN_USED_AF* | Used size (in kilobytes) of the Tenured area after a Java heap overflow |
| *TN_TOTAL* | Allocated size (in kilobytes) of the Tenured area after a Java heap overflow |
|  | User CPU time (in seconds) from the start of the processing to release Explicit memory blocks until the Java heap overflow |
| *SYSCPU* | System CPU time (in seconds) from the start of the processing to release Explicit memory blocks until the Java heap overflow |
| *CAUSE* | `Reclaiming` is always output. This indicates that this log data was output by a Java heap overflow during the processing to release Explicit memory blocks. |

### If the log output level is `normal` (processing to migrate Explicit memory blocks):

#### Output format

```
[ENS]ctime[EH:  EH_USED_BF->EH_USED_AF(EH_TOTAL>/EH_MAX),
ELAPSED secs][E/F/D:  AC_NUM/FL_NUM/DA_NUM]
[DefNew::Eden:  ED_USED_BF->ED_USED_AF>(ED_TOTAL)]
[DefNew::Survivor:  SV_USED_BF->SV_USED_AF(SV_TOTAL)]
[Tenured:  TN_USED_BF->TN_USED_AF(TN_TOTAL)]
[target:EH_MIG_TRG/EH_MIG_DED/EH_MIG_LIV]
[User:  USERCPU secs][Sys:  SYSCPU secs][cause:CAUSE]
```

#### Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when automatic migration of Explicit memory blocks was reserved. This is the same time format as the one for items output by the extended verbosegc functionality.<br>If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EH_USED_BF* | Used size (in kilobytes) of the Explicit heap before the processing to migrate Explicit memory blocks |
| *EH_USED_AF* | Used size (in kilobytes) of the Explicit heap after the processing to migrate Explicit memory blocks |

| Output items | Description |
|---|---|
| *EH_TOTAL* | Size (in kilobytes) of the allocated Explicit heap after the processing to migrate Explicit memory blocks |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *ELAPSED* | Time (in seconds) from the start of the processing to reserve migration of Explicit memory blocks until the end of the migration processing |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is `Enable` after execution of the processing to migrate Explicit memory blocks |
| *FL_NUM* | This output item is used for an extension and always shows `0`. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is `Disable` after execution of the processing to migrate Explicit memory blocks |
| *ED_USED_BF* | Used size (in kilobytes) of the Eden area before the processing to migrate Explicit memory blocks |
| *ED_USED_AF* | Used size (in kilobytes) of the Eden area after the processing to migrate Explicit memory blocks |
| *ED_TOTAL* | Allocated size (in kilobytes) of the Eden area after the processing to migrate Explicit memory blocks |
| *SV_USED_BF* | Used size (in kilobytes) of the Survivor area before the processing to migrate Explicit memory blocks |
| *SV_USED_AF* | Used size (in kilobytes) of the Survivor area after the processing to migrate Explicit memory blocks |
| *SV_TOTAL* | Allocated size (in kilobytes) of the Survivor area after the processing to migrate Explicit memory blocks |
| *TN_USED_BF* | Used size (in kilobytes) of the Tenured area before the processing to migrate Explicit memory blocks |
| *TN_USED_AF* | Used size (in kilobytes) of the Tenured area after the processing to migrate Explicit memory blocks |
| *TN_TOTAL* | Allocated size (in kilobytes) of the Tenured area after the processing to migrate Explicit memory blocks |
| *EH_MIG_TRG* | Used size (in kilobytes) of the Explicit heap to which the processing to migrate Explicit memory blocks was executed |
| *EH_MIG_DED* | Used size (in kilobytes) of the Explicit heap that decreased by the execution of the processing to migrate Explicit memory blocks |
| *EH_MIG_LIV* | Used size (in kilobytes) of the Explicit heap that did not decrease by the execution of the processing to migrate Explicit memory blocks |
| *USERCPU* | User CPU time (in seconds) from the start of the processing to reserve migration of Explicit memory blocks until the end of the migration processing |
| *SYSCPU* | System CPU time (in seconds) from the start of the processing to reserve automatic migration of Explicit memory blocks until the end of the migration processing |
| *CAUSE* | `Migrate` is always output. This indicates that this log data was output by the processing to migrate Explicit memory blocks |

If the log output level is `normal` (Java heap overflow during processing to migrate Explicit memory blocks):

Output format

```
[ENS]ctime[EH:  EH_USED_BF->EH_USED_AF(EH_TOTAL/EH_MAX),
ELAPSED secs][E/F/D:  AC_NUM/FL_NUM/DA_NUM]
[DefNew::Eden:  ED_USED_BF->ED_USED_AF(ED_TOTAL)]
[DefNew::Survivor:  SV_USED_BF->SV_USED_AF(SV_TOTAL)]
[Tenured:  TN_USED_BF->TN_USED_AF(TN_TOTAL)]
[target:EH_MIG_TRG/EH_MIG_DED/EH_MIG_LIV]
[User:  USERCPU secs][Sys:  SYSCPU secs][cause:CAUSE]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when processing to migrate Explicit memory blocks occurred.<br>This is the same time format as the one for items output by the extended verbosegc functionality.<br>If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EH_USED_BF* | Used size (in kilobytes) of the Explicit heap before the processing to migrate Explicit memory blocks |
| *EH_USED_AF* | Used size (in kilobytes) of the Explicit heap after a Java heap overflow |
| *EH_TOTAL* | Size (in kilobytes) of the allocated Explicit heap after a Java heap overflow |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *ELAPSED* | Time (in seconds) from the start of the processing to migrate Explicit memory blocks until the Java heap overflow |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is `Enable` after execution of the processing to migrate Explicit memory blocks |
| *FL_NUM* | This output item is used for an extension and always shows `0`. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is `Disable` after a Java heap overflow |
| *ED_USED_BF* | Used size (in kilobytes) of the Eden area before the processing to migrate Explicit memory blocks |
| *ED_USED_AF* | Used size (in kilobytes) of the Eden area after a Java heap overflow |
| *ED_TOTAL* | Allocated size (in kilobytes) of the Eden area after a Java heap overflow |
| *SV_USED_BF* | Used size (in kilobytes) of the Survivor area before the processing to migrate Explicit memory blocks |
| *SV_USED_AF* | Used size (in kilobytes) of the Survivor area after a Java heap overflow |
| *SV_TOTAL* | Allocated size (in kilobytes) of the Survivor area after a Java heap overflow |
| *TN_USED_BF* | Used size (in kilobytes) of the Tenured area before the processing to migrate Explicit memory blocks |
| *TN_USED_AF* | Used size (in kilobytes) of the Tenured area after a Java heap overflow |
| *TN_TOTAL* | Allocated size (in kilobytes) of the Tenured area after a Java heap overflow |
| *EH_MIG_TRG* | Used size (in kilobytes) of the Explicit heap to which the processing to migrate Explicit memory blocks was executed |
| *EH_MIG_DED* | Used size (in kilobytes) of the Explicit heap that had decreased by the execution of the processing to migrate Explicit memory blocks performed until the Java heap overflow `0K` is always output. |
| *EH_MIG_LIV* | Used size (in kilobytes) of the Explicit heap that had not decreased by the execution of the processing to migrate Explicit memory blocks performed until the Java heap overflow This does not include the size of the object that caused the Java heap overflow. |
| *USERCPU* | User CPU time (in seconds) from the start of the processing to migrate Explicit memory blocks until the Java heap overflow |
| *SYSCPU* | System CPU time (in seconds) from the start of the processing to migrate Explicit memory blocks until the Java heap overflow |
| *CAUSE* | `Migrating` is always output. This indicates that this log data was output by Java heap overflow during the processing to migrate Explicit memory blocks. |

If the log output level is `normal` (error in opening the automatic allocation configuration file for explicitly managed memory):

Output format

```
[ENA]ctime failed to open file. [file=FILENAME]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when an attempt to open the automatic allocation configuration file for explicitly managed memory failed. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *FILENAME* | Name of the automatic allocation configuration file that failed to open (excluding the directory name). |

If the log output level is `normal` (error in parsing the automatic allocation configuration file for explicitly managed memory):

Output format

```
[ENA]ctime parsed error line. [file=FILENAME line=LINENO]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the automatic allocation configuration file for the explicit memory management functionality could not be parsed. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *FILENAME* | Name of the automatic allocation configuration file that could not be parsed (excluding the directory name) |
| *LINENO* | Line number for which the parsing failed |

If the log output level is `normal` (error in automatic allocation of explicitly managed memory):

Output format

```
[ENA]ctime creation
CLASS_LIST class's object in explicit memory is failed.
[target=CLASS_METHOD
detail=MESSAGE]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when an attempt of the explicit memory management functionality to allocate explicitly managed memory failed. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *CLASS_LIST* | List of fully qualified class names of the objects for which an attempt was made to allocate explicitly managed memory. The list might be blank. |
| *CLASS_METHOD* | Fully qualified names of the classes for which an attempt to allocate explicitly managed memory failed. The method names indicating more detailed failure locations might also be output. |
| *MESSAGE* | Detailed message indicating the cause of the failure during the allocation of explicitly managed memory. |

If the log output level is `normal` (skipping automatic allocation of explicitly managed memory):

Output format

```
[ENA]ctime creation class's object in explicit memory is skipped. [
detail = MESSAGE]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the explicit memory management functionality skipped allocation of explicitly managed memory. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *MESSAGE* | Detailed message indicating the cause of the skipping of the allocation of explicitly managed memory. |

If the log output level is `normal` (error in opening the configuration file of the functionality for specifying the classes to be excluded from the explicit memory management functionality):

Output format

```
[ENO]ctime failed to open file. [TYPE] [file=FILENAME]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when an attempt to open the configuration file of the functionality for specifying the classes to be excluded from the explicit memory management functionality failed. This is the same time format as the one for items output by the extended verbosegc functionality.<br>If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *TYPE* | This indicates the type of the configuration file that could not be opened or read.<br>`SYS`: Configuration file configured by the system<br>`USR`: Configuration file whose file path was specified in an option by the user<br>`DEF`: Configuration file, at the default location, for the user |
| *FILENAME* | Name of the configuration file that failed to open (excluding the directory name). |

If the log output level is `normal` (error in parsing the configuration file of the functionality for specifying the classes to be excluded from the explicit memory management functionality):

Output format

```
[ENO]ctime parsed error line. [TYPE] [file=FILENAME line=LINENO]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the configuration file of the functionality for specifying the classes to be excluded from the explicit memory management functionality could not be parsed. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *TYPE* | This indicates the type of the configuration file that could not be parsed.<br>`SYS`: Configuration file configured by the system<br>`USR`: Configuration file whose file path was specified in the option by the user<br>`DEF`: Configuration file, at the default location, for the user |
| *FILENAME* | Name of the configuration file that could not be parsed (excluding the directory name). |
| *LINENO* | Line number for which the parsing failed. |

If the log output level is `verbose` (initialization of Explicit memory blocks):

Output format

```
[EVO]ctime>[Created]["EM_NAME" eid=EID(EM_PTR)/EM_TYPE]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the Explicit memory block was initialized. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EM_NAME* | Name of the initialized Explicit memory block.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. |
| *EID* | ID of the initialized Explicit memory block |
| *EM_PTR* | Value that indicates the internal status of the Explicit memory block |
| *EM_TYPE* | This indicates the type of the Explicit memory block in the Java VM. |

If the log output level is `verbose` (failure in initialization of Explicit memory blocks):

Output format

```
[EVO]ctime[Creation failed][EH:  EH_USED(EH_GARB)/EH_TOTAL/EH_MAX]
[E/F/D:  AC_NUM/FL_NUM/DA_NUM][Thread:  TH_PTR]
[EVO][Thread:  TH_PTR] at FRAMESOURCE
...
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when initialization of the Explicit memory block failed. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EH_USED* | Used size of the Explicit heap when initialization of the Explicit memory block failed |
| *EH_GARB* | This output item is used for an extension and indicates the internal status (in kilobytes) of the Explicit heap. |
| *EH_TOTAL* | Allocated size (in kilobytes) of the Explicit heap when initialization of the Explicit memory block failed |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is `Enable` after the initialization of the Explicit memory block failed |
| *FL_NUM* | This output item is used for an extension and always shows `0`. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is `Disable` after the initialization of the Explicit memory block failed |
| *TH_PTR* | Thread ID of the thread in which initialization of the Explicit memory block failed. This is the same as the tid output to the thread dump. |
| *FRAME* | One frame in the stack trace when initialization of the Explicit memory block failed. The full class name and the method name are output, delimited by a period (`.`). |
| *SOURCE* | The name of the source file in which the method indicated by *FRAME* is coded and the line number that matches the stack trace are output, delimited by a colon (`:`).<br>For a native method, `(Native Method)` is output. If the source file name cannot be obtained, `(Unknown Source)` is output. |

If the log output level is `verbose` (sub status of Explicit memory blocks changed to FreeList):

Output format

```
[EVO]ctime[Alloc failed(FreeList)]
[EH:  EH_USED(EH_GARB)/EH_TOTAL/EH_MAX]
[E/F/D:  AC_NUM/FL_NUM/DA_NUM][cause:CAUSE]
```

```
["EM_NAME" eid=EID/EM_TYPE:  EM_USED(EM_GARB)/EM_TOTAL]
[Thread:  TH_PTR]
[EVO][Thread:  TH_PTR] at FRAMESOURCE
...
```

Note:

"[Thread: *TH_PTR*] [EVO][Thread: *TH_PTR*] at *FRAMESOURCE*" is output only for `New`.

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the sub status of the Explicit memory block indicated by *EID* changed to FreeList. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EH_USED* | Used size (in kilobytes) of the Explicit heap when the sub status of the Explicit memory block indicated by *EID* changed to FreeList |
| *EH_GARB* | This output item is used for an extension and indicates the internal status of the Explicit heap. |
| *EH_TOTAL* | Allocated size (in kilobytes) of the Explicit heap when the sub status of the Explicit memory block indicated by *EID* changed to FreeList |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is `Enable` after the sub status of the Explicit memory block indicated by *EID* changed to FreeList |
| *FL_NUM* | This output item is used for an extension and always shows `0`. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is `Disable` after the sub status of the Explicit memory block indicated by *EID* changed to FreeList |
| *CAUSE* | Processing that caused the sub status to become FreeList. `New` indicates that the cause was direct generation of the object in the Explicit heap by using `newInstance()`. `GC` indicates that the cause was processing to move to the Explicit heap at Copy GC. `Full GC` indicates that the cause was processing to move to the Explicit heap at Full GC. |
| *EM_NAME* | Name of the Explicit memory block whose sub status became FreeList.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. `NULL` might be output if the log data was output at about the same time as initialization of the Explicit memory block, or if the Java VM generated the Explicit memory block internally. |
| *EID* | ID of the Explicit memory block whose sub status became FreeList |
| *EM_TYPE* | Type of the Explicit memory block whose sub status became FreeList.<br>This indicates the type of the Explicit memory block in the Java VM. |
| *EM_USED* | Used size (in kilobytes) of the Explicit memory block whose sub status became FreeList |
| *EM_GARB* | This output item is used for an extension and indicates the internal status of the Explicit memory block. |
| *EM_TOTAL* | Allocated size (in kilobytes) of the Explicit memory block whose sub status became FreeList |
| *TH_PTR* | Thread ID of the thread that executed the generation to the Explicit heap, which caused the sub status to become FreeList This is the same as the tid output to the thread dump.<br>This item is output only if *CAUSE* is `New`. |
| *FRAME* | One frame in the stack trace of the direct generation to the Explicit heap, which caused the sub status to become FreeList. The full class name and the method name are output, delimited by a period (`.`).<br>This item is output only if *CAUSE* is `New`. |
| *SOURCE* | The name of the source file in which the method indicated by *FRAME* is coded and the line number that matches the stack trace are output, delimited by a colon (`:`). |

| Output items | Description |
|---|---|
|  | For a native method, (Native Method) is output. If the source file name cannot be obtained, (Unknown Source) is output.<br>This item is output only if *CAUSE* is New. |

If the log output level is verbose (sub status of Explicit memory block changed to Disable):

Output format

```
[EVO]ctime[Alloc failed(Disable)]
[EH:  EH_USED(EH_GARB)/EH_TOTAL/EH_MAX]
[E/F/D:  AC_NUM/FL_NUM/DA_NUM][cause:CAUSE]
["EM_NAME" eid=EID/EM_TYPE:  EM_USED(EM_GARB)/EM_TOTAL]
[Thread:  TH_PTR]
[EVO][Thread:  TH_PTR] at FRAMESOURCE
...
```

Note:

"*TH_PTR*] at *FRAMESOURCE*" is output only for New.

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the sub status of the Explicit memory block indicated by *EID* changed to Disable. This is the same time format as the one for items output by the extended verbosegc functionality. If the HitachiOutputMilliTime option is enabled, the time is output in milliseconds. |
| *EH_USED* | Used size (in kilobytes) of the Explicit heap when the sub status of the Explicit memory block indicated by *EID* changed to Disable |
| *EH_GARB* | This output item is used for an extension and indicates the internal status of the Explicit heap. |
| *EH_TOTAL* | Allocated size (in kilobytes) of the Explicit heap when the sub status of the Explicit memory block indicated by *EID* changed to Disable |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is Enable after the sub status of the Explicit memory block indicated by *EID* changed to Disable |
| *FL_NUM* | This output item is used for an extension and always shows 0. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is Enable after the sub status of the Explicit memory block indicated by *EID* changed to Disable |
| *CAUSE* | Processing that caused the sub status to become Disable.<br>New directly generates objects into the Explicit heap, such as newInstance().GC indicates that the cause was processing to move to the Explicit heap at Copy GC. Full GC indicates that the cause was processing to move to the Explicit heap at Full GC. |
| *EM_NAME* | Name of the Explicit memory block whose sub status became Disable.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. NULL might be output if the log data was output at about the same time as initialization of the Explicit memory block, or if the Java VM generated the Explicit memory block internally. |
| *EID* | ID of the Explicit memory block whose sub status became Disable |
| *EM_TYPE* | Type of the Explicit memory block whose sub status became Disable.<br>This indicates the type of the Explicit memory block in the Java VM. |
| *EM_USED* | Used size (in kilobytes) of the Explicit memory block whose sub status became Disable |
| *EM_GARB* | This output item is used for an extension and indicates the internal status of the Explicit memory block. |

| Output items | Description |
|---|---|
| *EM_TOTAL* | Allocated size (in kilobytes) of the Explicit memory block whose sub status became `Disable` |
| *TH_PTR* | ID of the thread that generated an Explicit heap, which caused the sub status to become Disable. This is the same as the tid output to the thread dump.<br>This item is output only if *CAUSE* is `New`. |
| *FRAME* | One frame in the stack trace of direct generation to the Explicit heap, which caused the sub status to become FreeList. The full class name and the method name are output, delimited by a period (`.`).<br>This item is output only if *CAUSE* is `New`. |
| *SOURCE* | The name of the source file in which the method indicated by *FRAME* is coded and the line number that matches the stack trace are output, delimited by a colon (`:`).<br>For a native method, `(Native Method)` is output. If the source file name cannot be obtained, `(Unknown Source)` is output.<br>This item is output only if *CAUSE* is `New`. |

If the log output level is `verbose` (object generation for Explicit memory blocks):

Output format

```
[EVS]ctime[EH:  EH_USED_BF->EH_USED_AF(EH_TOTAL/EH_MAX)]
[E/F/D:  AC_NUM/FL_NUM/DA_NUM][cause:CAUSE]
["EM_NAME" eid=EID/EM_TYPE:  EM_USED_BF->EM_USED_AF(EM_TOTAL)]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the object was generated. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EH_USED_BF* | Used size (in kilobytes) of the Explicit heap before the object is generated |
| *EH_USED_AF* | Used size (in kilobytes) of the Explicit heap after the object was generated |
| *EH_TOTAL* | This indicates the size (in kilobytes) of the allocated Explicit heap after the object was generated. |
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *AC_NUM* | Number of valid Explicit memory blocks whose sub status is `Enable` after the object was generated |
| *FL_NUM* | This output item is used for an extension and always shows `0`. |
| *DA_NUM* | Number of valid Explicit memory blocks whose sub status is `Disable` after the object was generated |
| *CAUSE* | `New` is always output. This indicates that the object for the Explicit memory block was generated from a Java program. |
| *EM_NAME* | Name of the Explicit memory block in which the object was generated.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. `NULL` might be output if the log data was output at about the same time as initialization of the Explicit memory block, or if the Java VM generated the Explicit memory block internally. |
| *EID* | ID of the Explicit memory block in which the object was generated |
| *EM_TYPE* | Type of the Explicit memory block in which the object was generated.<br>This indicates the type of the Explicit memory block in the Java VM. |
| *EM_USED_BF* | Used size (in kilobytes) of the target Explicit memory block before the object is generated |
| *EM_USED_AF* | Used size (in kilobytes) of the target Explicit memory block after the object was generated |
| *EM_TOTAL* | Allocated size (in kilobytes) of the target Explicit memory block after the object was generated |

If the log output level is `verbose` (details about moving to Explicit memory blocks):

Output format

```
output_items_for_the_normal_log_output_level
[EVS]{["EM_NAME" eid=EID/EM_TYPE:  EM_USED_BF->EM_USED_AF(EM_TOTAL)]}{1,5}
...
```

Note:

A linefeed is output whenever five items of the Explicit memory block information are output.

Output items

| Output items | Description |
|---|---|
| *output_items_for_the_normal_log_output_level* | Same as the output items for the `normal` log output level (usage status of the Explicit heap when GC occurred). |
| *EM_NAME* | Name of the target Explicit memory block in which the object is to be moved during GC.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. `NULL` might be output if the log data was output at about the same time as initialization of the Explicit memory block, or if the Java VM generated the Explicit memory block internally. |
| *EID* | ID of the target Explicit memory block in which the object is to be moved during GC |
| *EM_TYPE* | Type of the target Explicit memory block in which the object is to be moved during GC.<br>This indicates the type of the Explicit memory block in the Java VM in `R` or `A`. |
| *EM_USED_BF* | Used size (in kilobytes) of the target Explicit memory block before GC, in which the object is to be moved during GC |
| *EM_USED_AF* | Used size (in kilobytes) of the target Explicit memory block after a GC during which the object was moved |
| *EM_TOTAL* | Allocated size (in kilobytes) of the target Explicit memory block after a GC during which the object was moved |

If the log output level is `verbose` (details about processing to release Explicit memory blocks):

Output format

```
output_items_for_the_normal_log_output_level
[EVS]{["EM_NAME" eid=EID/EM_TYPE: EM_TOTAL]}{1,5}
...
```

Note:

A linefeed is output whenever five items of the Explicit memory block information are output.

Output items

| Output items | Description |
|---|---|
| *output_items_for_the_normal_log_output_level* | Same as the output items for the `normal` log output level (processing to release Explicit memory blocks). |
| *EM_NAME* | Name of the released Explicit memory block.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. |
| *EID* | ID of the released Explicit memory block |
| *EM_TYPE* | Type of the released Explicit memory block.<br>This indicates the type of the Explicit memory block in the Java VM. |
| *EM_TOTAL* | Size (in kilobytes) of the released Explicit heap that was allocated (released size) |

If the log output level is `verbose` (Explicit memory block release reservation by the finalizer):

Output format

```
[EVO]ctime[Finalized]["EM_NAME" eid=EID/EM_TYPE:  EM_TOTAL]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time of release reservation. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *EM_NAME* | Name of the Explicit memory block that was reserved for release.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. |
| *EID* | ID of the Explicit memory block that was reserved for release |
| *EM_TYPE* | Type of the Explicit memory block that was reserved for release.<br>This indicates the type of the Explicit memory block in the Java VM. |
| *EM_TOTAL* | Size (in kilobytes) of the Explicit heap that was reserved for release and that was allocated (released size) |

If the log output level is `verbose` (allocation of explicitly managed memory):

Output format

```
[EVA]ctime creation in explicit memory is succeeded. [class=CLASSNAME]
```

Output items

| Output items | Description |
|---|---|
| *ctime* | Date and time when the explicit memory management functionality succeeded in allocating explicitly managed memory of the specified class. This is the same time format as the one for items output by the extended verbosegc functionality. If the `HitachiOutputMilliTime` option is enabled, the time is output in milliseconds. |
| *CLASSNAME* | Name of the fully qualified class that succeeded in allocating explicitly managed memory |

If the log output level is `debug` (moving objects to Java heap by Explicit memory block release processing):

Output format

```
[EDO][eid=EID:  Reference to REFED_NAME(REFED_PTR), total R_SIZE]
[EDO]   REF_NAME(REF_PTR)REF_GEN
```

Output items

| Output items | Description |
|---|---|
| *EID* | ID of the Explicit memory block that holds the object referenced by heaps other than the release target Explicit heap during processing to release Explicit memory blocks |
| *REFED_NAME* | Full class name of the Explicit memory block referenced by objects (*REF_NAME* (*REF_PTR*)) other than the release target Explicit heap during processing to release Explicit memory blocks |
| *REF_PTR* | Memory address of the object indicated by *REFED_NAME* (Before moving to Java heap) |
| *R_SIZE* | Total size of the object that is to be returned to Java heap by referencing from *REF_NAME* (*REF_PTR*).<br>Value (in kilobytes) including the objects in the release-target Explicit memory block that is indirectly referenced from *REF_NAME* (*REF_PTR*). |

| Output items | Description |
|---|---|
| *REFED_NAME* | Full class name of the object that holds references to *REFED_NAME* (*REFED_PTR*). However, if *REFED_NAME* (*REFED_PTR* is referenced from the stack or from inside the Java VM, `JVM` is output. |
| *REFED_PTR* | Memory address for *REF_NAME* |
| *REF_GEN* | Name of the area (or generation) where *REF_NAME* (*REF_PTR*) belongs.<br>For an Explicit memory block, eid is output. If referenced from the stack or from inside the Java VM, `JVM` is output.<br>For the New area, `DefNew` is output when serial garbage collector is used for minor GC. |

If the log output level is `debug` (details about initialization of Explicit memory blocks):

Output format

```
output_items_for_the_verbose_log_output_level
[Thread: TH_PTR]
[EDO][Thread: TH_PTR] at FRAMESOURCE
...
```

Output items

| Output items | Description |
|---|---|
| *output_items_for_the_verbose_log_output_level* | Same as the output items for the `verbose` log output level (initialization of Explicit memory blocks). |
| *TH_PTR* | Thread ID of the thread in which the Explicit memory block was initialized. This is the same as the *tid* output to the thread dump. |
| *FRAME* | One frame in the stack trace when the Explicit memory block was initialized. The full class name and the method name are output, delimited by a period (`.`). |
| *SOURCE* | The name of the source file in which the method indicated by *FRAME* is coded and the line number that matches the stack trace are output, delimited by a colon (`:`).<br>For a native method, `(Native Method)` is output. If the source file name cannot be obtained, `(Unknown Source)` is output. |

If the log output level is `debug` (details about processing to migrate Explicit memory blocks):

Output format

```
output_items_for_the_normal_log_output_level
[EDO][migrate:(EID_DEL{,EID_DEL}*|)/(EID_MBF{,EID_MBF}*->EID_MAF|)
/(EID_MIG{,EID_MIG}*|)]
```

Output items

| Output items | Description |
|---|---|
| *output_items_for_the_normal_log_output_level* | Same as the output items for the `normal` log output level (processing to migrate Explicit memory blocks). |
| *EID_DEL* | ID of the Explicit memory block whose object was not moved, among Explicit memory blocks that are released during the processing to migrate Explicit memory blocks |
| *EID_MBF* | ID of the Explicit memory block before many-to-one migration is performed, among Explicit memory blocks that are released during processing to migrate Explicit memory blocks |
| *EID_MAF* | ID of the Explicit memory block that is generated during processing to migrate Explicit memory blocks |
| *EID_MIG* | ID of the Explicit memory block that is released during processing to migrate Explicit memory blocks |

## Thread dump log

The following table shows the configuration of Java VM thread dump information.

| Output items | Description |
|---|---|
| Header | The date and time, Java VM version information, and start command line are output. |
| System settings | The following information is output:<br>• Java home path indicating the installation location of the JDK execution environment<br>• Java DLL path indicating the installation directory of the libraries making up the JDK<br>• System class path<br>• Java command options |
| Operation environment | The following information is output:<br>• Host name<br>• OS version<br>• CPU information<br>• Resource information |
| Java heap information | The memory usage status of each generation of Java heap is output. |
| C heap map information | The area information of the memory secured by the Java VM itself is output. |
| C heap size information | The size information of the memory secured by the Java VM itself is output. |
| Application environment | The following information is output:<br>• Signal handler<br>• Environment variable |
| Library information | The loaded library information is output. |
| Thread information<br>*thread_1*<br>...<br>*thread_n* | The thread information for each thread is output. |
| Java monitor dump | A list of Java monitor objects is displayed. |
| JNI global reference information | The number of global references to JNI maintained by the Java VM. |
| Explicit heap information (when -XX:+HitachiUseExplicitMemory is enabled) | If you use the explicit memory management functionality, the following information is output for each class of the Java process:<br>• Use status of the entire Explicit heap<br>• Use status for each Explicit memory block<br>In addition, if you execute the `eheapprof` command while using the explicit memory management functionality, the statistics about the objects within the Explicit memory block and the release rate information of the Explicit memory block are output. |
| Class-separated statistics | The following information is output for each class of the Java process specified by the `jheapprof` command:<br>• The total size of the instances possessed by an instance as members, and reference relationship of the instances<br>• The total size of the instances possessed by the static member<br>• The total size of the instances and classes of the object that causes the Tenured area to increase |
| Footer | The time when the thread dump ended is displayed. |

The following describes the output format of each item.

Header

Output format

```
EEE_MMM_dd_hh:mm:ss_yyyy

Full thread dump Java HotSpot(TM) VM_type (Oracle_version_information
-Developer's_Kit_for_Java_version_information-build-date mixed mode)
command_line
```

Note:

*EEE* indicates a day of the week. *MMM* indicates a month. *dd* indicates a day.

Output items

| Output items | Description |
|---|---|
| *VM_type* | Client VM, Server VM, or 64-Bit Server VM |
| *Oracle_version_information* | The version of the base Oracle JDK.<br>`25.20-b23` |
| *Developer's_Kit_for_Java_version_information* | The version of the Developer's Kit for Java.<br>`HJDK1010ZZ`<br>(*ZZ* indicates the revision number. There is no regular version.) |
| *build-date* | The data when the product was built |
| *command_line* | Command line when the VM starts |

System settings

Output format

```
System Properties
-----------------
Java Home Dir    :
installation_directory_for_the_Developer's_Kit_for_Java_execution_environment
Java DLL Dir    :
installation_directory_for_the_Developer's_Kit_for_Java_library
Sys Classpath    : system_class_path
User Args    :
command_option_1
command_option_2
...
```

Output items

| Output items | Description |
|---|---|
| *installation_directory_for_the_Developer's_Kit_for_Java_execution_environment* | Outputs the Java home path indicating the installation location of the Developer's Kit for Java execution environment. |
| *installation_directory_for_the_Developer's_Kit_for_Java_library* | Outputs the Java DLL path indicating the installation directory of the libraries making up the Developer's Kit for Java. |
| *system_class_path* | The system class path is output. |
| *command_option_n* | The Java command options are output. |

Operation environment

Output format

For the Linux/EM64T version, information is output as follows:

```
Operating Environment
---------------------
```

```
Host    : host_name:IP_address
OS    : OS_version
CPU    : CPU_type, number_of_available_CPUs
/number_of_CPUs_of_the_entire_system active

Resource Limits -
        RLIMIT_CPU              :
number_of_seconds_that_can_be_used_by_the_process
        RLIMIT_FSIZE            : maximum_file_size_(in_bytes)
        RLIMIT_DATA             : dynamically_allocatable_size_(in_bytes)
        RLIMIT_STACK            : maximum_stack_size_(in_bytes)
        RLIMIT_CORE             : maximum_core_size_(in_bytes)
        RLIMIT_RSS              : process_resident_size_(in_bytes)
        RLIMIT_NOFILE           : maximum_file_descriptor_value
        RLIMIT_AS               : available_memory_for_all_processes_(in_bytes)
        RLIMIT_NPROC            : maximum_number_of_processes
        RLIMIT_MEMLOCK          : size_of_memory_that_can_be_locked_(in_bytes)
```

#### Output items

The following shows the output items of the Linux/EM64T version.

| Output items | Description |
|---|---|
| *host_name* | The machine name |
| *IP_address* | The IP address of the machine If there are multiple IP addresses, they are output delimited by a comma (`,`). |
| *OS_version* | The following information is output:<br>`OS_nameVersionRelease`<br>*OS_name*<br>`Linux` is always output.<br>Information equivalent to the UNIX command `uname -s` is output.<br>*Version*<br>The version information.<br>Information equivalent to the UNIX command `uname -v` is output.<br>*Release*<br>Release information.<br>Information equivalent to the UNIX command `uname -r` is output. |
| *CPU_type* | The CPU name is output.<br>Information equivalent to the UNIX command `uname -m` is output. |
| *number_of_available_CPUs* | The number of available logical CPUs is output. |
| *number_of_CPUs_of_the_entire_system* | The number of installed logical CPUs is output. |

### Java heap information

#### Output format

```
Heap Status
-----------
def new generation   max max_size, total capacity, used size
(max_usage% used/max, total_usage% used/total)
                       [bottom, commit_address, reserve_address)
  eden space capacity,  usage% used [bottom,
top, reserve_address)
  from space capacity,   usage% used [bottom,
top, reserve_address)
  to   space capacity,   usage% used [bottom,
top, reserve_address)
tenured generation   max max_size, total capacity>, used size
(max_usage% used/max, total_usage% used/total)
                      [bottom, commit_address, reserve_address)
   the space capacity,  usage% used [bottom, top,
used_block, reserve_address)
Metaspace       max max_size, capacity capacity_words,
committed committed_size,
reserved reserve_size, used size
(max_usage% used/max, total_usage% used/committed)
  class space    max max_size, capacity capacity_words,
committed committed_size,reserved reserve_size, used size
(max_usage% used/max, total_usage% used/committed)
                [bottom, top, commit_address,
reserve_address)
```

Output items

| Output items | Description |
|---|---|
| *max_size* | The maximum capacity (in kilobytes). For Metaspace, if `-XX:MaxMetaspaceSize` is not specified, the maximum Metaspace value is unlimited. Therefore, `unlimited` is output as the maximum capacity. |
| *capacity* | The current capacity (in kilobytes) |
| *capacity words* | The total size (in kilobytes) of the committed memory in Metaspace, from which the free area is excluded |
| *committed_size* | The total size (in kilobytes) of the committed memory in Metaspace |
| *reserve_size* | The reserved memory size (in kilobytes) |
| *size* | The size of the memory in use (in kilobytes) |
| *max_usage* | The usage rate relative to the maximum capacity. For Metaspace, if `-XX:MaxMetaspaceSize` is not specified, the maximum Metaspace value is unlimited. Therefore, `-%` is output as the usage rate relative to the maximum capacity. |
| *total_usage* | The usage rate relative to the current capacity |
| *bottom* | The beginning address of the area |
| *top* | The beginning address of the area in use |
| *commit address* | The end address of the committed area |
| *reserve address* | The end address of the reserved area |
| *usage* | The usage rate |
| *used block* | The beginning address of the next free block |

Java VM internal memory map information

Output format

```
JVM Internal Memory Map
-----------------------
function_that_secures_memory    :address = starting_address -
end_address ( size: size)
...
```

Output items

| Output items | Description |
|---|---|
| *function_that_secures_memory* | Either `mmap()` or `malloc()`. The address is indicated by a hexadecimal number. |

Java VM internal memory size information

Output format

```
JVM Internal Memory Status
--------------------------
  Heap Size    :secured_memory_size (in bytes)
  Alloc Size    :size_of_the_memory_being_used (in bytes)
  Free Size    :size_of_the_unused_memory (in bytes)
```

Application environment

Output format

```
Application Environment
-----------------------
Signal Handlers -
signal_type:  [signal_handler_address], sa_mask[0]=signal_mask,
sa_flags=special_flag
:

Changed Signal Handlers -
signal_type:  [signal_handler_address], sa_mask[0]=signal_mask,
sa_flags=special_flag
:

Environment Variables -
environment_variable=value

Current Directory -
  /opt/Cosminexus/CC/server/...:current_directory
```

Output items

| Output items | Description |
|---|---|
| *signal_type* | The signal name defined in `/usr/include/sys/signal.h` (Example: `SIGSEGV`, `SIGBUS`, etc.) |
| *signal_handler_address* | The address of the signal handler The address might also be displayed in the format, *`library_name+offset`*. This item is output as a hexadecimal number. |
| *signal_mask* | The sa_mask field value in the structure that can be fetched by `sigaction()`. This item is output as a hexadecimal number. |
| *special_flag* | The sa_flags field value in the structure that can be fetched by `sigaction()`. This item is output as a hexadecimal number. |

Thread information

Output format

```
Stack Trace
-----------

"thread_name" #thread_ID daemon prio=priority os_prio=OS_priority jid=hash_value
tid=thread_ID
nid=native ID status [start_address..end_address][CRLF]
    java.lang.Thread.State: current_status_of_thread

    stack=[stack_start_address..YellowPage_address..
RedPage_address..stack_end_address][CRLF]
    [user cpu time=user_timems, kernel cpu time=kernel_timems]
[blocked count=blocked_count, waited count=wait_count][CRLF]
         at class_name.method_name(method_type)
         ...
```

Output items

| **Output items** | **Description** |
|---|---|
| *thread_name* | The thread name specified for a constructor of the Thread class |
| *thread_ID* | A unique number generated when a Java thread is created<br>The value is the same as the value obtained by `java.lang.Thread.getId()`. |
| *daemon* | `daemon` is output for a daemon thread. |
| *priority* | The priority set by Thread#setPriority |
| *OS_priority* | OS level priority. For OSs for which priority cannot be set, `0` is output. |
| *hash_value* | A hexadecimal number of 8 digits. The value is the same as the value obtained by calling `System.identityHashCode()` for a thread object. Note that if a hash value is not assigned to the object, `<N/A>` is output. This item is not output for non-Java threads. |
| *thread_ID* | The address of a thread object in memory |
| *native ID* | Thread ID of the OS level |
| *status* | The status of the thread.<br>One of the following values is output:<br>`runnable`<br>The thread is executable.<br>`waiting on condition`<br>The thread is waiting for `notify`.<br>`waiting for monitor entry`<br>The thread is waiting for acquisition of a synchronized lock.<br>`suspend` or `sleeping`<br>Execution of the thread is suspended. |
| *current_status_of_thread* | A message that indicates the current status of the thread is output. The content of the message corresponds to the `java.lang.Thread.State` enumerated type. This item is not output for non-Java threads. |
| *stack_start_address* | The stack start address. This item is output as a hexadecimal number. This item is not output for non-Java threads. |
| *YellowPage_address* | The first address of the stack Yellow guard page. This item is output as a hexadecimal number. This item is not output for non-Java threads. |
| *RedPage_address* | The first address of the stack Red guard page. This item is output as a hexadecimal number. This item is not output for non-Java threads. |
| *stack_end_address* | The stack end address. This item is output as a hexadecimal number. This item is not output for non-Java threads. |
| *start_address* | The highest stack address for a Java frame This item is output as a hexadecimal number. |
| *end_address* | The highest stack address for a JavaLock This item is output as a hexadecimal number. |
| *user_time* | The user time since the start of the thread (in milliseconds). This item is not output in environments where the time cannot be obtained. This item is also not output for non-Java threads.<br>If an attempt to obtain the user time failed, `unknown` is displayed as follows: |

| Output items | Description |
|---|---|
| | `[user cpu time=unknown, kernel cpu time=`*`kernel_time`*`ms]` |
| *kernel_time* | The kernel time since the start of the thread (in milliseconds). This item is not output in environments where the time cannot be obtained. This item is also not output for non-Java threads. |
| *blocked_count* | The number of times processing was blocked since the start of the thread. This item is not output for non-Java threads. |
| *wait_count* | The number of times processing was suspended since the start of the thread. This item is not output for non-Java threads. |
| *class_name* | Class name. |
| *method_name* | Method name. |
| *method_type* | For native method:<br>`Native Method`<br>If compiled with line numbers for Java methods:<br>*`file_name`*`:`*`line_number`*<br>If compiled without line numbers for Java methods:<br>`Unknown Source` |

## Java monitor dump

### Output format

```
Java monitor
------------
lock_object@hash_code owner_information
   wait_state:number _of_wait_threads
      wait_thread_information
```

### Output items

| Output items | Description |
|---|---|
| *lock_object* | The class name of the object to be locked is output. |
| *hash_code* | The hash code that can be obtained by `Object.hashCode` |
| *owner_information* | If an owner exists:<br>`owner` *`thread_name thread_object_address`*<br>If no owner exists:<br>`no owner` |
| *wait_state* | If waiting for synchronized block or method to be executed:<br>`waiting to enter`<br>If waiting for a notification:<br>`waiting to be notified` |
| *wait_thread_information* | Information is output as follows:<br>*`thread_name thread_object_address`* |

## Raw monitor dump

### Output format

```
Raw monitor dump
----------------
```

```
monitor_name monitor_address owner "thread_name" thread_address
    (The_remaining_part_is_the_same_as_a_Java_monitor_dump)
```

Output items

| Output items | Description |
|---|---|
| *monitor_name* | The name of the monitor to be locked |
| *monitor_address* | The address of the monitor |
| *thread_name* | The name of the thread that acquired the monitor lock |
| *thread_address* | The thread address of the owner thread |

JNI global reference information

Output format

```
JNI Information
---------------
JNI global references:  number_of_JNI_global_references
```

Output items

| Output items | Description |
|---|---|
| *number_of_JNI_global_references* | The number of global references kept by the Java VM is output. |

Explicit heap information (if `-XX:+HitachiUseExplicitMemory` is enabled)

Output format (if the information about the object release ratio for Explicit memory blocks is invalid)

```
Explicit Heap:
 max EH_MAX, total EH_TOTAL, used EH_USED, garbage EH_GARB
(EH_PER1 used/max, EH_PER2
used/total, EH_PER3 garbage/used), EM_NUMS spaces exist
 Explicit Memories(EM_MGR_PTR)#1
  "EM_NAME" eid=EID(EM_PTR)/EM_TYPE, total EM_TOTAL,
used EM_USED, garbage EM_GARB
(EM_PER1 used/total, EM_PER2 garbage/used) EM_STAT#2
    deployed objects
    ________________Size__Instances__Class________________
                 ISIZE      INUM CNAME
                  ...
                AISIZE    AINUM total#3
```

#1:

Explicit heap information is from "`Explicit Heap:`" to "`(`*`EM_MGR_PTR`*`)`".

#2:

Explicit memory block information is from "*EM_NAME*" to the end.

#3

*ISIZE*, *INUM*, *CNAME*, *AISIZE*, and *AINUM* are statistics about objects in Explicit memory blocks.

Notes on the output format:

- The Explicit heap information and the Explicit memory block information are not separated by a blank line, which is different from the extended thread dump information.
- The order in which the Explicit memory block information is output (which memory block item comes first) is not defined.
- Prior to *EM_NAME*, two spaces are inserted.

- There is no blank line at the end. This means that the information output for each Explicit memory block is not separated by a blank line, which is different from the extended thread dump.

Output format (if the information about the object release ratio for Explicit memory blocks is valid)

```
Explicit Heap Status
--------------------
 max EH_MAX, total EH_TOTAL, used EH_USED,
garbage EH_GARB (EH_PER1 used/max, EH_PER2
used/total, EH_PER3 garbage/used), EM_NUMS spaces exist

 Explicit Memories(EM_MGR_PTR)#1

  "EM_NAME" eid=EID(EM_PTR)/EM_TYPE, total EM_TOTAL,
used EM_USED, garbage EM_GARB
(EM_PER1 used/total, EM_PER2 garbage/used, FL_BLOCKS blocks)
EM_STAT#2
    deployed objects
    ________________Size__Instances__FreeRatio__Class________________
                 ISIZE     INUM   FRATIO#4 CNAME
                 ...
                AISIZE    AINUM total#3
```

#1:

Explicit heap information is from "`Explicit Heap:`" to "`(`*EM_MGR_PTR*`)`".

#2:

Explicit memory block information is from "*EM_NAME*" to "*EM_STAT*".

#3:

*ISIZE*, *INUM*, *CNAME*, *AISIZE*, and *AINUM* are statistics about objects in Explicit memory blocks.

#4:

*FRATIO* is the information about the object release ratio for Explicit memory blocks.

Supplementary information about the output format

- For *FRATIO*, the end of the line must be aligned to `o` of `__FreeRatio__`.

Output items

The following shows the output items of the Explicit heap information:

| Output items | Description |
|---|---|
| *EH_MAX* | Maximum Explicit heap size (in kilobytes) |
| *EH_TOTAL* | Size (in kilobytes) of the allocated Explicit heap |
| *EH_USED* | Size (in kilobytes) of the used Explicit heap |
| *EH_GARB* | This output item is used for an extension and indicates the internal status of the Explicit heap. |
| *EH_PER1* | Explicit heap usage (*EH_USED*/*EH_MAX*) |
| *EH_PER2* | Explicit heap usage (*EH_USED*/*EH_TOTAL*) |
| *EH_PER3* | This output item is used for an extension and indicates the internal status of the Explicit heap. |
| *EM_NUMS* | Number of the enabled Explicit memory blocks |
| *EM_MGR_PTR* | The address of the memory that contains internal information about Explicit heap control |

The following describes the output format of Explicit memory block information.

| Output items | Description |
|---|---|
| *EM_NAME* | Name of the Explicit memory block.<br>The output contents are undefined (usually garbled) if the Explicit memory block name contains multibyte characters. `NULL` might be output if the log data was output at about the same time as initialization of the Explicit memory block, or if the Java VM generated the Explicit memory block internally. |
| *EID* | ID of the Explicit memory block |
| *EM_PTR* | The address of the memory that contains the internal structure of an Explicit memory block |
| *EM_TYPE* | ID of the Explicit memory block.<br>One of the following values is output:<br>`R`<br>`B`<br>`A` |
| *EM_TOTAL* | Allocated memory size (in kilobytes) of the Explicit heap block |
| *EM_USED* | Used size (in kilobytes) of the Explicit memory block |
| *EM_GARB* | This output item is used for an extension and indicates the internal status of the Explicit memory block. |
| *EM_PER1* | Usage of the Explicit memory block (*EM_USED*/*EM_TOTAL*) is indicated as a percentage |
| *EM_PER2* | This output item is used for an extension and indicates the internal status of the Explicit memory block. |
| *FL_BLOCKS* | This output item is used for an extension and always shows `0`. |
| *EM_STAT* | Sub status of the Explicit memory block |

The following describes the output format of statistics about objects in Explicit memory blocks.

| Output items | Description |
|---|---|
| *ISIZE* | Size of objects that are instantiated from a class and located in Explicit memory blocks |
| *INUM* | Number of objects that are instantiated from a class and located in Explicit memory blocks |
| *CNAME* | Full class name of the class indicated by *ISIZE* and *INUM* |
| *AISIZE* | Total size of all objects in Explicit memory blocks |
| *AINUM* | Total number of all objects in Explicit memory blocks |

The following describes the output format of the information about the object release ratio for Explicit memory blocks.

| Output items | Description |
|---|---|
| FRATIO | Object release ratio, which is the ratio of objects released by the auto release processing for Explicit memory blocks. The unit is percent.<br>*object_ release_ratio* = (*number_of_objects_in_class_before_auto_release_processing* - *number_of_objects_in_class_after_auto_release_processing*) / *number_of_objects_in_class_before_auto_release_processing* * 100<br>− is output to Explicit memory blocks that were not the targets of the auto release processing when the information about the object release ratio was output. |

Class-separated statistics

Output format

```
Java Heap Profile
------------------------
```

```
________________Size__Instances__Class________________
   total_size       instance_count   class_name
   total_size       instance_count   class_name
...
```

Output items

| Output items | Description |
|---|---|
| *total_size* | The total size of the instances (in bytes) |
| *instance_count* | Number of instances |
| *class_name* | Class name |

Footer

Output format

```
Full thread dump completed.   EEE MMM dd hh:mm:ss yyyy
```

Note:

*EEE* indicates a day of the week. *MMM* indicates a month. *dd* indicates a day.

### Process start log

Output format

This is the same as the message log.

Output items

This is the same as the message log.

## 9.2.3 Output formats of the web server logs

The output formats of the request log and access log of the web server are described below.

### Request log

Output format

```
[time] (server_process_ID) client : hws <-- (client_IP_address:port_number,
server_IP_address:port_number[R])(route_application_information)
```

Output example

```
[Tue May 27 10:10:08.045 2014] (564) client : hws -->
(10.210.185.27:49222,10.210.185.27:80[A])
[Tue May 27 10:10:08.046 2014] (564) client : hws <--
(10.210.185.27:49222,10.210.185.27:80[R])(10.209.15.47/250/0x0000000000000001)
```

### Access log

You can output the route application information to the access log by specifying `%{hws_ap_root}n` for the log format.

The example below shows the access log data that is output when the log format is specified as follows:

```
LogFormat "%h %l %u %t \"%r\" %>s %b %T %{hws_thread_id}P %{hws_ap_root}n"
hws_std
```

Output example

```
10.210.185.27 - - [27/May/2014:10:10:08.045 +0900] "GET /
HTTP/1.0" 200 52 0.001 564 10.209.15.47/250/0x0000000000000001
```

## 9.2.4 Output format of the performance tracer log

Performance tracer messages are output to the `%PRFSPOOL%\log` directory.

The output directory is shown below. The file name is `prf_message` with a number from 01 to 32.

| Output type | Output directory |
|---|---|
| PRF daemon or PRF command<br>PRF output API (CPRF.jar, CPRF.DLL) | `$PRFSPOOL/log/`*`PRF_identifier`* |

The log file to which data is output is switched by using the shift mode. In this mode, log data is always output to the file `prf_message01`. The log files `prf_message02` to `prf_message32` are the shift-destination files that are created when a switch occurs.

When a switch occurs, the file `prf_message01` is renamed by changing the serial number to a number from 02 to 32 (in order) for which no log file exists. For example, when the first switch occurs, the file `prf_message01` is renamed `prf_message02`. If a log file exists for each serial number, when the next switch occurs, the oldest log file (the file that has the oldest update time) is deleted or renamed.

Output format

```
date_and_time:message_ID pid tid:message_text
```

Output items

| Item | Description |
|---|---|
| *date_and_time* | The date and time when the trace was acquired is output in the format *aaa bbb dd hh:mm:ss.sss YYYY*. |
| *message_ID* | The message ID is output. |
| *pid* | The process ID is output. |
| *tid* | The thread ID is output. |
| *message_text* | The message text, which might include a line feed, is output. |

## 9.2.5 Output format of the application development environment log

The output format and output items for the server building error file (`BuildEnvironment.err`) provided by the application development environment is described below.

Log data is output if the creation of a debug environment fails during a new installation.

Output format

```
error_message
end_in_error_message
date_and_time
```

Output items

| Item | Description |
|---|---|
| *error_message* | The error message for the command for which an error occurred is output. |
| *end_in_error_message* | A message indicating that setup ended in an error and the detailed information about the command for which the error occurred are output. |
| *date_and_time* | The date and time when the command ended are output in the format *YYYY/MM/DD hh:mm:ss.ff*. |

## 9.2.6 Output format of the performance analysis trace file

The performance analysis trace collects the trace information for each functionality layer.

Output format

Dump format

```
PRF: record_status Process: process_ID Thread: thread_ID(hash_value)
Trace: trace_serial_number
ProcessName: process_name
Event: event_ID Time: year:month:day hour:minute:second
 millisecond/microsecond/nanosecond
Rc: return_code
ClientAP: IP_address_of_the_client_application
process_ID_of_the_client_application -
communication_number_of_the_client_application
RootAP: IP_address_of_the_root_application
process_ID_of_the_root_application -
communication_number_of_the_root_application
INT: interface_name OPR: operation_name
Offset +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f 0123456789abcdef
  dump_information
```

CSV format

```
PRF,Process,Thread(hashcode),Trace,ProcessName,Event,Date,Time,
Time(msec/usec/nsec),Rc,ClientAP IP,ClientAP PID, ClientAP CommNo.,
RootAP IP,RootAP PID,RootAP CommNo.,INT,OPR,OPT,ASCII
record_status,process_ID,thread_ID(hash_value),trace_serial_number,process_name,
event_ID,year/month/day,hour:minute:second,
millisecond/microsecond/nanosecond,return_code,
IP_address_of_the_client_application,
process_ID_of_the_client_application,
communication_number_of_the_client_application,
IP_address_of_the_root_application,
IP_address_of_the_root_application,
communication_number_of_the_root_application,interface_name,operation_name,
dump_information,ASCII_character_information
```

Output items

| Item | Description |
|---|---|
| PRF | Status of the process record. One of the following is output:<br>`Rec`: The record status is normal.<br>`ErrRec`: The record status is abnormal. |
| Process | ID of the process whose trace information was acquired. A decimal number consisting of 10 or fewer digits is output. |
| Thread | ID or hash value of the thread in the process for which trace information was acquired. A hexadecimal number consisting of a maximum of 18 digits is output. |
| Trace | Trace serial number in the relevant thread. A decimal number consisting of 10 or fewer digits is output. |
| ProcessName | Process name. A character string that consists of 32 or fewer characters and that identifies the process is output. |
| Event | Event ID. A 6-digit hexadecimal number is output. |
| Time | Year, month, day, and time when the trace information was acquired. This information is output in the following format:<br>*`year`*`/`*`month`*`/`*`day hour`*`:`*`minute`*`:`*`second millisecond`*`/`*`microsecond`*`/`*`nanosecond`*<br>Output example<br>`2000/02/12 13:43:44 363/200/000` |
| Rc | Return code. A 16-digit decimal number is output. |
| ClientAP | The following information about the client application is output:<br>IP address<br>Process ID<br>Communication number |
| RootAP | The following information about the root application is output:<br>IP address<br>Process ID<br>Communication number |
| INT | Interface name. A character string# consisting of 33 or fewer characters is output. |
| OPR | Operation name. A character string# consisting of 33 or fewer characters is output. |
| OPT | Information that was acquired for each process. Dump information consisting of a maximum of 514 characters is output. |
| ASCII | Information that was acquired for each process. ASCII character information consisting of a maximum of 514 is output. |

#:

If the length of the interface name or operation name exceeds 33 characters, the name is changed by using one of the following methods so that the output name consists of 33 characters:

*first_32_characters* + *

*first_16_characters* + * + *last_16_characters*

* + *last_32_characters*

## 9.3 Trace collection points

The following explains the trace collection points of each layer and the PRF trace collection level.

### 9.3.1 Trace collection points of performance tracer

The following explains the trace collection points of performance tracer and information that can be collected.

PRF trace refers to the trace information that is output at a predetermined processing point of the Java EE server. It is a binary file that can be output by setting the PRF trace collection level.

To enable a series of processing steps to be traced, consistent keys are set and managed for a series of processing steps of each event unit. Key information is added to the trace that is output at the point (trance collection point) when trace information in the event is output. The table below lists the key information.

Table 9-1: Key information of PRF traces

| Number | Key information of PRF traces | Description |
|---|---|---|
| 1 | Root application information | Information that must be kept as a unique value in the processing sequence across multiple processes<br>Application Server adds root application information at the place identified as the start point of the sequence. For example, if a connection is made to Application Server from an application client, root application information is added when the application client starts. If a connection is made to Application Server from another client, such as a web browser, root application information is added when the request is received by the web server. |
| 2 | Client application information | Information that must be kept as a unique value in the processing sequence between the client and the server<br>Application Server adds client application information at the place identified as the start point of communications between the client and the server.<br>For example, the client application information is added immediately before a communication message is transmitted, such as for HTTP communications for a web service call or for RMI/IIOP communications for an EJB call. |

Because root application information is output to the Web Server log file (request logs and access logs), you can check root application information by comparing the Web Server log file with the performance analysis trace file#.

#:

A file in which a PRF trace is changed to text (CSV) format.

The table below lists the trace collection points in each function layer.

Table 9-2: Trace collection points

| Number | Function layer | Event ID# |
|---|---|---|
| 1 | Web container | From 0xB100 to 0xB101<br>From 0xB200 to 0xB203 |
| 2 | EJB container | From 0xB300 to 0xB307<br>From 0xBB00 to 0xBB03 |

| Number | Function layer | Event ID# |
|---|---|---|
| 3 | JNDI | From 0xB400 to 0xB401 |
| 4 | JTA | From 0xB500 to 0xB507 |
| 5 | JDBC | From 0xB600 to 0xB665 |
| 6 | JSF | From 0xB700 to 0xB70D |
| 7 | JMS | From 0xB800 to 0xB81F |
| 8 | JAX-RS | From 0xB900 to 0xB905 |
| 9 | JAX-WS | From 0xBA00 to 0xBA0B |

#:

PRF trace output point and position information in each process

**Related topics**

- 9.3.2 Trace collection points of the web container
- 9.3.3 Trace collection points of the EJB container
- 9.3.4 Trace collection points of JNDI
- 9.3.5 Trace collection points of JTA
- 9.3.6 Trace collection points of JDBC
- 9.3.7 Trace collection points of JSF
- 9.3.8 Trace collection points of JMS
- 9.3.9 Trace collection points of JAX-RS
- 9.3.10 Trace collection points of JAX-WS
- 9.3.11 Trace collection points of Concurrency Utilities

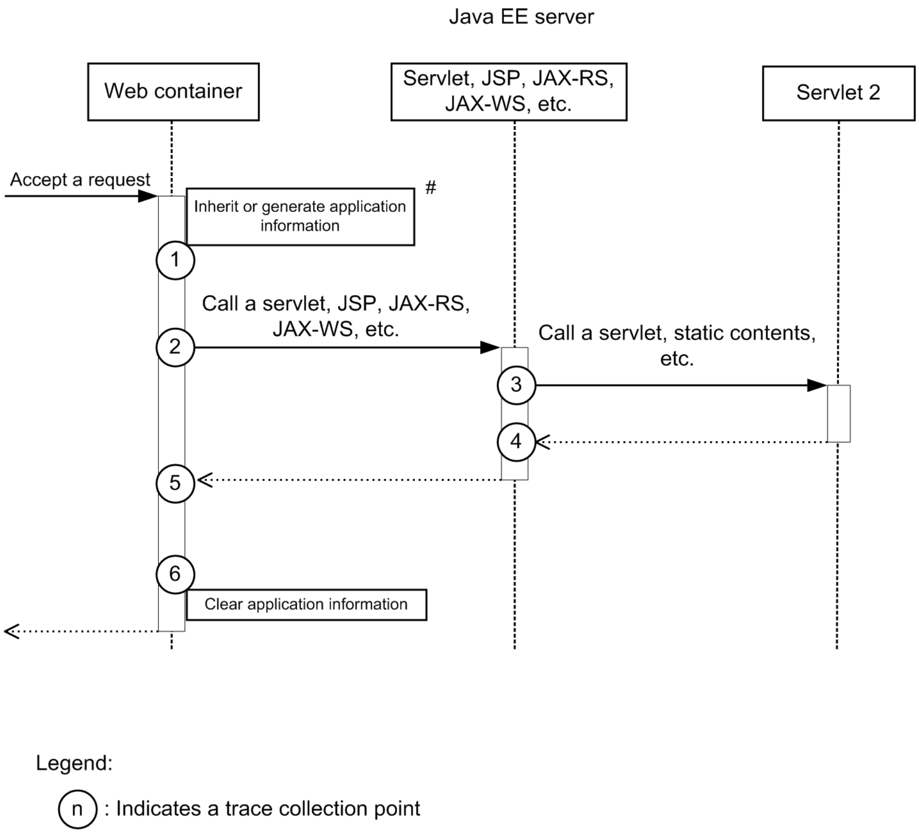
## 9.3.2 Trace collection points of the web container

The following explains the details of the trace collection points of the web container.

### When synchronous processing is used

The following shows the trace collection points of the web container.

## Figure 9-1: Trace collection points of the web container (in synchronous processing)


Java EE server
Web container
Servlet, JSP, JAX-RS, JAX-WS, etc.
Servlet 2
Accept a request
Inherit or generate application information
#
1
Call a servlet, JSP, JAX-RS, JAX-WS, etc.
2
Call a servlet, static contents, etc.
3
4
5
6
Clear application information
Legend:
n : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

#:

If any application information is included in the request header, the application information is inherited.

If no application information is included, application information will be generated.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

## Table 9-3: Details on the trace collection points of the web container (when synchronous processing is used)

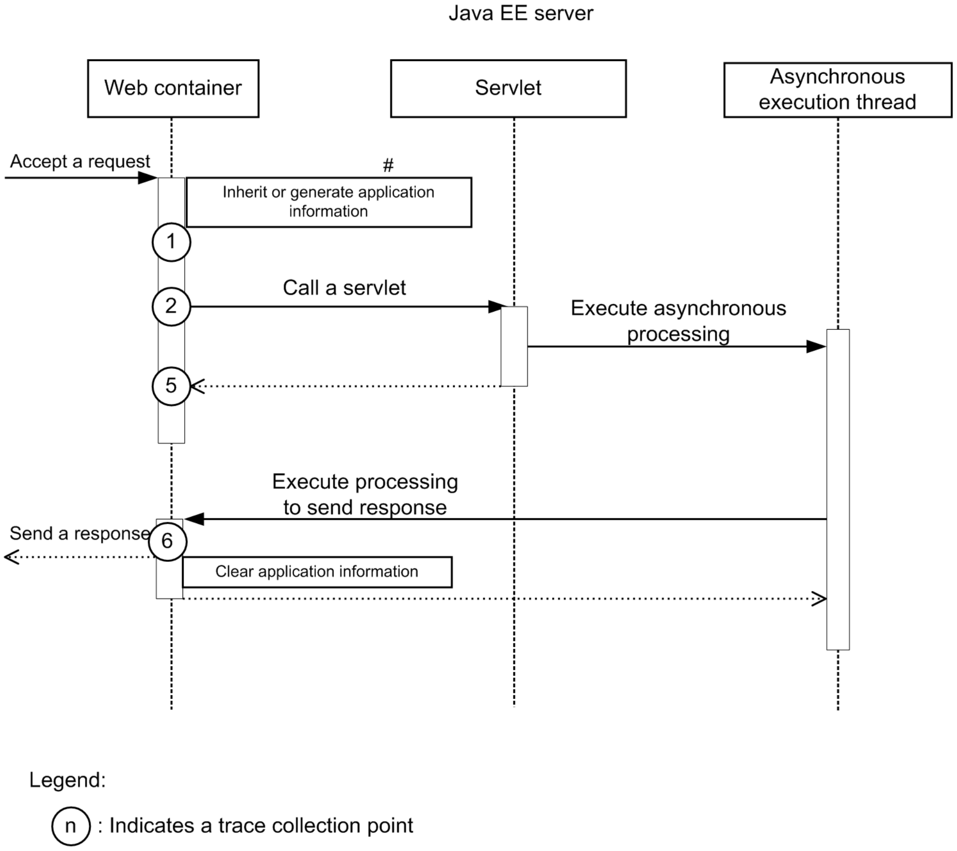
| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB100 | 1 | Standard level | Immediately after the completion of request collection and request header analysis | HTTP method name | URI | - |
| 0xB101 | 6 | Standard level | Immediately after the completion of request processing | HTTP method name | URI | In a normal state<br>Status code |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| | | | | | | In an abnormal state<br>Status code: exception name |
| 0xB200 | 2, 3 | Standard level | Immediately before calling a servlet, JSP, JAX-RS, or JAX-WS | For something other than JSP<br>Class name<br>For JSP<br>JSP file path that begins at the context root | - | Session ID |
| 0xB201 | 4, 5 | Standard level | Immediately after the completion of servlet, JSP, JAX-RS, or JAX-WS processing | For something other than JSP<br>Class name<br>For JSP<br>JSP file path that begins at the context root | - | In a normal state<br>Session ID<br>In an abnormal state<br>Session ID: exception name |
| 0xB202 | 3 | Standard level | Immediately before calling static content (DefaultServlet) | - | Context root name | Session ID |
| 0xB203 | 4 | Standard level | Immediately after the completion of static content processing (DefaultServlet) | - | Context root name | In a normal state<br>Session ID<br>In an abnormal state<br>Session ID: exception name |

## When asynchronous processing is used

The following shows the trace collection points of the web container when responses are sent by asynchronous processing of servlet.

## Figure 9-2: Trace collection points of the web container (when asynchronous processing is used)


Java EE server
Web container
Servlet
Asynchronous execution thread
Accept a request
#
Inherit or generate application information
1
Call a servlet
2
Execute asynchronous processing
5
Execute processing to send response
Send a response
6
Clear application information
Legend:
n : Indicates a trace collection point

**Reference note**

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

#:

If any application information is included in the request header, the application information is inherited.

If no application information is included, application information will be generated.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

## Table 9-4: Details of the trace collection points of the web container (when asynchronous processing is used)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB100 | 1 | Standard level | Immediately after the completion of request collection and request header analysis | HTTP method name | URI | - |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB101 | 6 | Standard level | Immediately after the completion of request processing | HTTP method name | URI | In a normal state<br>Status code<br>In an abnormal state<br>Status code: exception name |
| 0xB200 | 2 | Standard level | Immediately before calling a servlet, JSP, JAX-RS, or JAX-WS | For something other than JSP<br>Class name<br>For JSP<br>JSP file path that begins at the context root | - | Session ID |
| 0xB201 | 5 | Standard level | Immediately after the completion of servlet, JSP, JAX-RS, or JAX-WS processing | For something other than JSP<br>Class name<br>For JSP<br>JSP file path that begins at the context root | - | In a normal state<br>Session ID<br>In an abnormal state<br>Session ID: exception name |

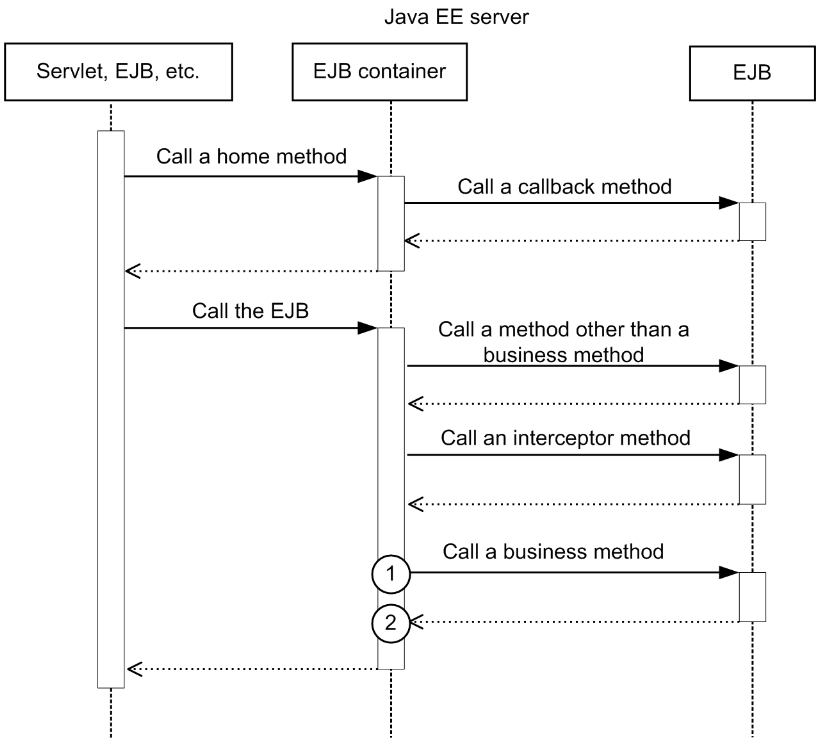
## 9.3.3 Trace collection points of the EJB container

The following explains the details of the trace collection points of the EJB container.

### For Session Bean and Entity Bean (local call)

The following shows the trace collection points for Session Bean and Entity Bean (local call).

## Figure 9-3: Collection points for Session Bean and Entity Bean (local call)



Legend:

(n) : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

## Table 9-5: Details on the trace collection points

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB300 | 1 | Standard level | Immediately before the EJB container calls an EJB business method | Implementation class name of EJB | Method name (number of parameters) | - |
| 0xB301 | 2 | Standard level | Immediately after calling an EJB business method | Implementation class name of EJB | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name[#] |

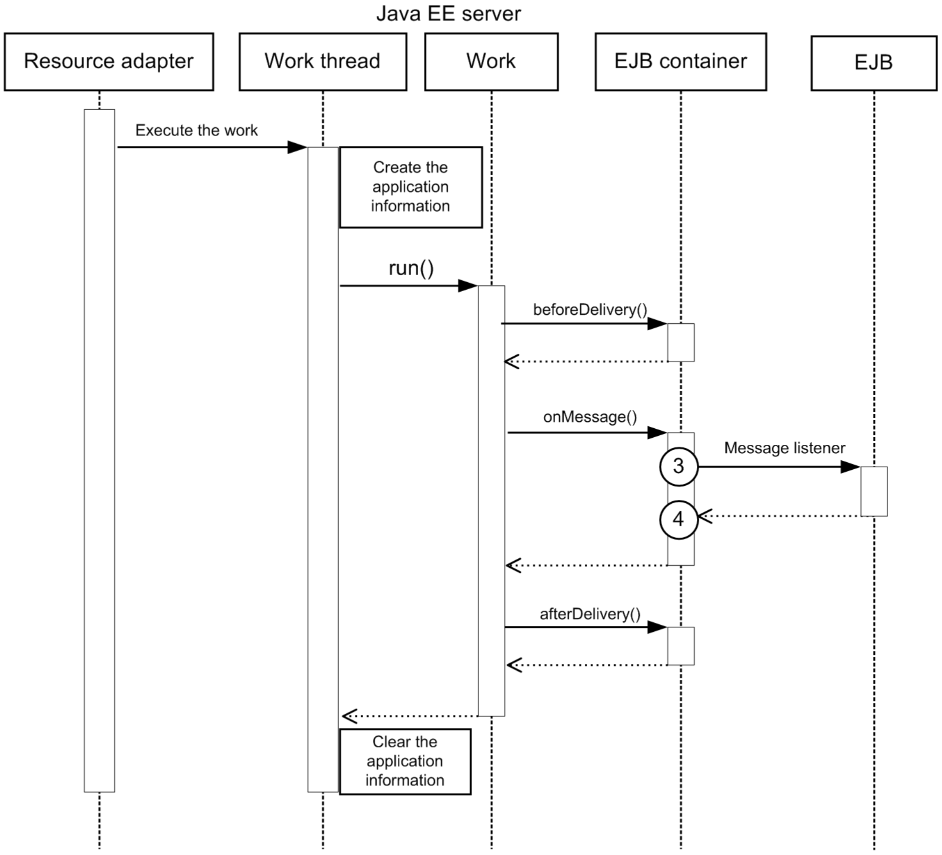
#:

If the exception collected at a trace collection point is `java.lang.reflect.InvocationTargetException`, the name of exception retained by `java.lang.reflect.InvocationTargetException` is output.

## For Message Driven Bean

The following shows the trace collection points of Message Driven Bean.

Figure 9-4: Trace collection points of Message Driven Bean



Legend:

(n) : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-6: Details on the trace collection points of Message Driven Bean

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB302 | 3 | Standard level | Immediately before calling a message listener of Message Driven Bean | Implementation class name of EJB | Method name (number of parameters) | - |
| 0xB303 | 4 | Standard level | Immediately after calling a message listener of Message Driven Bean | Implementation class name of EJB | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name# |

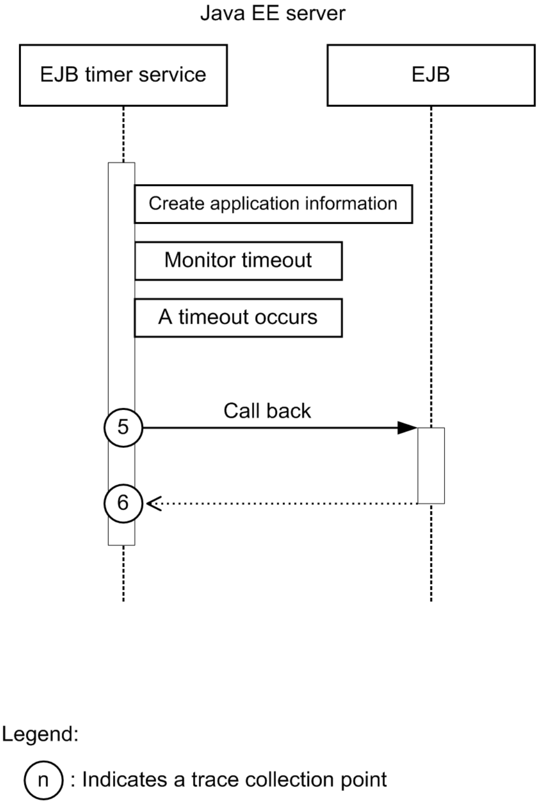
#:

If the exception collected at a trace collection point is `java.lang.reflect.InvocationTargetException`, the name of exception retained by `java.lang.reflect.InvocationTargetException` is output.

## For Timer Service

The following shows the trace collection points of Timer Service.

Figure 9-5: Trace collection points of Timer Service



### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-7: Details on the trace collection points of Timer Service

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB304 | 5 | Standard level | Immediately before calling a callback method of the EJB timer service | Implementation class name of EJB | Method name (number of parameters) | - |
| 0xB305 | 6 | Standard level | Immediately after calling a callback method of the EJB timer service | Implementation class name of EJB | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name# |

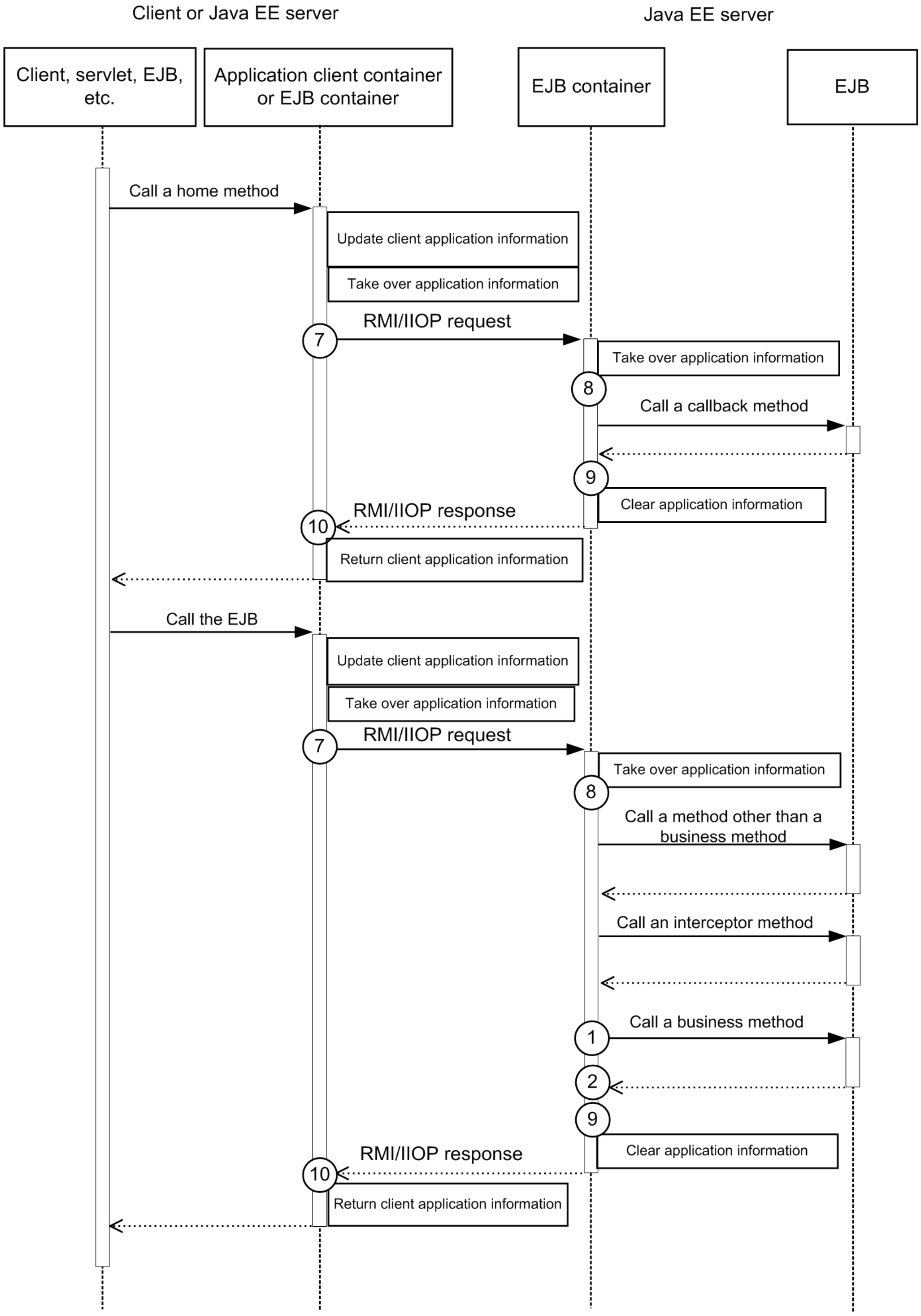
#:

If the exception collected at a trace collection point is `java.lang.reflect.InvocationTargetException`, the name of exception retained by `java.lang.reflect.InvocationTargetException` is output.

## For Session Bean and Entity Bean (remote call)

The following shows the trace collection points for Session Bean and Entity Bean (remote call).

Figure 9-6: Collection points for Session Bean and Entity Bean (remote call)


Client or Java EE server
Java EE server
Client, servlet, EJB, etc.
Application client container or EJB container
EJB container
EJB
Call a home method
Update client application information
Take over application information
RMI/IIOP request
7
Take over application information
8
Call a callback method
9
Clear application information
RMI/IIOP response
10
Return client application information
Call the EJB
Update client application information
Take over application information
RMI/IIOP request
7
Take over application information
8
Call a method other than a business method
Call an interceptor method
Call a business method
1
2
9
Clear application information
RMI/IIOP response
10
Return client application information

## Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-8: Details on the collection points for Session Bean and Entity Bean (remote call)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB300 | 1 | Standard level | Immediately before the EJB container calls an EJB business method | Implementation class name of EJB | Method name (number of parameters) | - |
| 0xB301 | 2 | Standard level | Immediately after calling an EJB business method | Implementation class name of EJB | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name# |
| 0xBB00 | 7 | Standard level | Before sending an RMI/IIOP request | Operation name of RMI/IIOP | Operation name of RMI/IIOP | - |
| 0xBB01 | 10 | Standard level | After receiving an RMI/IIOP response | Interface name of RMI/IIOP | Operation name of RMI/IIOP | - |
| 0xBB02 | 8 | Standard level | After receiving an RMI/IIOP request | Interface name of RMI/IIOP | Operation name of RMI/IIOP | - |
| 0xBB03 | 9 | Standard level | Before sending an RMI/IIOP response | Interface name of RMI/IIOP | Operation name of RMI/IIOP | - |

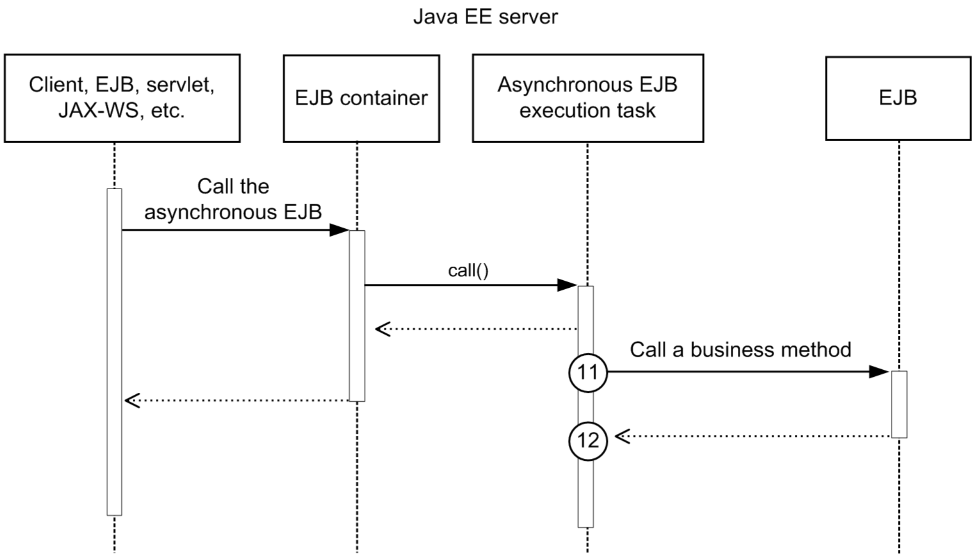
#:

If the exception collected at a trace collection point is `java.lang.reflect.InvocationTargetException`, the name of exception retained by `java.lang.reflect.InvocationTargetException` is output.

## For an asynchronous EJB call

The following shows the trace collection points of an asynchronous EJB call.

## Figure 9-7: Trace collection points of an asynchronous EJB call


Java EE server
Client, EJB, servlet, JAX-WS, etc.
EJB container
Asynchronous EJB execution task
EJB
Call the asynchronous EJB
call()
Call a business method
11
12

Legend:

(n) : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

## Table 9-9: Details on the trace collection points of an asynchronous EJB call

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB306 | 11 | Standard level | Immediately before an asynchronous EJB task calls an asynchronous EJB business method | Implementation class name of EJB | Method name (number of parameters) | - |
| 0xB307 | 12 | Standard level | Immediately after calling an asynchronous EJB business method | Implementation class name of EJB | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name# |

#:

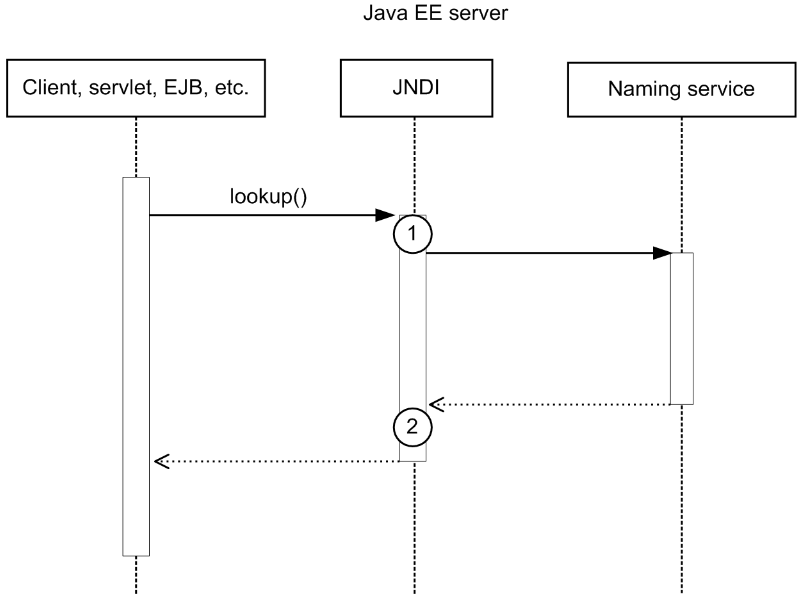
If the exception collected at a trace collection point is `java.lang.reflect.InvocationTargetException`, the name of exception retained by `java.lang.reflect.InvocationTargetException` is output.

## 9.3.4 Trace collection points of JNDI

The following provides details on the trace collection points of JNDI.

The following shows the trace collection points of JNDI

Figure 9-8: Trace collection points of JNDI



Legend:

(n) : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-10: Details on the trace collection points of JNDI

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB400 | 1 | Standard level | Immediately after calling `javax.naming.Context.lookup()` (java: when searching for a namespace) | - | Specified name | - |
| 0xB401 | 2 | Standard level | Immediately before returning | - | Specified name | In a normal state<br>- |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| | | | `javax.naming.Context.lookup()` (java: when searching for a namespace) | | | In an abnormal state<br>Message: exception name |

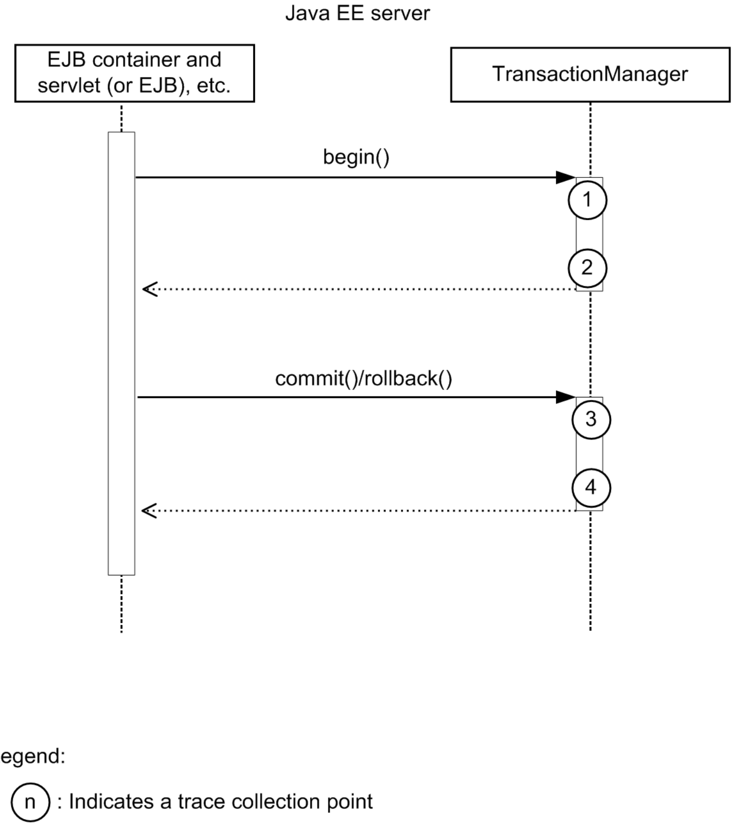
## 9.3.5 Trace collection points of JTA

The following provides details on the trace collection points of JTA.

### When a transaction is explicitly settled

The following shows the trace collection points of JTA when a transaction is explicitly settled.

Figure 9-9: Trace collection points of JTA (when a transaction is explicitly settled)


Java EE server
EJB container and servlet (or EJB), etc.
TransactionManager
begin()
1
2
commit()/rollback()
3
4
Legend:
n : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

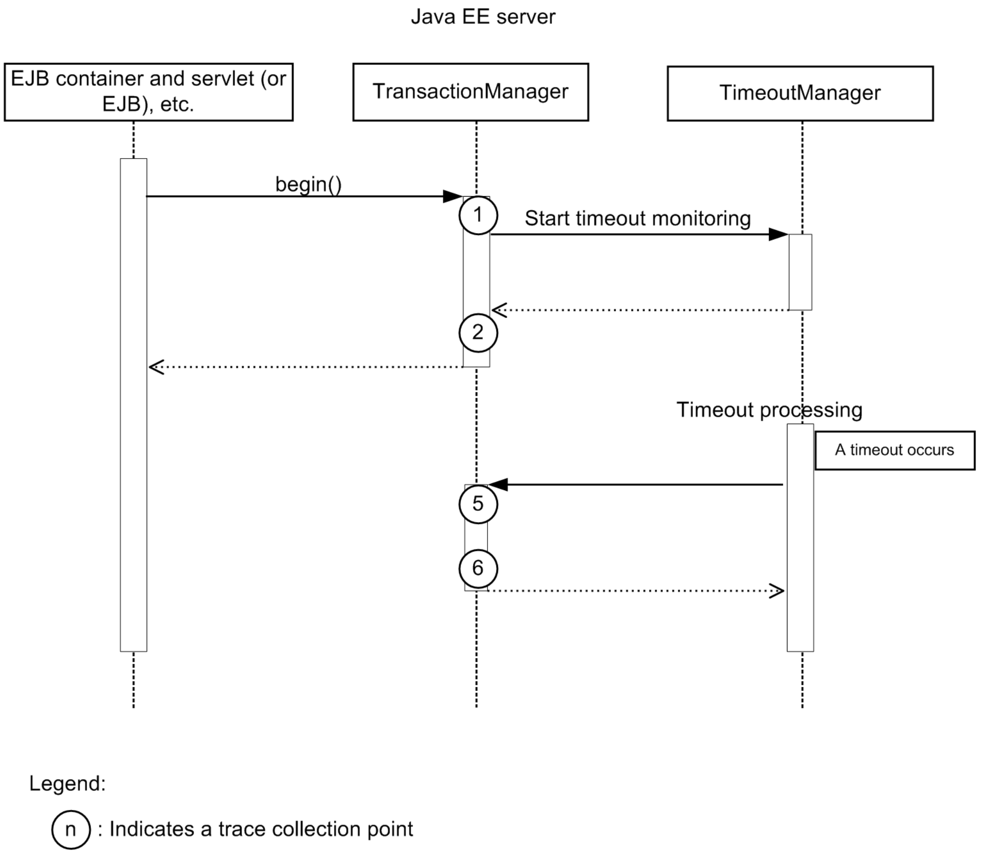
Table 9-11: Details on the trace collection points of JTA (when a transaction is explicitly settled)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB500 | 1 | Standard level | Immediately before the start processing of a transaction | - | - | - |
| 0xB501 | 2 | Standard level | Immediately after the start processing of a transaction | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB502 | 3 | Standard level | Immediately before the commit processing of a transaction | - | - | - |
| 0xB503 | 4 | Standard level | Immediately after the commit processing of a transaction | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB504 | 3 | Standard level | Immediately before the rollback processing of a transaction | - | - | - |
| 0xB505 | 4 | Standard level | Immediately after the rollback processing of a transaction | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When a transaction times out

The following shows the trace collection points of JTA when a transaction times out.

## Figure 9-10: Trace collection points of JTA (when a transaction times out)



### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

## Table 9-12: Details on the trace collection points of JTA (when a transaction times out)

| Event ID | Number in the figure | Trace level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB500 | 1 | Standard level | Immediately before the start processing of a transaction | - | - | - |
| 0xB501 | 2 | Standard level | Immediately after the start processing of a transaction | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB506 | 5 | Standard level | Immediately before the timeout processing of a transaction | - | - | - |

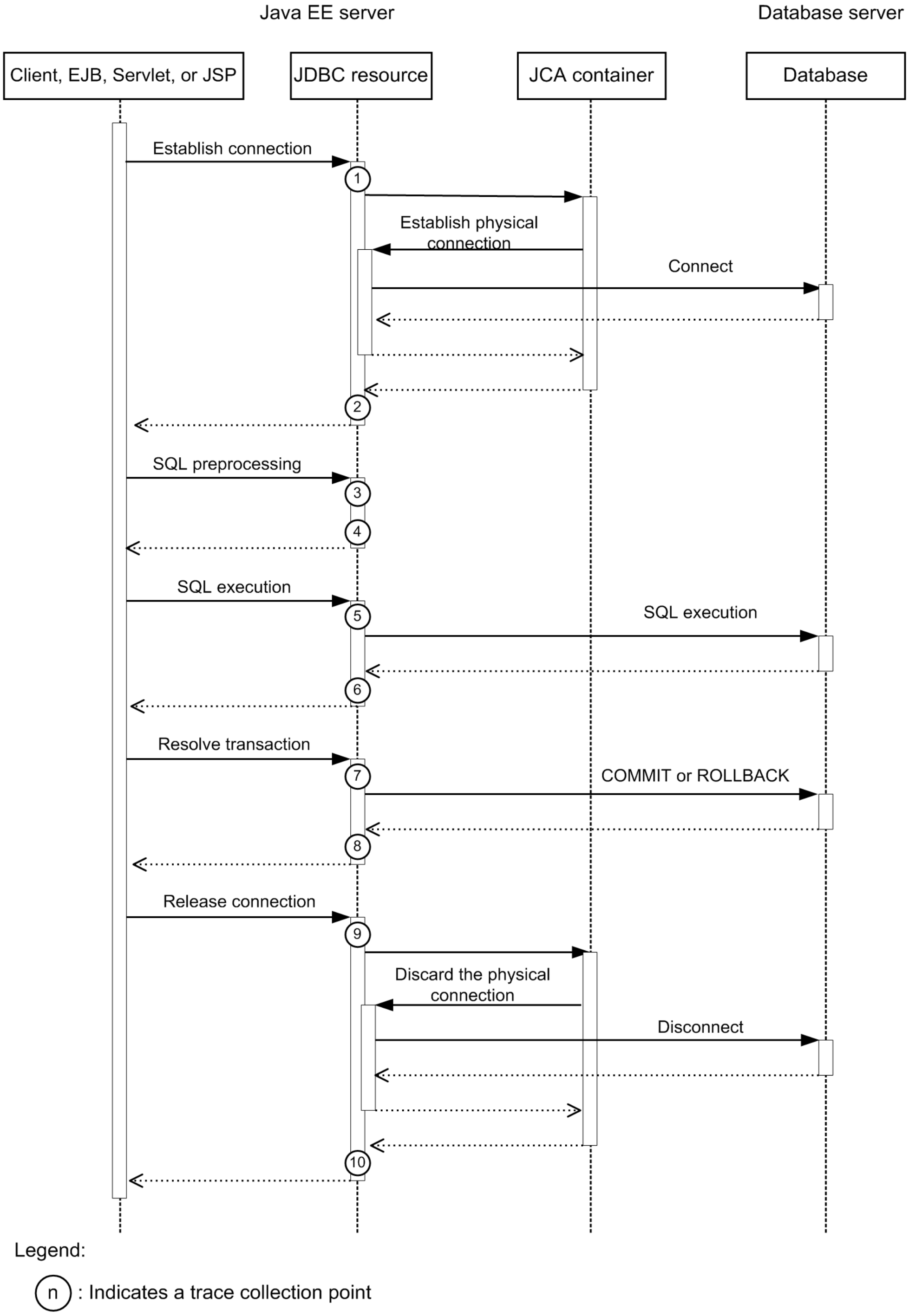
| Event ID | Number in the figure | Trace level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB507 | 6 | Standard level | Immediately after the timeout processing of a transaction | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## 9.3.6 Trace collection points of JDBC

The following provides details on the trace collection points of JDBC.

The figure below shows the trace collection points of JDBC.

Figure 9-11: Trace collection points of JDBC



## Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces Other triggers for collecting traces exist.

The table below lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-13: Details on the trace collection points of JDBC

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB600 | 1 | Standard level | Start of the processing to establish database connection by using `javax.sql.DataSource.getConnection()` | - | - | - |
| 0xB601 | 2 | Standard level | End of the processing to establish database connection by using `javax.sql.DataSource.getConnection()` | - | - | In a normal state<br>Connection ID<br>In an abnormal state<br>Exception name |
| 0xB602 | 1 | Standard level | Start of the processing to establish database connection by using `javax.sql.DataSource.getConnection`(String username, String password) | - | - | - |
| 0xB603 | 2 | Standard level | End of the processing to establish database connection by using `javax.sql.DataSource.getConnection`(String username, String password) | - | - | In a normal state<br>Connection ID<br>In an abnormal state<br>Exception name |
| 0xB604 | 9 | Standard level | Start of the processing to release the Connection object's database and JDBC resources by using `java.sql.Connection.close()` | - | - | Connection ID |
| 0xB605 | 10 | Standard level | End of the processing to release the Connection object's database and JDBC resources by using `java.sql.Connection.close()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB606 | 7 | Standard level | Start of the processing for `java.sql.Connection.commit()` | - | - | - |
| 0xB607 | 8 | Standard level | End of the processing for `java.sql.Connection.commit()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB608 | 7 | Standard level | Start of the processing for `java.sql.Connection.rollback()` | - | - | - |
| 0xB609 | 8 | Standard level | End of the processing for `java.sql.Connection.rollback()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB60A | 7 | Standard level | Start of the processing for `java.sql.Connection.rollback`(Savepoint savepoint) | - | - | - |
| 0xB60B | 8 | Standard level | End of the processing for `java.sql.Connection.rollback`(Savepoint savepoint) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB60C | 3 | Standard level | Start of the processing for `Connection.prepareCall`(String sql) | - | - | SQL statement |
| 0xB60D | 4 | Standard level | End of the processing for `Connection.prepareCall`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB60E | 3 | Standard level | Start of the processing for `Connection.prepareCall`(String sql, int resultSetType, int resultSetConcurrency) | - | - | SQL statement |
| 0xB60F | 3 | Standard level | End of the processing for `Connection.prepareCall`(String sql, int resultSetType, int resultSetConcurrency) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB610 | 4 | Standard level | Start of the processing for `Connection.prepareCall`(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability) | - | - | SQL statement |
| 0xB611 | 3 | Standard level | End of the processing for `Connection.prepareCall`(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB612 | 3 | Standard level | Start of the processing for `Connection.prepareStatement`(String sql) | - | - | SQL statement |
| 0xB613 | 4 | Standard level | End of the processing for `Connection.prepareStatement`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB614 | 3 | Standard level | Start of the processing for `Connection.prepareStatement`(String sql, int autoGeneratedKeys) | - | - | SQL statement |
| 0xB615 | 4 | Standard level | End of the processing for `Connection.prepareSta` | - | - | In a normal state<br>- |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| | | | `tement`(String sql, int autoGeneratedKeys) | | | In an abnormal state<br>Exception name |
| 0xB616 | 3 | Standard level | Start of the processing for `Connection.prepareStatement`(String sql, int[] columnIndexes) | - | - | SQL statement |
| 0xB617 | 4 | Standard level | End of the processing for `Connection.prepareStatement`(String sql, int[] columnIndexes) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB618 | 3 | Standard level | Start of the processing for `Connection.prepareStatement`(String sql, int resultSetType, int resultSetConcurrency) | - | - | SQL statement |
| 0xB619 | 4 | Standard level | End of the processing for `Connection.prepareStatement`(String sql, int resultSetType, int resultSetConcurrency) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB61A | 3 | Standard level | Start of the processing for `Connection.prepareStatement`(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability) | - | - | SQL statement |
| 0xB61B | 4 | Standard level | End of the processing for `Connection.prepareStatement`(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB61C | 3 | Standard level | Start of the processing for `Connection.prepareStatement`(String sql, String[] columnNames) | - | - | SQL statement |
| 0xB61D | 4 | Standard level | End of the processing for `Connection.prepareStatement`(String sql, String[] columnNames) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB61E | 5 | Standard level | Start of the processing for `Statement.execute`(String sql) | - | - | SQL statement |
| 0xB61F | 6 | Standard level | End of the processing for `Statement.execute`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB620 | 5 | Standard level | Start of the processing for `Statement.execute`(String sql, int autoGeneratedKeys) | - | - | SQL statement |
| 0xB621 | 6 | Standard level | End of the processing for `Statement.execute`(String sql, int autoGeneratedKeys) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB622 | 5 | Standard level | Start of the processing for `Statement.execute`(String sql, int[] columnIndexes) | - | - | SQL statement |
| 0xB623 | 6 | Standard level | End of the processing for `Statement.execute`(String sql, int[] columnIndexes) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB624 | 5 | Standard level | Start of the processing for `Statement.execute`(String sql, String[] columnNames) | - | - | SQL statement |
| 0xB625 | 6 | Standard level | End of the processing for `Statement.execute`(String sql, String[] columnNames) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB626 | 5 | Standard level | Start of the processing for `Statement.executeBatch()` | - | - | - |
| 0xB627 | 6 | Standard level | End of the processing for `Statement.executeBatch()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB628 | 5 | Standard level | Start of the processing for `Statement.executeQuery`(String sql) | - | - | SQL statement |
| 0xB629 | 6 | Standard level | End of the processing for `Statement.executeQuery`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB62A | 5 | Standard level | Start of the processing for `Statement.executeUpdate`(String sql) | - | - | SQL statement |
| 0xB62B | 6 | Standard level | End of the processing for `Statement.executeUpdate`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB62C | 5 | Standard level | Start of the processing for `Statement.executeUpda` | - | - | SQL statement |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| | | | te(String sql, int autoGeneratedKeys) | | | |
| 0xB62D | 6 | Standard level | End of the processing for Statement.executeUpdate(String sql, int autoGeneratedKeys) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB62E | 5 | Standard level | Start of the processing for Statement.executeUpdate(String sql, int[] columnIndexes) | - | - | SQL statement |
| 0xB62F | 6 | Standard level | End of the processing for Statement.executeUpdate(String sql, int[] columnIndexes) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB630 | 5 | Standard level | Start of the processing for Statement.executeUpdate(String sql, String[] columnNames) | - | - | SQL statement |
| 0xB631 | 6 | Standard level | End of the processing for Statement.executeUpdate(String sql, String[] columnNames) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB632 | 5 | Standard level | Start of the processing for PreparedStatement.execute() | - | - | - |
| 0xB633 | 6 | Standard level | End of the processing for PreparedStatement.execute() | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB634 | 5 | Standard level | Start of the processing for PreparedStatement.execute(String sql) | - | - | SQL statement |
| 0xB635 | 6 | Standard level | End of the processing for PreparedStatement.execute(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB636 | 5 | Standard level | Start of the processing for PreparedStatement.execute(String sql, int autoGeneratedKeys) | - | - | SQL statement |
| 0xB637 | 6 | Standard level | End of the processing for PreparedStatement.execute(String sql, int autoGeneratedKeys) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB638 | 5 | Standard level | Start of the processing for `PreparedStatement.execute`(String sql, int[] columnIndexes) | - | - | SQL statement |
| 0xB639 | 6 | Standard level | End of the processing for `PreparedStatement.execute`(String sql, int[] columnIndexes) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB63A | 5 | Standard level | Start of the processing for `PreparedStatement.execute`(String sql, String[] columnNames) | - | - | SQL statement |
| 0xB63B | 6 | Standard level | End of the processing for `PreparedStatement.execute`(String sql, String[] columnNames) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB63C | 5 | Standard level | Start of the processing for `PreparedStatement.executeBatch()` | - | - | - |
| 0xB63D | 6 | Standard level | End of the processing for `PreparedStatement.executeBatch()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB63E | 5 | Standard level | Start of the processing for `PreparedStatement.executeQuery()` | - | - | - |
| 0xB63F | 6 | Standard level | End of the processing for `PreparedStatement.executeQuery()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB640 | 5 | Standard level | Start of the processing for `PreparedStatement.executeQuery`(String sql) | - | - | SQL statement |
| 0xB641 | 6 | Standard level | End of the processing for `PreparedStatement.executeQuery`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB642 | 5 | Standard level | Start of the processing for `PreparedStatement.executeUpdate()` | - | - | - |
| 0xB643 | 6 | Standard level | End of the processing for `PreparedStatement.executeUpdate()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB644 | 5 | Standard level | Start of the processing for `PreparedStatement.executeUpdate`(String sql) | - | - | SQL statement |
| 0xB645 | 6 | Standard level | End of the processing for `PreparedStatement.executeUpdate`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB646 | 5 | Standard level | Start of the processing for `PreparedStatement.executeUpdate`(String sql, int autoGeneratedKeys) | - | - | SQL statement |
| 0xB647 | 6 | Standard level | End of the processing for `PreparedStatement.executeUpdate`(String sql, int autoGeneratedKeys) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB648 | 5 | Standard level | Start of the processing for `PreparedStatement.executeUpdate`(String sql, int[] columnIndexes) | - | - | SQL statement |
| 0xB649 | 6 | Standard level | End of the processing for `PreparedStatement.executeUpdate`(String sql, int[] columnIndexes) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB64A | 5 | Standard level | Start of the processing for `PreparedStatement.executeUpdate`(String sql, String[] columnNames) | - | - | SQL statement |
| 0xB64B | 6 | Standard level | End of the processing for `PreparedStatement.executeUpdate`(String sql, String[] columnNames) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB64C | 5 | Standard level | Start of the processing for `CallableStatement.execute()` | - | - | - |
| 0xB64D | 6 | Standard level | End of the processing for `CallableStatement.execute()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB64E | 5 | Standard level | Start of the processing for `CallableStatement.execute`(String sql) | - | - | SQL statement |
| 0xB64F | 6 | Standard level | End of the processing for `CallableStatement.execute`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB650 | 5 | Standard level | Start of the processing for `CallableStatement.execute`(String sql, int autoGeneratedKeys) | - | - | SQL statement |
| 0xB651 | 6 | Standard level | End of the processing for `CallableStatement.execute`(String sql, int autoGeneratedKeys) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB652 | 5 | Standard level | Start of the processing for `CallableStatement.execute`(String sql, int[] columnIndexes) | - | - | SQL statement |
| 0xB653 | 6 | Standard level | End of the processing for `CallableStatement.execute`(String sql, int[] columnIndexes) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB654 | 5 | Standard level | Start of the processing for `CallableStatement.execute`(String sql, String[] columnNames) | - | - | SQL statement |
| 0xB655 | 6 | Standard level | End of the processing for `CallableStatement.execute`(String sql, String[] columnNames) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB656 | 5 | Standard level | Start of the processing for `CallableStatement.executeBatch()` | - | - | - |
| 0xB657 | 6 | Standard level | End of the processing for `CallableStatement.executeBatch()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB658 | 5 | Standard level | Start of the processing for `CallableStatement.executeQuery()` | - | - | - |
| 0xB659 | 6 | Standard level | End of the processing for `CallableStatement.executeQuery()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB65A | 5 | Standard level | Start of the processing for `CallableStatement.executeQuery`(String sql) | - | - | SQL statement |
| 0xB65B | 6 | Standard level | End of the processing for `CallableStatement.executeQuery`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB65C | 5 | Standard level | Start of the processing for `CallableStatement.executeUpdate()` | - | - | - |
| 0xB65D | 6 | Standard level | End of the processing for `CallableStatement.executeUpdate()` | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB65E | 5 | Standard level | Start of the processing for `CallableStatement.executeUpdate`(String sql) | - | - | SQL statement |
| 0xB65F | 6 | Standard level | End of the processing for `CallableStatement.executeUpdate`(String sql) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB660 | 5 | Standard level | Start of the processing for `CallableStatement.executeUpdate`(String sql, int autoGeneratedKeys) | - | - | SQL statement |
| 0xB661 | 6 | Standard level | End of the processing for `CallableStatement.executeUpdate`(String sql, int autoGeneratedKeys) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB662 | 5 | Standard level | Start of the processing for `CallableStatement.executeUpdate`(String sql, int[] columnIndexes) | - | - | SQL statement |
| 0xB663 | 6 | Standard level | End of the processing for `CallableStatement.executeUpdate`(String sql, int[] columnIndexes) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB664 | 5 | Standard level | Start of the processing for `CallableStatement.executeUpdate`(String sql, String[] columnNames) | - | - | SQL statement |
| 0xB665 | 6 | Standard level | End of the processing for `CallableStatement.executeUpdate`(String sql, String[] columnNames) | - | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

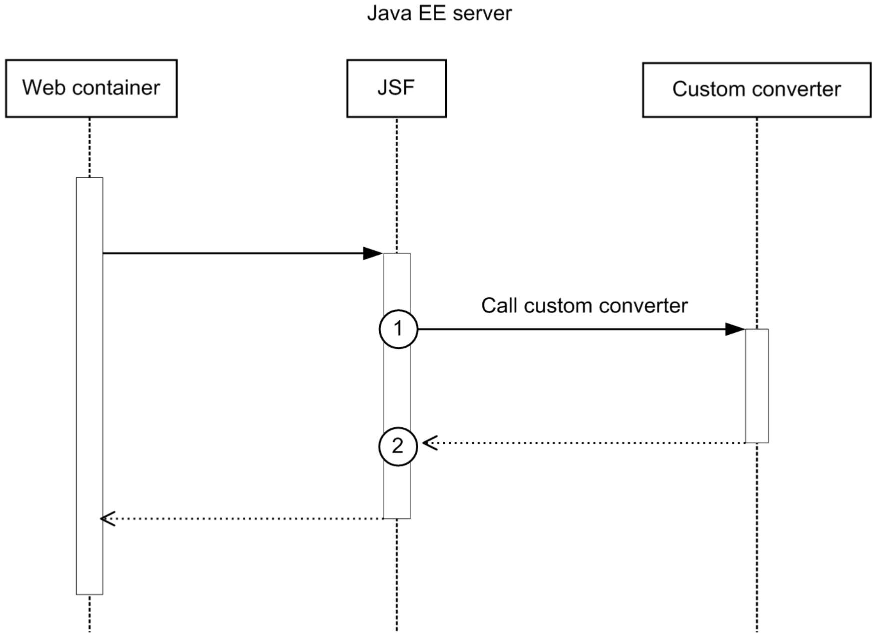
## 9.3.7 Trace collection points of JSF

The following provides details on the trace collection points of JSF.

## When a custom converter is called

The following shows the trace collection points of JSF when a custom converter is called.

Figure 9-12: Trace collection points of JSF (when a custom converter is called)



Legend:

(n) : Indicates a trace collection point

> **Reference note**
>
> This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

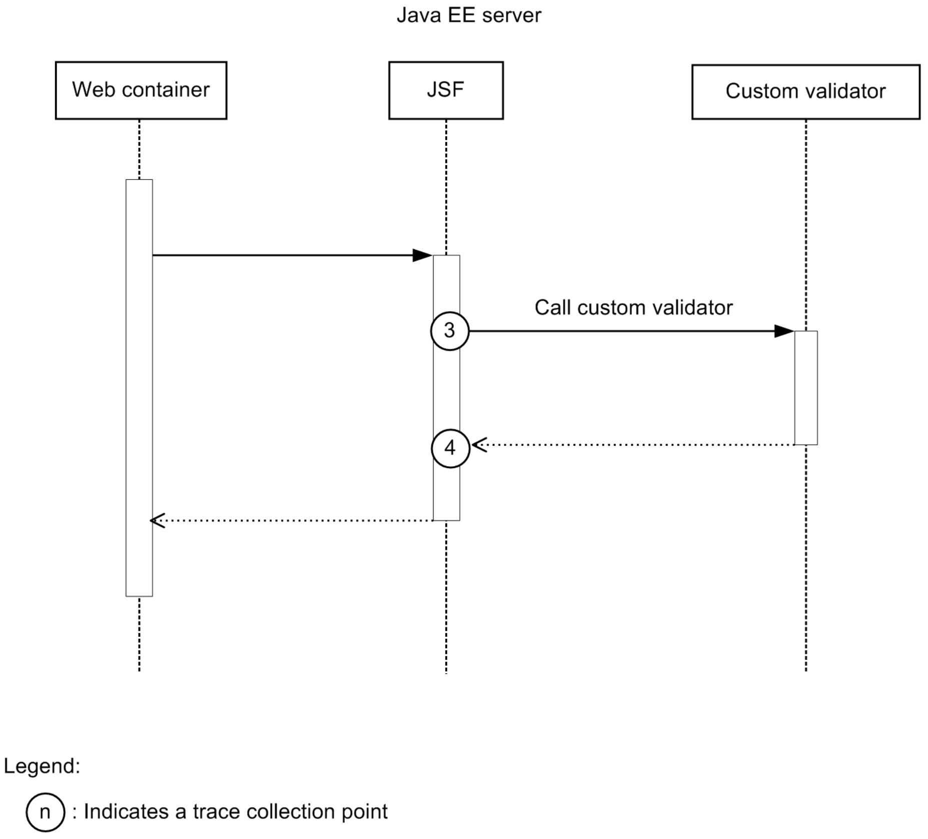
Table 9-14: Details on the trace collection points of JSF (when a custom converter is called)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB700 | 1 | Standard level | Immediately before a custom converter is called | Client ID | The class name and method name of the custom converter are collected. | - |
| 0xB701 | 2 | Standard level | Immediately after the processing of the custom converter ends | Client ID | The class name and method name of the custom converter are collected. | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When a custom validator is called

The following shows the trace collection points of JSF when a custom validator is called.

Figure 9-13: Trace collection points of JSF (when a custom validator is called)



### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-15: Details on the trace collection points of JSF (when a custom validator is called)

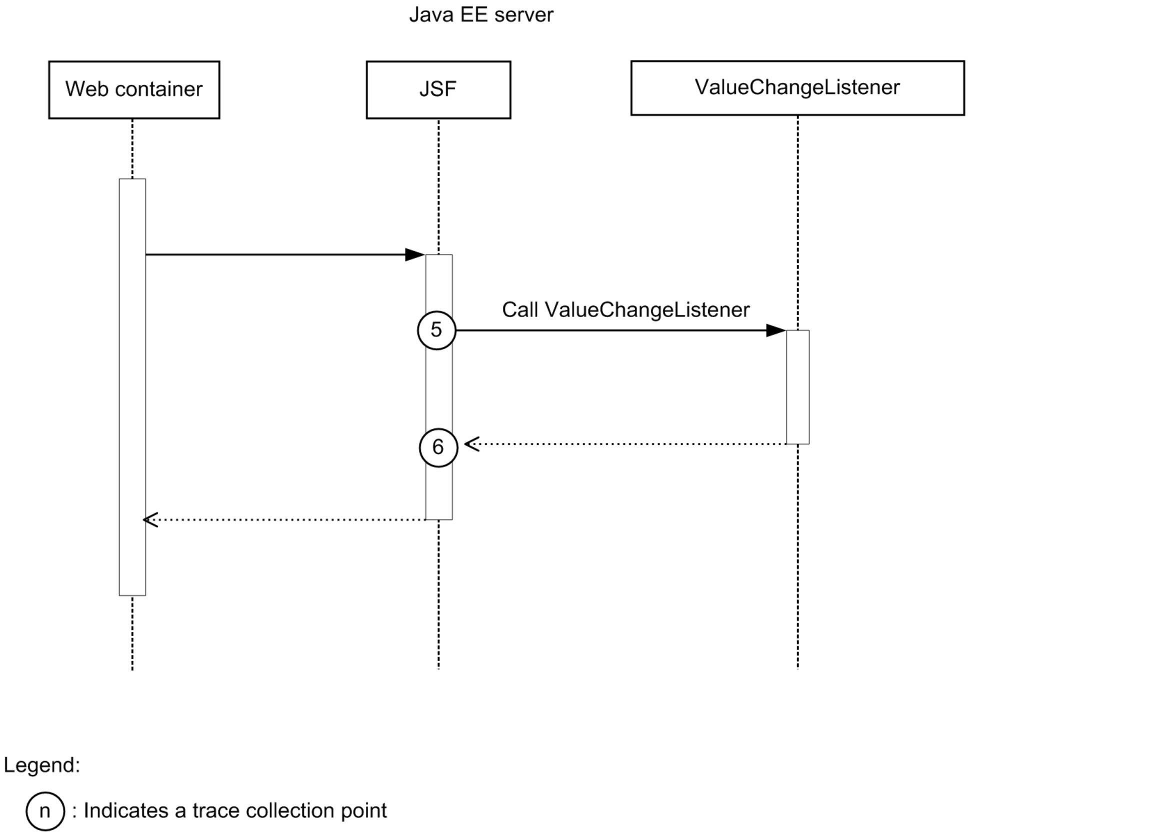
| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB702 | 3 | Standard level | Immediately before a custom validator is called | Client ID | The class name of the custom validator is collected. If MethodExpression was used to call the custom validator, MethodExpression is collected. | - |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB703 | 4 | Standard level | Immediately after the processing of the custom validator ends | Client ID | The class name of the custom validator is collected. If MethodExpression was used to call the custom validator, MethodExpression is collected. | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When ValueChangeListener is called

The following shows the trace collection points of JSF when ValueChangeListener is called.

Figure 9-14: Trace collection points of JSF (when ValueChangeListener is called)



### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

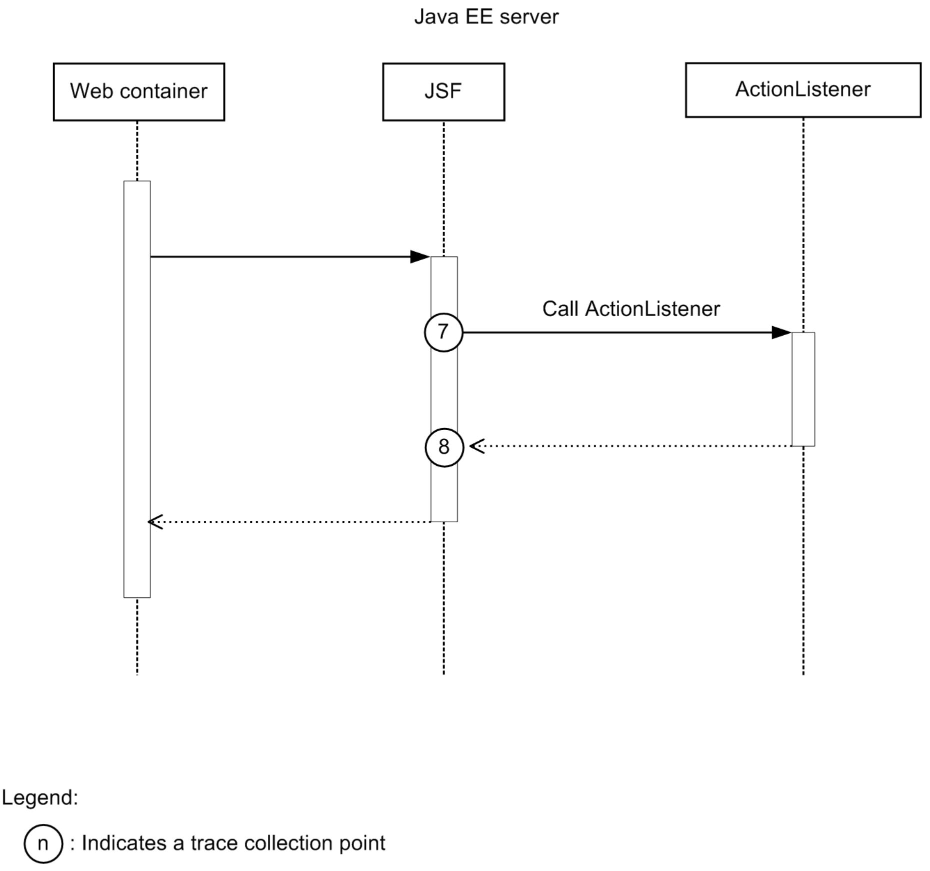
Table 9-16: Details on the trace collection points of JSF (when ValueChangeListener is called)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB704 | 5 | Standard level | Immediately before ValueChangeListener is called | Client ID | The class name of ValueChangeListener is collected.<br>If MethodExpression was used to call ValueChangeListener, MethodExpression is collected. | - |
| 0xB705 | 6 | Standard level | Immediately after the processing of ValueChangeListener ends | Client ID | In a normal state<br>When ValueChangeListener is called by MethodExpression and ends normally, if a method with one parameter is called, `one argument` is output. If a method with no parameter is called, `no argument` is output.<br>In an abnormal state<br>- | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When ActionListener is called

The following shows the trace collection points of JSF when ActionListener is called.

## Figure 9-15: Trace collection points of JSF (when ActionListener is called)



> **Reference note**
>
> This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

## Table 9-17: Details on the trace collection points of JSF (when ActionListener is called)

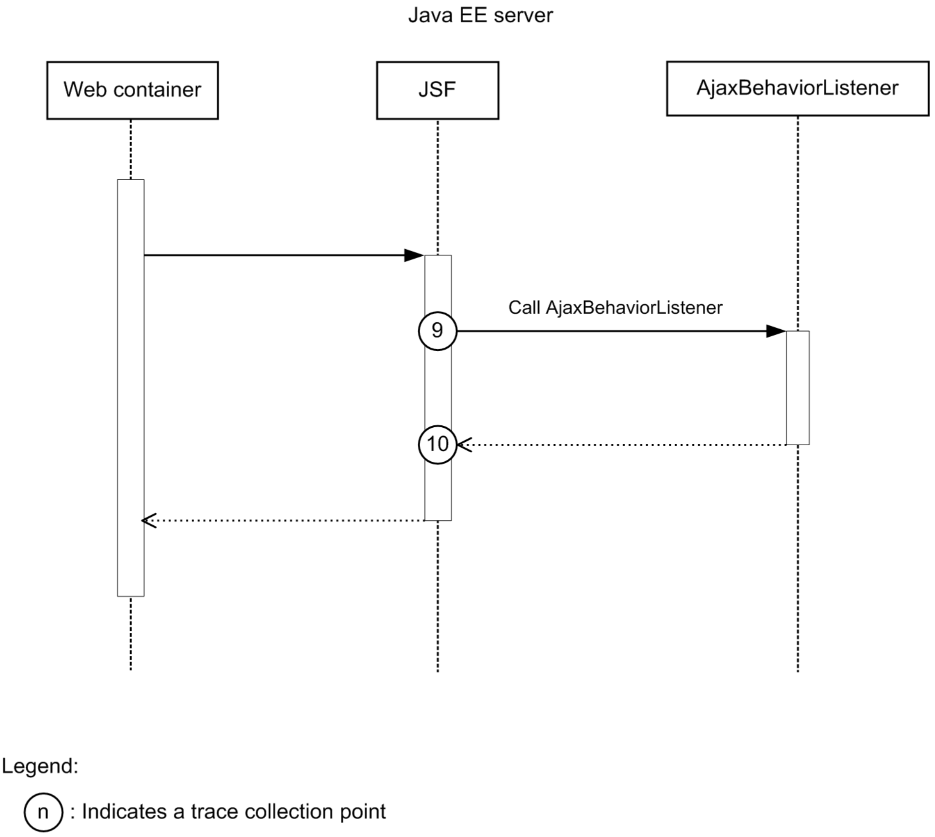
| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB706 | 7 | Standard level | Immediately before ActionListener is called | Client ID | The class name of ActionListener is collected.<br>If MethodExpression was used to call ActionListener, MethodExpression is collected. | - |
| 0xB707 | 8 | Standard level | Immediately after the processing of ActionListener ends | Client ID | In a normal state<br>When ActionListener is called by MethodExpression | In a normal state<br>- |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| | | | | | and ends normally, if a method with one parameter is called, `one argument` is output. If a method with no parameter is called, `no argument` is output.<br>In an abnormal state<br>- | In an abnormal state<br>Exception name |

## When AjaxBehaviorListener is called

The following shows the trace collection points of JSF when AjaxBehaviorListener is called.

Figure 9-16: Trace collection points of JSF (when AjaxBehaviorListener is called)


Java EE server
Web container
JSF
AjaxBehaviorListener
Call AjaxBehaviorListener
9
10
Legend:
n : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

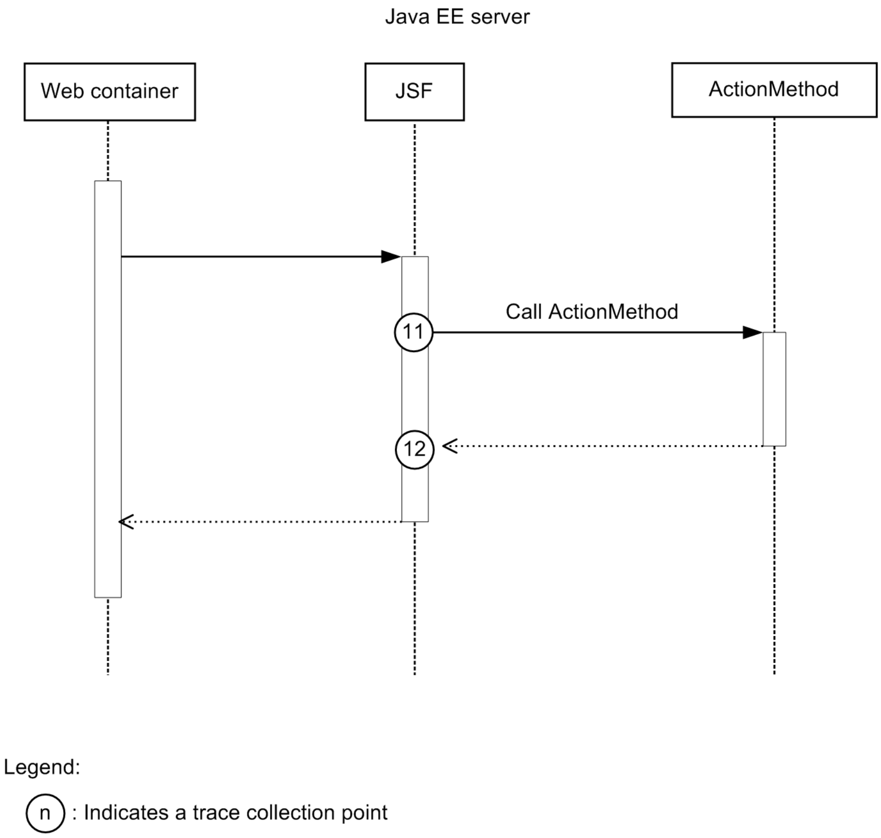
Table 9-18: Details on the trace collection points of JSF (when AjaxBehaviorListener is called)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB708 | 9 | PRF trace collection level | Immediately before AjaxBehaviorListener is called | Client ID | The class name of AjaxBehaviorListener is collected.<br>If MethodExpression was used to call AjaxBehaviorListener, MethodExpression is collected. | - |
| 0xB709 | 10 | Standard level | Immediately after the processing of AjaxBehaviorListener ends | Client ID | In a normal state<br>When AjaxBehaviorListener is called by MethodExpression and ends normally, if a method with one parameter is called, `one argument` is output. If a method with no parameter is called, `no argument` is output.<br>In an abnormal state<br>- | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When ActionMethod is called

The following shows the trace collection points of JSF when ActionMethod is called.

Figure 9-17: Trace collection points of JSF (when ActionMethod is called)



## Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

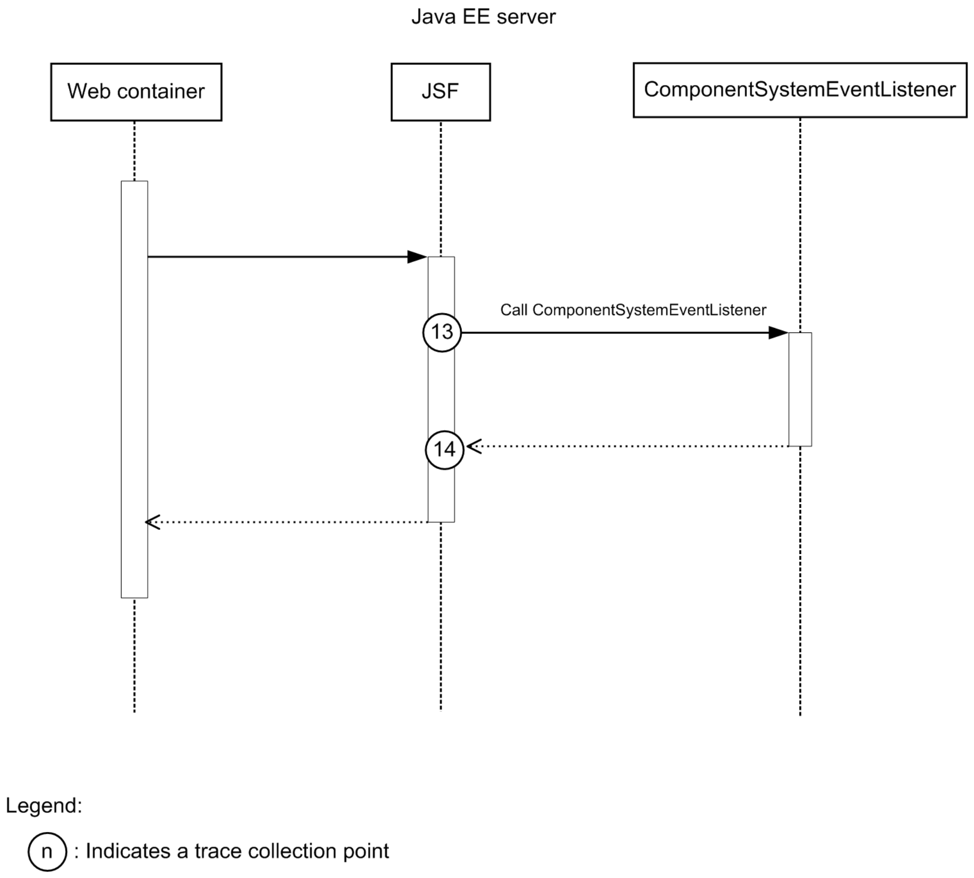
Table 9-19: Details on the trace collection points of JSF (when ActionMethod is called)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB70A | 11 | Standard level | Immediately before ActionMethod is called | Client ID | - | - |
| 0xB70B | 12 | Standard level | Immediately after the processing of ActionMethod ends | Client ID | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When ComponentSystemEventListener is called

The following shows the trace collection points of JSF when ComponentSystemEventListener is called.

## Figure 9-18: Trace collection points of JSF (when ComponentSystemEventListener is called)


Java EE server
Web container
JSF
ComponentSystemEventListener
Call ComponentSystemEventListener
13
14
Legend:
n : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

## Table 9-20: Details on the trace collection points of JSF (when ComponentSystemEventListener is called)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB70C | 13 | Standard level | Immediately before ComponentSystemEvent Listener is called | Client ID | The class name of ComponentSystemEvent Listener is collected.<br>If MethodExpression was used to call ComponentSystemEvent Listener, MethodExpression is collected. | - |

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB70D | 14 | Standard level | Immediately after the processing of ComponentSystemEventListener ends | Client ID | In a normal state<br>When ComponentSystemEventListener is called by MethodExpression and ends normally, if a method with one parameter is called, `one argument` is output. If a method with no parameter is called, `no argument` is output.<br>In an abnormal state<br>- | In a normal state<br>-<br>In an abnormal state<br>Exception name |

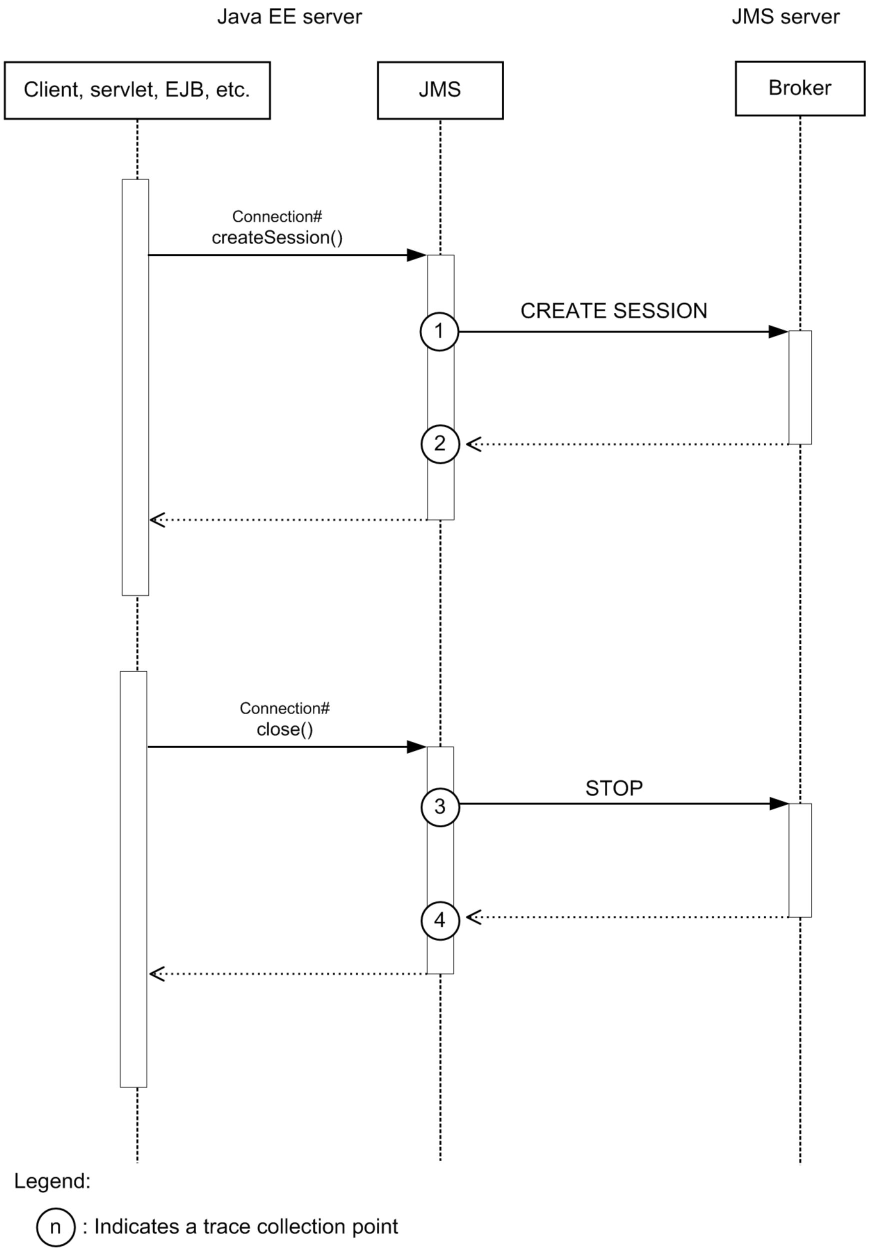
## 9.3.8 Trace collection points of JMS

The following provides details on the trace collection points of JMS.

### For Connection object operations

The following shows the trace collection points for Connection object operations.

Figure 9-19: Trace collection points of JMS (for Connection object operations)



## Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

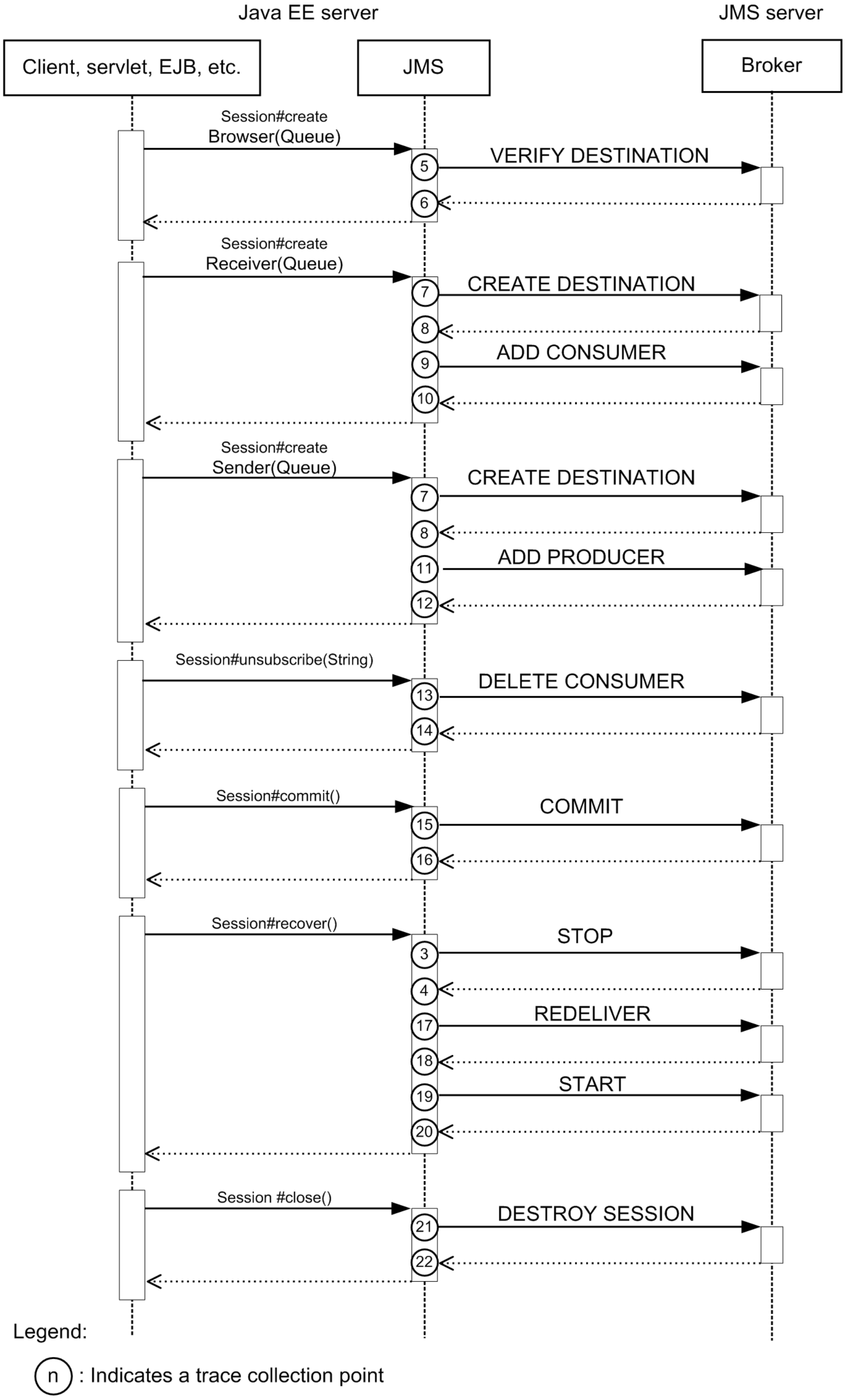
Table 9-21: Details on the trace collection points of JMS (for Connection object operations)

| Event ID | Num ber in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB800 | 1 | Standard level | Immediately before sending CREATE SESSION to Broker | - | - | - |
| 0xB801 | 2 | Standard level | Immediately after receiving the response to CREATE SESSION from Broker | - | - | - |
| 0xB812 | 3 | Standard level | Immediately before sending STOP to Broker | - | - | - |
| 0xB813 | 4 | Standard level | Immediately after receiving the response to STOP from Broker | - | - | - |

## For Session object operations

The following shows the trace collection points for Session object operations.

Figure 9-20: Trace collection points of JMS (for Session object operations)



## Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-22: Details on the trace collection points of JMS (for Session object operations)

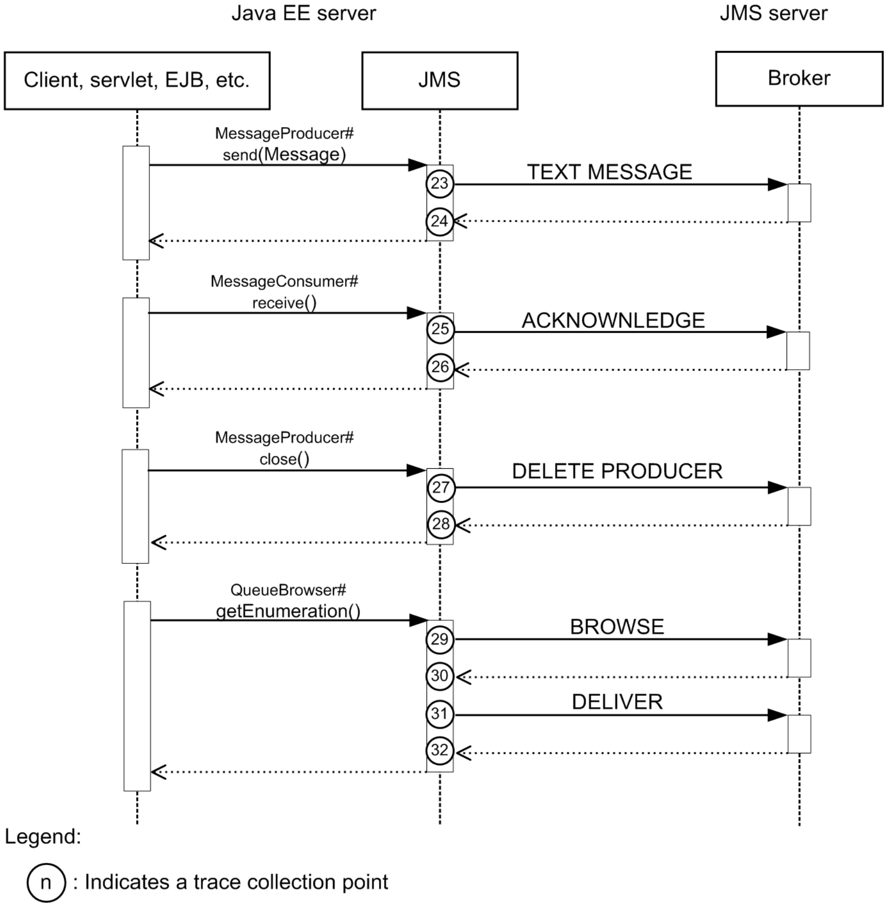
| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB802 | 21 | Standard level | Immediately before sending DESTROY SESSION to Broker | - | - | - |
| 0xB803 | 22 | Standard level | Immediately after receiving the response to DESTROY SESSION from Broker | - | - | - |
| 0xB804 | 7 | Standard level | Immediately before sending CREATE DESTINATION to Broker | - | - | - |
| 0xB805 | 8 | Standard level | Immediately after receiving the response to CREATE DESTINATION from Broker | - | - | - |
| 0xB806 | 5 | Standard level | Immediately before sending VERIFY DESTINATION to Broker | - | - | - |
| 0xB807 | 6 | Standard level | Immediately after receiving the response to VERIFY DESTINATION from Broker | - | - | - |
| 0xB808 | 9 | Standard level | Immediately before sending ADD CONSUMER to Broker | - | - | - |
| 0xB809 | 10 | Standard level | Immediately after receiving the response to ADD CONSUMER from Broker | - | - | - |
| 0xB80A | 13 | Standard level | Immediately before sending DELETE CONSUMER to Broker | - | - | - |
| 0xB80B | 14 | Standard level | Immediately after receiving the response to DELETE CONSUMER from Broker | - | - | - |
| 0xB80C | 11 | Standard level | Immediately before sending ADD PRODUCER to Broker | - | - | - |
| 0xB80D | 12 | Standard level | Immediately after receiving the response to ADD PRODUCER from Broker | - | - | - |
| 0xB810 | 19 | Standard level | Immediately before sending START to Broker | - | - | - |
| 0xB811 | 20 | Standard level | Immediately after receiving the response to START from Broker | - | - | - |
| 0xB812 | 3 | Standard level | Immediately before sending STOP to Broker | - | - | - |

| Event ID | Num ber in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB813 | 4 | Standard level | Immediately after receiving the response to STOP from Broker | - | - | - |
| 0xB814 | 15 | Standard level | Immediately before sending COMMIT or ROLLBACK to Broker | - | - | - |
| 0xB815 | 16 | Standard level | Immediately after receiving the response to COMMIT or ROLLBACK from Broker | - | - | - |
| 0xB81E | 17 | Standard level | Immediately before sending REDELIVER to Broker | - | - | - |
| 0xB81F | 18 | Standard level | Immediately after receiving the response to REDELIVER from Broker | - | - | - |

## For other object operations

The following shows the trace collection points for other object operations.

Figure 9-21: Details on the trace collection points of JMS (for other object operations)



**Reference note**

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-23: Details on the trace collection points of JMS (for other object operations)

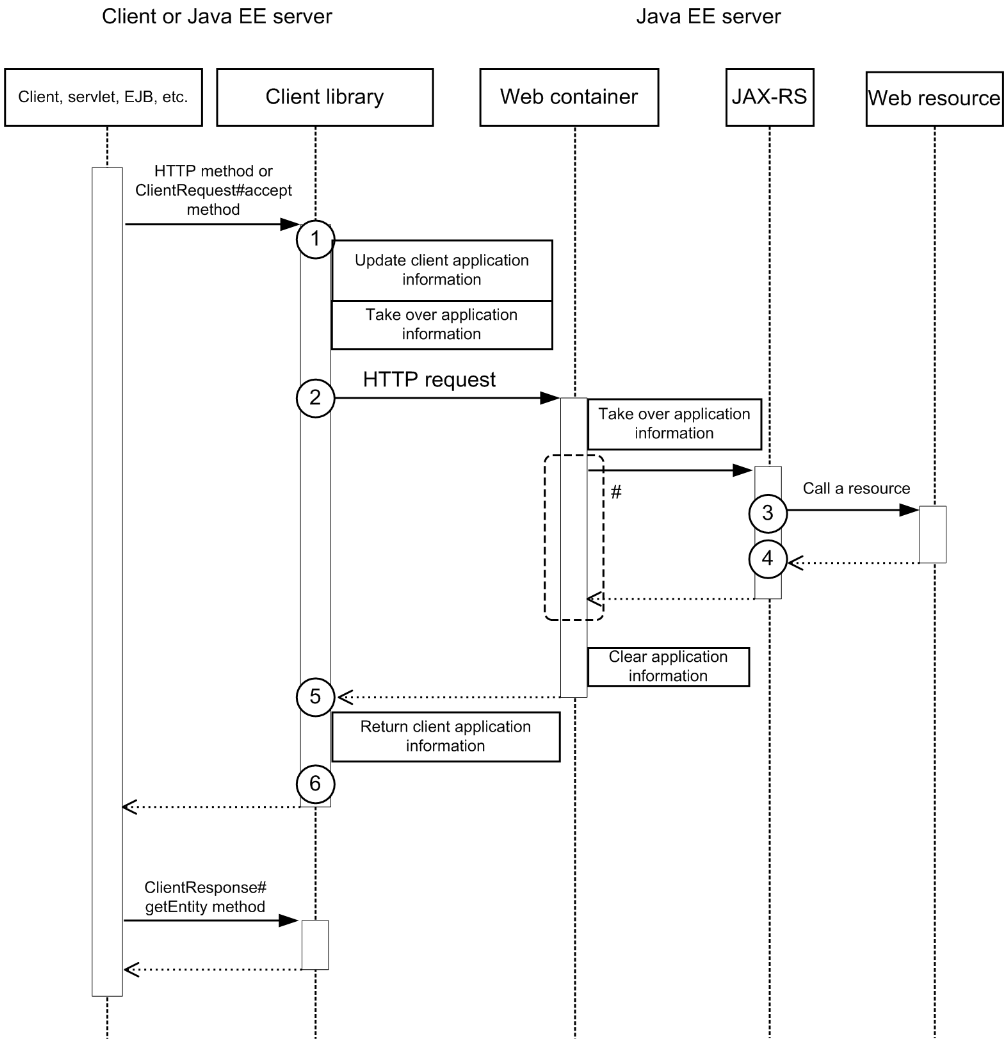
| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB80E | 27 | Standard level | Immediately before sending DELETE PRODUCER to Broker | - | - | - |
| 0xB80F | 28 | Standard level | Immediately after receiving the response to DELETE PRODUCER from Broker | - | - | - |
| 0xB816 | 23 | Standard level | Immediately before sending TEXT MESSAGE, MAP MESSAGE, BYTE MESSAGE, or OBJECT MESSAGE to Broker | - | - | - |
| 0xB817 | 24 | Standard level | Immediately after receiving the response to TEXT MESSAGE, MAP MESSAGE, BYTE MESSAGE, or OBJECT MESSAGE from Broker | - | - | - |
| 0xB818 | 25 | Standard level | Immediately before sending ACKNOWLEDGE to Broker | - | - | - |
| 0xB819 | 26 | Standard level | Immediately after receiving the response to ACKNOWLEDGE from Broker | - | - | - |
| 0xB81A | 29 | Standard level | Immediately before sending BROWSE to Broker | - | - | - |
| 0xB81B | 30 | Standard level | Immediately after receiving the response to BROWSE from Broker | - | - | - |
| 0xB81C | 31 | Standard level | Immediately before sending DELIVER to Broker | - | - | - |
| 0xB81D | 32 | Standard level | Immediately after receiving the response to DELIVER from Broker | - | - | - |

## 9.3.9 Trace collection points of JAX-RS

The following provides details on the trace collection points of JAX-RS.

The following shows the trace collection points of JAX-RS.

Figure 9-22: Trace collection points of JAX-RS


Client or Java EE server
Java EE server
Client, servlet, EJB, etc.
Client library
Web container
JAX-RS
Web resource
HTTP method or ClientRequest#accept method
1
Update client application information
Take over application information
HTTP request
2
Take over application information
#
Call a resource
3
4
Clear application information
5
Return client application information
6
ClientResponse# getEntity method

Legend:

(n) : Indicates a trace collection point.

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

#:

The part enclosed by the broken line is the same as the trace collection points of the web container when synchronous processing is used.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-24: Details on the trace collection points of JAX-RS

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xB900 | 1 | Standard level | The entry point of the method of the client library that sends the HTTP message | Class name | Method name | - |
| 0xB901 | 6 | Standard level | The exit point of the method of the client library that sends the HTTP message | Class name | Method name | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB902 | 2 | Standard level | Before sending the HTTP message of the client library | Class name | Method name | Endpoint URI |
| 0xB903 | 5 | Standard level | After receiving the HTTP message of the client library | Class name | Method name | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xB904 | 3 | Standard level | Before calling a resource | Class name | Method name | Types of calls<br>Outputs one of the following:<br>• ObjectOutInvoker<br>• ResponseOutInvoker<br>• TypeOutInvoker<br>• VoidOutInvoker<br>• VoidToVoidDispatcher |
| 0xB905 | 4 | Standard level | After calling a resource | Class name | Method name | In a normal state<br>-<br>In an abnormal state<br>Exception name |

**Related topics**

- 9.3.2 Trace collection points of the web container

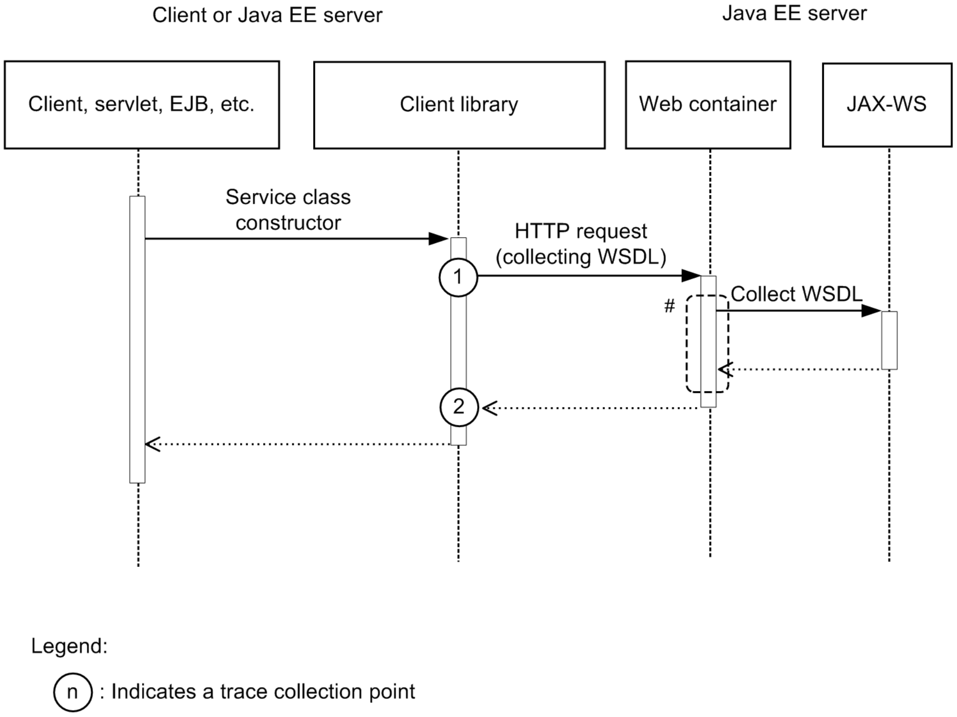
## 9.3.10 Trace collection points of JAX-WS

The following describes the trace collection points of JAX-WS.

## When collecting remote WSDL files

The following shows the trace collection points of JAX-WS when collecting remote WSDL files.

Figure 9-23: Trace collection points of JAX-WS (when collecting remote WSDL files)


Client or Java EE server
Java EE server
Client, servlet, EJB, etc.
Client library
Web container
JAX-WS
Service class constructor
HTTP request (collecting WSDL)
1
Collect WSDL
#
2
Legend:
n : Indicates a trace collection point

### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

#:

The part enclosed by the broken line is the same as the trace collection points of the web container when synchronous processing is used.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

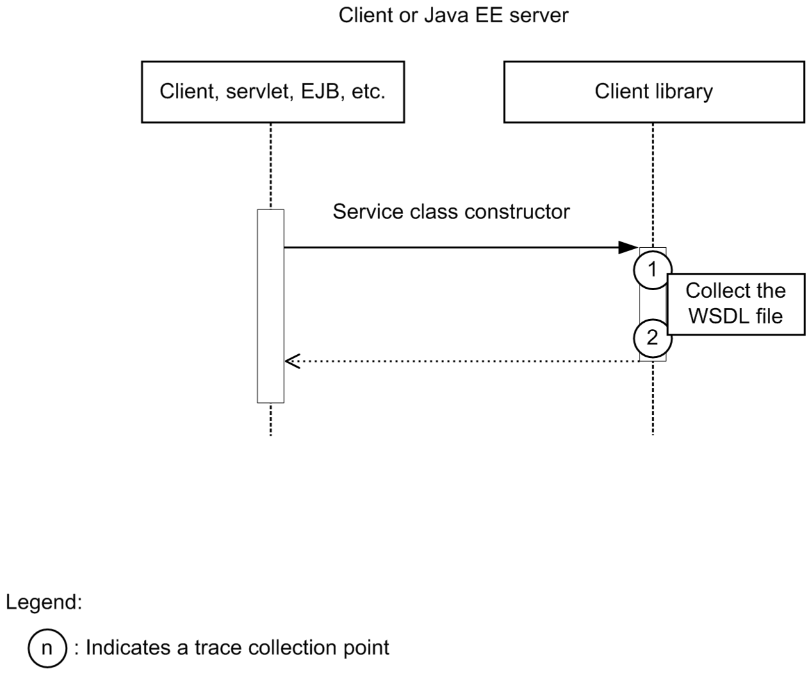
Table 9-25: Details on the trace collection points of JAX-WS (when collecting remote WSDL files)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xBA0A | 1 | Standard level | Before collecting WSDL files | Class name | Method name (number of parameters) | Endpoint URI |
| 0xBA0B | 2 | Standard level | After collecting WSDL files | Class name | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When collecting local WSDL files

The following shows the trace collection points of JAX-WS when collecting local WSDL files.

Figure 9-24: Trace collection points of JAX-WS (when collecting local WSDL FILES)


Client or Java EE server
Client, servlet, EJB, etc.
Client library
Service class constructor
1
Collect the WSDL file
2
Legend:
n : Indicates a trace collection point

### Reference note

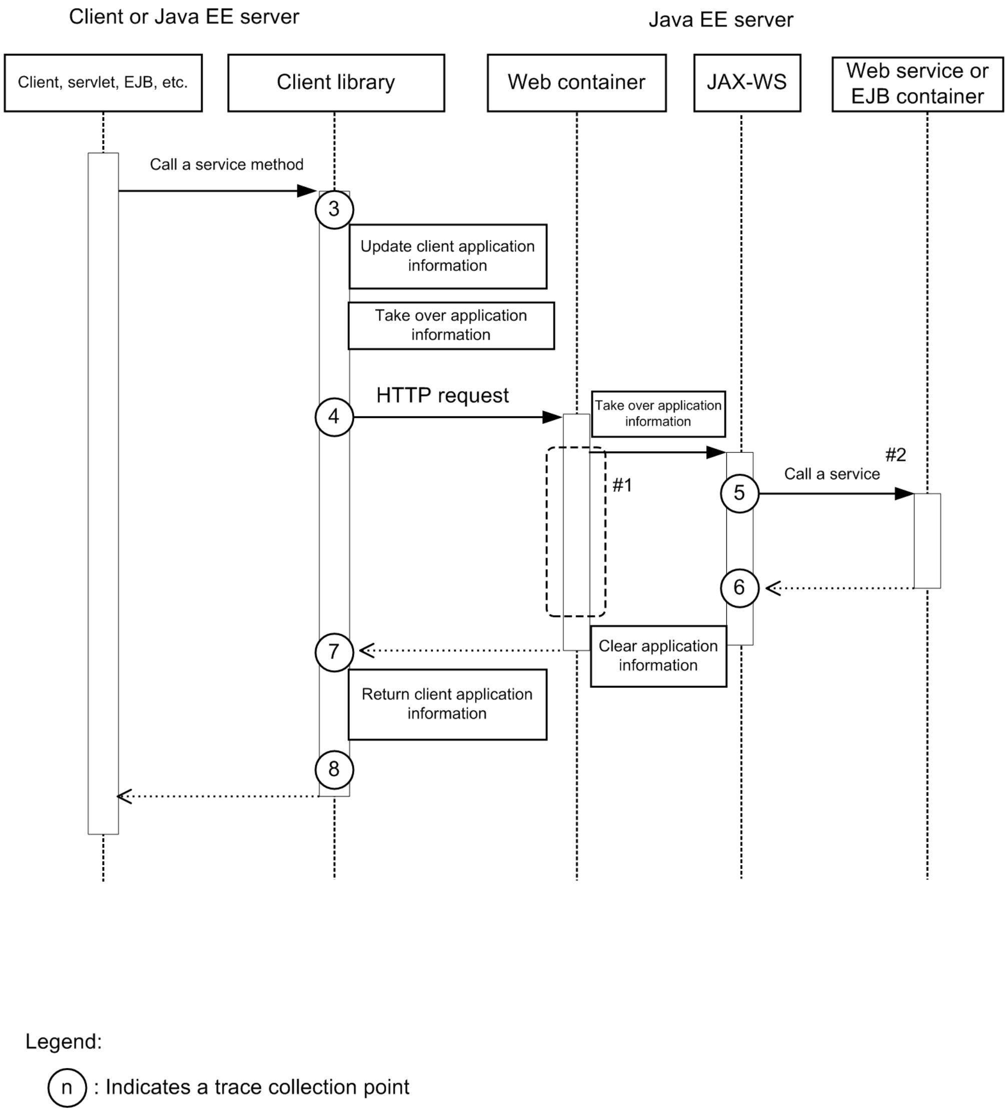
This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

The event IDs, trace levels, trace collection points, and information that can be collected are the same as those of collecting remote WSDL files.

## For a request-response operation (synchronous)

The following shows the trace collection points of JAX-WS for a request-response operation (synchronous).

Figure 9-25: Trace collection points of JAX-WS for a request-response operation (synchronous)


Client or Java EE server
Java EE server
Client, servlet, EJB, etc.
Client library
Web container
JAX-WS
Web service or EJB container
Call a service method
3
Update client application information
Take over application information
HTTP request
4
Take over application information
#2
Call a service
#1
5
6
7
Clear application information
Return client application information
8
Legend:
n : Indicates a trace collection point

## Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

#1:

The part enclosed by the broken line is the same as the trace collection points of the web container when synchronous processing is used.

#2:

The EJB container is called when EJB is implemented as the web service.

The collection points in the workflow after the EJB container is called are the same as those of local call of Session Bean and Entity Bean.

The following table lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-26: Details on the trace collection points of JAX-WS for a request-response operation (synchronous)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xBA00 | 3 | Standard level | The entry point of the service method of the stub-based SOAP client library | Class name | Method name (number of parameters) | - |
| 0xBA01 | 8 | Standard level | The exit point of the service method of the stub-based SOAP client library | Class name | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xBA02 | 3 | Standard level | The entry point of the service method of the dispatch-based SOAP client library | Class name | Method name (number of parameters) | - |
| 0xBA03 | 8 | Standard level | The exit point of the service method of the dispatch-based SOAP client library | Class name | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xBA04 | 4 | Standard level | Before sending the HTTP message of the SOAP client library | Class name | Method name (number of parameters) | Endpoint URI |
| 0xBA05 | 7 | Standard level | After receiving the HTTP message of the SOAP client library | Class name | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name |
| 0xBA06 | 5 | Standard level | Before calling the SOAP service (Web service implementation class) | Class name | Method name (number of parameters) | Service method name |
| 0xBA07 | 6 | Standard level | After calling the SOAP service (Web service implementation class) | Class name | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name# |
| 0xBA08 | 5 | Standard level | Before calling the SOAP service (Provider implementation class) | Class name | Method name (number of parameters) | - |
| 0xBA09 | 6 | Standard level | After calling the SOAP service (Provider implementation class) | Class name | Method name (number of parameters) | In a normal state<br>-<br>In an abnormal state<br>Exception name# |

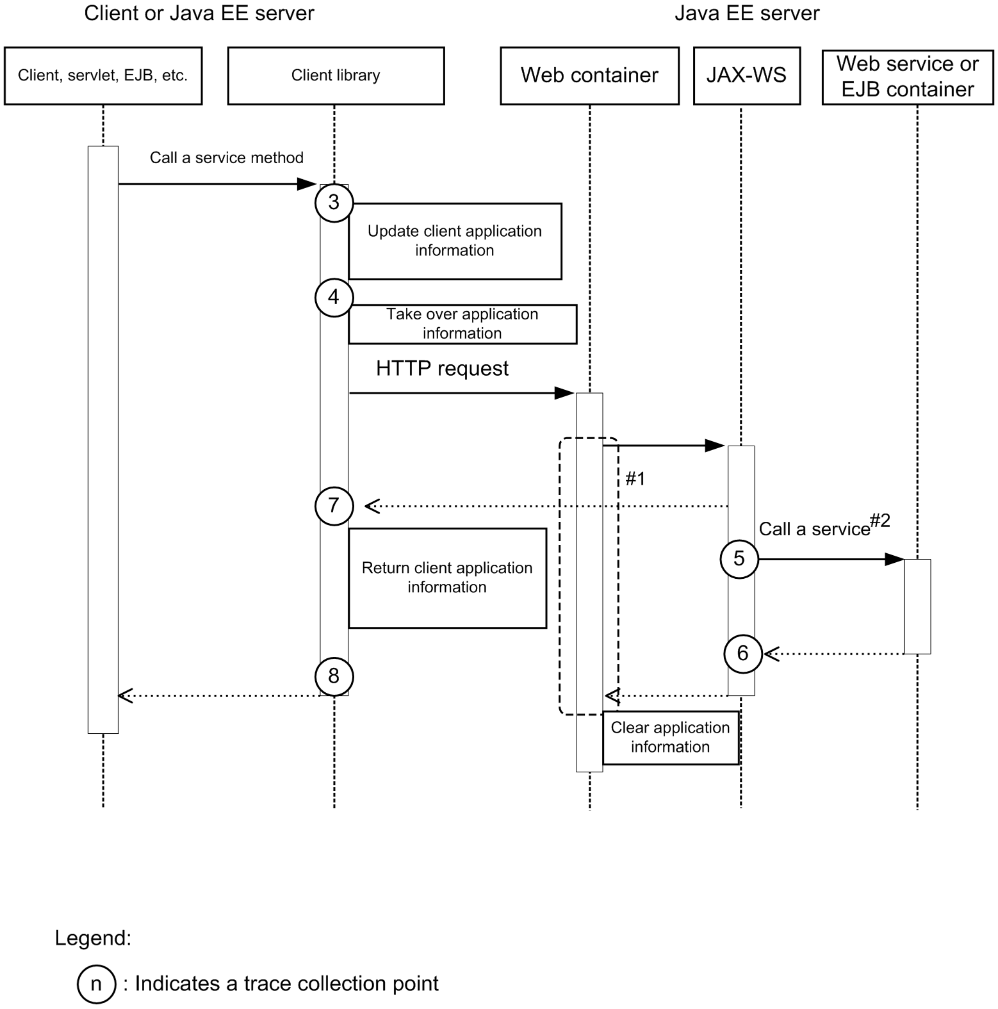
#:

If the exception collected at a trace collection point is java.lang.reflect.InvocationTargetException, the name of the exception retained by java.lang.reflect.InvocationTargetException is output.

## For a one-way operation

The following shows the trace collection points of JAX-WS for a one-way operation (synchronous).

Figure 9-26: Trace collection points of JAX-WS (for a one-way operation)


Client or Java EE server
Java EE server
Client, servlet, EJB, etc.
Client library
Web container
JAX-WS
Web service or EJB container
Call a service method
3
Update client application information
4
Take over application information
HTTP request
#1
7
Call a service#2
5
Return client application information
6
8
Clear application information
Legend:
n : Indicates a trace collection point

> **Reference note**
>
> This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

#1:

The part enclosed by the broken line is the same as the trace collection points of the web container when synchronous processing is used.

#2:

The EJB container is called when EJB is implemented as the web service.

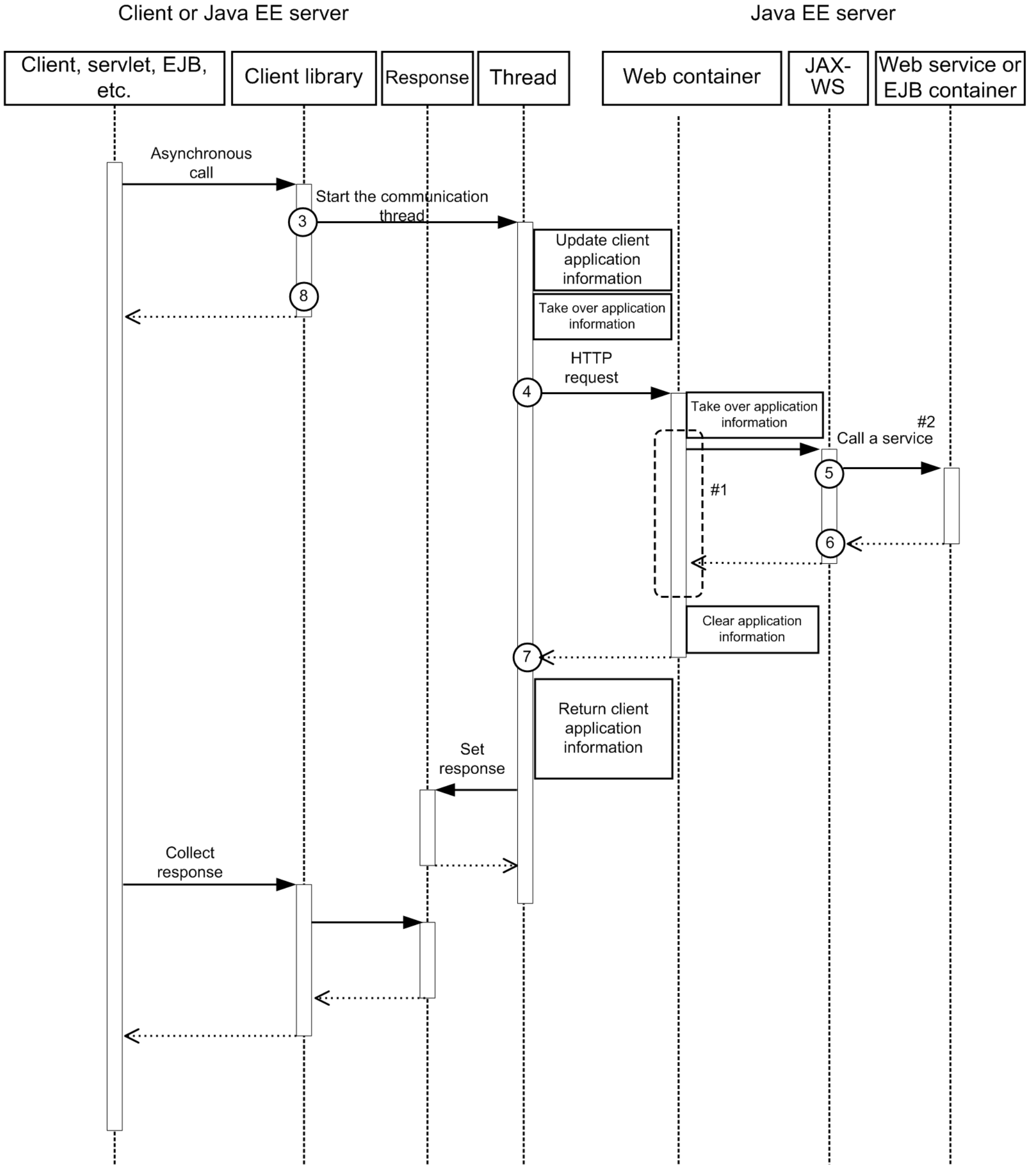
The collection points in the workflow after the EJB container is called are the same as those of local call of Session Bean and Entity Bean.

The event IDs, trace levels, trace collection points, and information that can be collected are the same as those in the details on the trace collection points of JAX-WS for a request-response operation (synchronous).

### For a request-response operation (asynchronous)

The following shows the trace collection points of JAX-WS for a request-response operation (asynchronous).

Figure 9-27: Trace collection points of JAX-WS for a request-response operation (asynchronous)


Client or Java EE server
Java EE server
Client, servlet, EJB, etc.
Client library
Response
Thread
Web container
JAX-WS
Web service or EJB container
Asynchronous call
Start the communication thread
3
8
Update client application information
Take over application information
HTTP request
4
Take over application information
#2
Call a service
5
#1
6
Clear application information
7
Return client application information
Set response
Collect response

Legend:

(n) : Indicates a trace collection point

#### Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all the triggers to collect traces. Other triggers for collecting traces exist.

#1:

The part enclosed by the broken line is the same as the trace collection points of the web container when synchronous processing is used.

#2:

The EJB container is called when EJB is implemented as the web service.

The collection points in the workflow after the EJB container is called are the same as those of local call of Session Bean and Entity Bean.

The event IDs, trace levels, trace collection points, and information that can be collected are the same as those in the details on the trace collection points of JAX-WS for a request-response operation (synchronous).

### Related topics

- 9.3.2 Trace collection points of the web container
- 9.3.3 Trace collection points of the EJB container

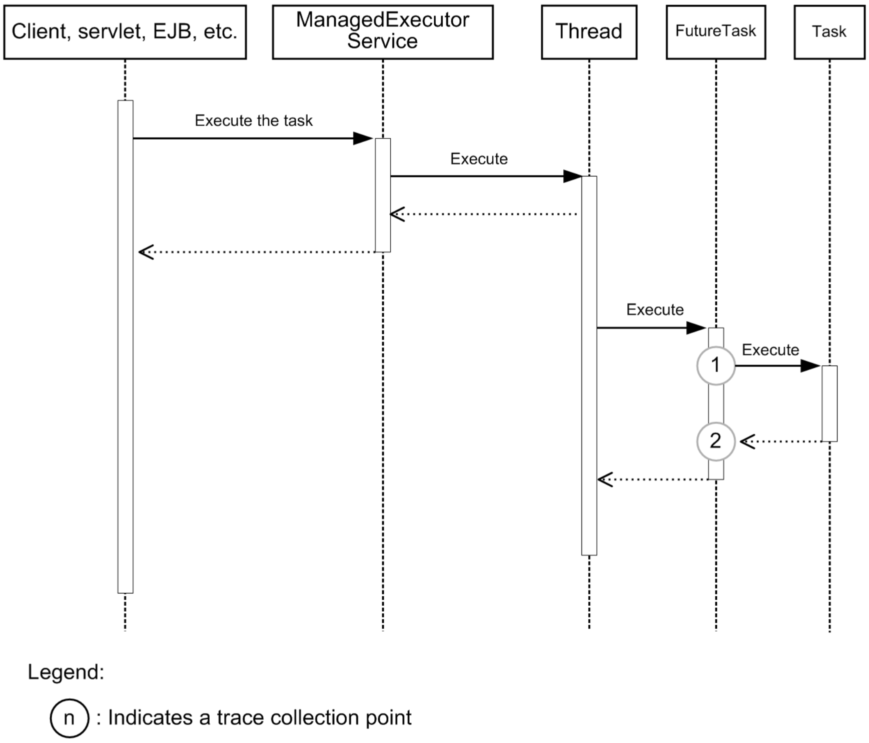
## 9.3.11 Trace collection points of Concurrency Utilities

The following provides details on the trace collection points of Concurrency Utilities.

### When a task is executed in an ExecutorService in Java EE

The figure below shows the trace collection points of Concurrency Utilities when a task is executed in an ExecutorService in Java EE.

Figure 9-28: Trace collection points of Concurrency Utilities (when a task is executed in an ExecutorService in Java EE)



**Reference note**

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all triggers for collecting traces. Other triggers for collecting traces exist.

The table below lists event IDs, trace levels, trace collection points, and information that can be collected.

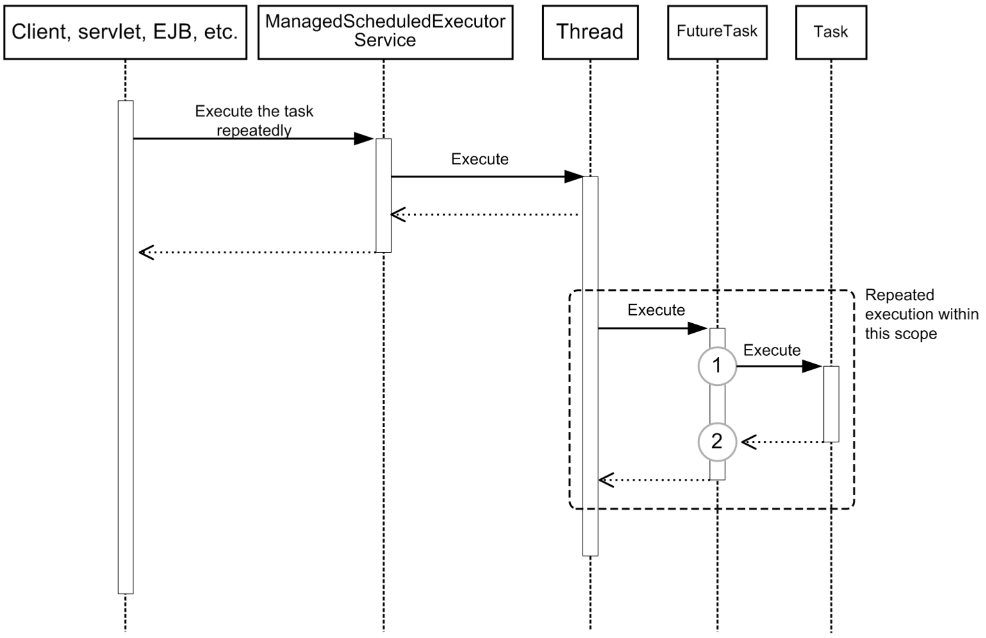
Table 9-27: Details on the trace collection points of Concurrency Utilities (when a task is executed in an ExecutorService in Java EE)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xBC00 | 1 | Standard level | Immediately before executing a task by using ManagedExecutorService or ManagedScheduledExecutorService | Class name | - | - |
| 0xBC01 | 2 | Standard level | Immediately after executing a task by using ManagedExecutorService or ManagedScheduledExecutorService | Class name | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

## When a task is repeatedly executed in a ScheduledExecutorService in Java EE

The figure below shows the trace collection points of Concurrency Utilities when a task is repeatedly executed in a ScheduledExecutorService in Java EE.

Figure 9-29: Trace collection points of Concurrency Utilities (when a task is repeatedly executed in a ScheduledExecutorService in Java EE)



Client, servlet, EJB, etc.
ManagedScheduledExecutor Service
Thread
FutureTask
Task
Execute the task repeatedly
Execute
Execute
Execute
Repeated execution within this scope
1
2

Legend:

(n) : Indicates a trace collection point

## Reference note

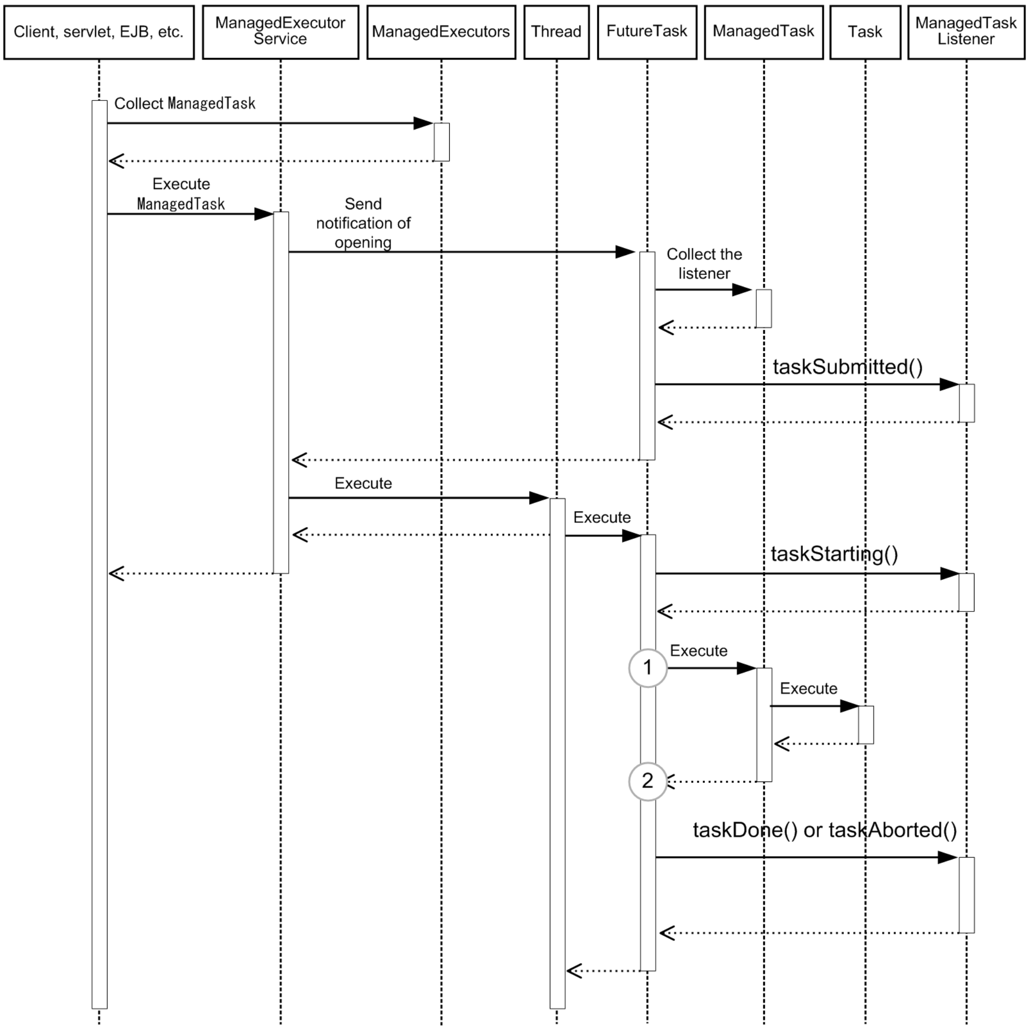
This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The event IDs, trance levels, trace collection points, and information that can be collected are the same as those described in "Details on trace collection points of Concurrency Utilities (when a task is executed in an ExecutorService in Java EE)".

## When ManagedTask is executed in an ExecutorService in Java EE

The figure below shows the trace collection points of Concurrency Utilities when ManagedTask is executed in an ExecutorService in Java EE.

Figure 9-30: Trace collection points of Concurrency Utilities (when ManagedTask is executed in an ExecutorService in Java EE)



Legend:

(n) : Indicates a trace collection point

## Reference note

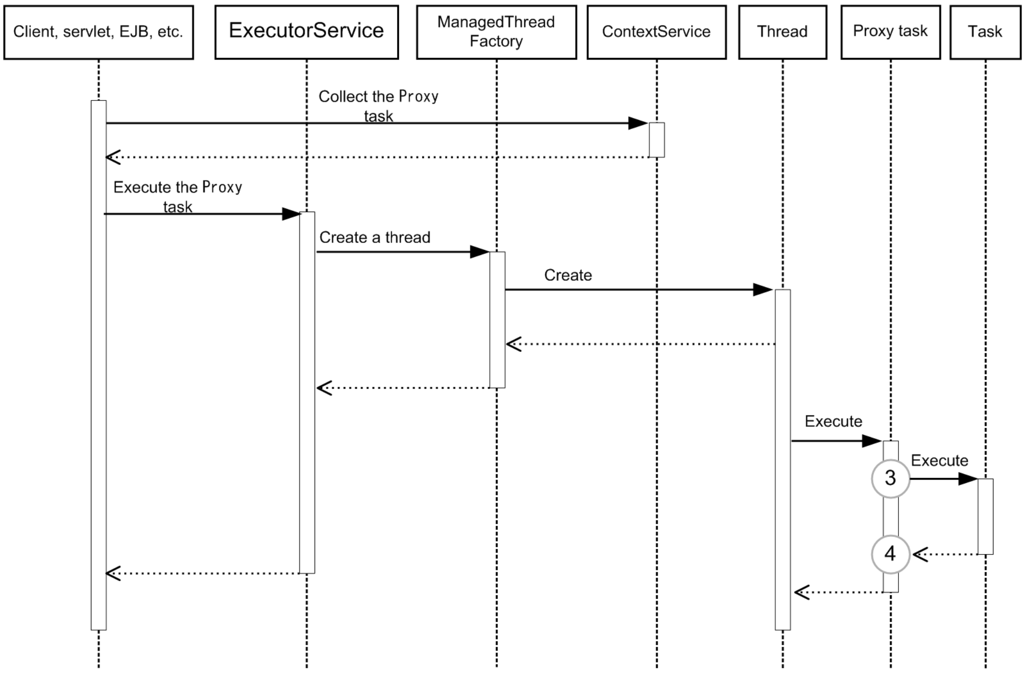
This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The event IDs, trance levels, trace collection points, and information that can be collected are the same as those described in "Details on trace collection points of Concurrency Utilities (when a task is executed in an ExecutorService in Java EE)".

## When a task with context information added is executed in an ExecutorService in Java SE

The figure below shows the trace collection points of Concurrency Utilities when a task with context information added is executed in an ExecutorService in Java SE.

Figure 9-31: Trace collection points of Concurrency Utilities (when a task with context information added is executed in an ExecutorService in Java SE)



Legend:

(n) : Indicates a trace collection point

## Reference note

This figure shows the workflow for collecting traces as an extension of a request. This figure does not show all of the triggers for collecting traces. Other triggers for collecting traces exist.

The table below lists event IDs, trace levels, trace collection points, and information that can be collected.

Table 9-28: Details on the trace collection points of Concurrency Utilities (when a task with context information added is executed in an ExecutorService in Java SE)

| Event ID | Number in the figure | PRF trace collection level | Trace collection point | Information that can be collected | | |
|---|---|---|---|---|---|---|
| | | | | Interface name | Operation name | Option |
| 0xBC02 | 3 | Standard level | Immediately before executing a task with context information added | Class name | - | - |
| 0xBC03 | 4 | Standard level | Immediately after executing a task with context information | Class name | - | In a normal state<br>-<br>In an abnormal state<br>Exception name |

# Glossary

## A

### Administration Console

A GUI used to build and operate systems using Application Server.

## C

### CA

This is an abbreviation of Certificate Authority. This is function for issuing certificate for using SSL is also called as authentication authority. This is entity in reliability that issues a valid certificate for other entity. The types of this certificate are:

Root CA : Correctness of self is certified by self, without receiving the certification of the upper level CA.

Intermediate CA : Receives the certificate from the upper level CA for certifying the correctness of self.

### cluster

A group of server instances sharing the same applications, resources, and settings information.

### CopyGC

GC performed for the New area when using SerialGC.

## D

### domain

A group that controls components forming a business system, namely server instances, web servers, and performance tracers.

### domain administration server

A server that administers a domain.

## E

### Explicit heap

Memory storage not subject to GC that is used with the explicit memory management functionality.

### explicit memory management functionality

A functionality that prevents full GC from occurring when using SerialGC.

## F

### full GC

GC performed for the Tenured area.

## G

### G1GC

A GC processing mode. Performs YoungGC for the New area, and MixedGC for the Tenured area. A processing mode that emphasizes the alleviation of application down time due to GC.

### GC

Memory recovery processing performed automatically by Java VM.

## H

### HTTP

HTTP is an abbreviation for Hypertext Transfer Protocol. This is the communication protocol for sending and receiving the HTML document between the web server and web client, by using Internet.

## I

### identifier targeted by the setting

An identifier that is a target of a setting for which setting values can be changed or acquired by using the `set`, `get` or `list` subcommand of the `asadmin` utility.

## J

### Java EE RI

A Java EE reference implementation.

### Java EE RI DD

A deployment descriptor provided by the Java EE reference implementation.

## L

### load balancer

A device or piece of software that controls the transfer of requests to equalize the load on back end servers.

## N

### node

The host on which a server instance, web server, or performance tracer is running.

## P

### Planned termination

Planned termination is a method to stop running server processes or server threads after execution completes.

### private key

A key used for public-key encryption to decrypt data or create an electronic signature. It is also called a *secret key*. Each private key is safely managed by their owners. The recipient of encrypted data uses his or her own private key to decrypt the data. Similarly, the sender of data uses his or her own private key to create an electronic signature when attaching an electronic signature to that data.

### public key

A key used for public-key encryption to encrypt data or verify electronic signatures. Normally, public keys are published on the network. The sender of encrypted data uses the recipient's public key to encrypt the data. Similarly, the recipient uses the sender's public key to verify the electronic signature of received data.

## S

### SerialGC

A GC processing mode. Performs CopyGC for the New area, and full GC for the Tenured area. A processing mode that emphasizes throughput.

### server instance

The server process that runs the Java EE application.

### server template

A file documenting the settings needed to operate a server instance, web server, and performance tracer, which is referenced by the domain administration server when building environments.

### specific DD

A definition file in XML format used to configure Java EE Server-specific functions for each Java EE application and module.

### SSL

This is an abbreviation of Secure Sockets Layer. This is one of the protocols for sending and receiving the encrypted information through the network. This is the topmost layer of TCP. This layer performs authentication, key exchange, encryption and message authentication by certification between the client and web server.

## T

### TLS

This is an abbreviation of Transport Layer Security. This is one of the protocols for sending and receiving the encrypted information through the network. This protocol has been made better based on SSL.

## U

### URI

This is an abbreviation of Uniform Resource Identifier. This is the standard that has the method of specifying the position of data on the web and method of specifying a certain data on the web with a specific name, as the method of URL. It has the combination of URL and URN and you can use the same for expressing the internet resources, and identifying the name space in an XML document. URI is a generic concept and holds a position above URL. It is prescribed with RFC1630.

## W

### web front system

A system pattern that Application Server mainly support. A system consisting of a security device, load balancer, Application Server, database server, job management server, and message monitoring server.

### Web server

A program to execute the processing related to requests receptions from the web browser and data transmission to web browser.

# Index