HITACHI
Inspire the Next

Hitachi OSGi Series
SuperJ DMS Installation and Configuration Guide

80142-CST-000-02
April 2013

Hitachi Communication Technologies America, Inc.

Copyright Notice

This edition (80142-CST-000-02 for product R3.3.1, April 2013) replaces previously dated versions of this document. Although this document was believed to be correct at the time of publication, Hitachi Communication Technologies, Ltd., Hitachi Communication Technologies America, Inc., and their affiliates assume no responsibility for errors or omissions. The contents of this manual and the specifications of this product are subject to change without notice. Hitachi-CTA reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including, but not limited to typographical and other errors relating to the publication. Technical bulletins may be issued at any time to further clarify product installation, maintenance, use, and administration. In the event of any differences, the information in more recently issued technical bulletin supersedes correlative information in associated product documentation.

For additional information about the products in this document, please contact:

Hitachi Communication Technologies America, Inc.

3617 Parkway Lane
Norcross, GA 30092
Tel (770) 446-8820 Fax (770) 242-1414

Trademarks

Hitachi® and AMN® are registered trademarks of Hitachi, Ltd. SuperJ™, SuperJ Engine®, and SuperJ Framework™ are trademarks or registered trademarks of Hitachi Solutions, Ltd. in Japan.

Certain software products within the SuperJ™ Suite contain Java® software which is owned by Oracle Corporation. The terms and conditions of its use are governed by your agreement with Hitachi. Oracle and Java are registered trademarks of Oracle and/or its affiliates.

All other trademarks are the properties of their respective owners.

SuperJ DMS Installation and Configuration Guide

80142-CST-000-02
April 2013

Revision History

The following table identifies the history of revisions to this document, the *SuperJ DMS Installation and Configuration Guide* (80142-CST-000-02).

Revision Date	Revision Level	For Product Release	Summary of Changes
April 2013	02	Release 3.3.1.3	Initial release of manual.

TABLE OF CONTENTS

PREFACE

Please read this Preface to learn important information about using this document and the product(s) that it describes. The Preface contains general information about this document as well as important safety notices.

Contents

Topics covered in this Preface include: **Page**

About this Document

This section provides some general information that you need to know when using this document.

Purpose

This document describes installation and administration of the SuperJ™ Deployment Management System (SJDMS). The document also serves as a configuration and ordering guide.

Intended audience

This document is intended for individuals who will install and administer the SuperJ™ Deployment Management System (SJDMS).

Related Documents

In addition to this guide, please refer to the following documents for more information:

Document	Version	Source
OSGi Service Platform Core Specification	Release 4, Version 4.2 June 2009	The OSGi Alliance
OSGi Service Platform Service Compendium	Release 4, Version 4.2 August 2009	The OSGi Alliance
OSGi Service Platform Mobile Specification	Release 4, Version 4.0 July 2006	The OSGi Alliance
J2EE Client Provisioning Specification	Version 1.0 (JSR124)	Sun Microsystems, Inc.
SuperJ Framework™ User's Guide	Current version	Hitachi Communication Technologies America, Inc.
SuperJ Tools User's Guide	Current version	Hitachi Communication Technologies America, Inc.

A Word about Notices

Our notices delineate possible conditions that could affect operability of software for all of our product lines. This section describes how different types of safety notices are presented in this document.

⚠IMPORTANT

Comply with all danger, warning, and caution statements in this document.

Hazardous conditions

This document includes safety notices that emphasize the potential for these hazardous conditions:

- Warning
- Caution

Warning message

Warning messages indicate that you are in the presence of a risk that can cause catastrophic software failure if the risk is not avoided, which includes:

- Unrecoverable software errors that could cause irretrievable data or complete corruption of data
- Server inoperability
- Necessitating Hitachi-CTA intervention to restore operation

This document uses the following format to indicate **Warning** messages:

⚠WARNING!

Text describing the risk condition.

Caution message

Caution messages indicate the presence of a risk that may cause losses if the risk is not avoided. Property damage includes:

- Software malfunction
- Loss of software or data
- Short-term service interruption

This document uses the following format to indicate **Caution** messages:

> ⚠ CAUTION!
>
> Text describing the risk condition.

Important message

Important messages indicate information that is important to the successful and/or efficient operation of the product, including:

- Settings for the hardware or software to avoid conflict with other devices
- Configuration information to ensure maximum efficiency
- Reminders to verify that vital prerequisite processes have been completed

This document uses the following format to indicate **Important** messages:

> ⚠ IMPORTANT!
>
> Text describing necessary information.

IMPORTANT NOTICES

This section contains important information about successfully and efficiently operating the Client Console.

⚠ IMPORTANT!

Please read this entire document before you use this product. Successful and efficient use of your Client Console depends upon careful reading of this document. Be sure to carefully follow all instructions that accompany this product. Pay particular attention to notice statements, and keep this document for future reference.

Technical support

After following the procedures in this document, if the product is still not functioning properly, contact the National Technical Assistance Center (NTAC) at 1-800-944-7185, or via email at ntac@hitachi-cta.com.

1 OSGi Platform and Network OS architecture

This section provides an overview of the Hitachi SuperJ OSGi Platform and the SuperJ™ Deployment Management System architecture.

Contents

Topics covered in this document include: **Page**

Hitachi SuperJ OSGi Overview

Hitachi's SuperJ OSGi Platform provides the necessary elements of a total Open Service Gateway Initiative (OSGi) ecosystem. The SuperJ product family consists of the following independently functioning elements:

Element	Description
Java Virtual Machine (JVM)	Java Virtual Machine (JVM)
SuperJ Framework (SJF)	OSGi Service Platform Release 4.x-compliant OSGi framework (CPE client)
SuperJ Deployment Management System (SJDMS) with integral CPE-resident SuperJ Management Agent (SJMA) for communication	OSGi management server system
SuperJ Tools (SJT)	OSGi Software Development Kit (SDK) and Application Programming Interfaces (see API (Application Programming Interface)

The basic workflow for Hitachi's OSGi solution is shown in the following figure:

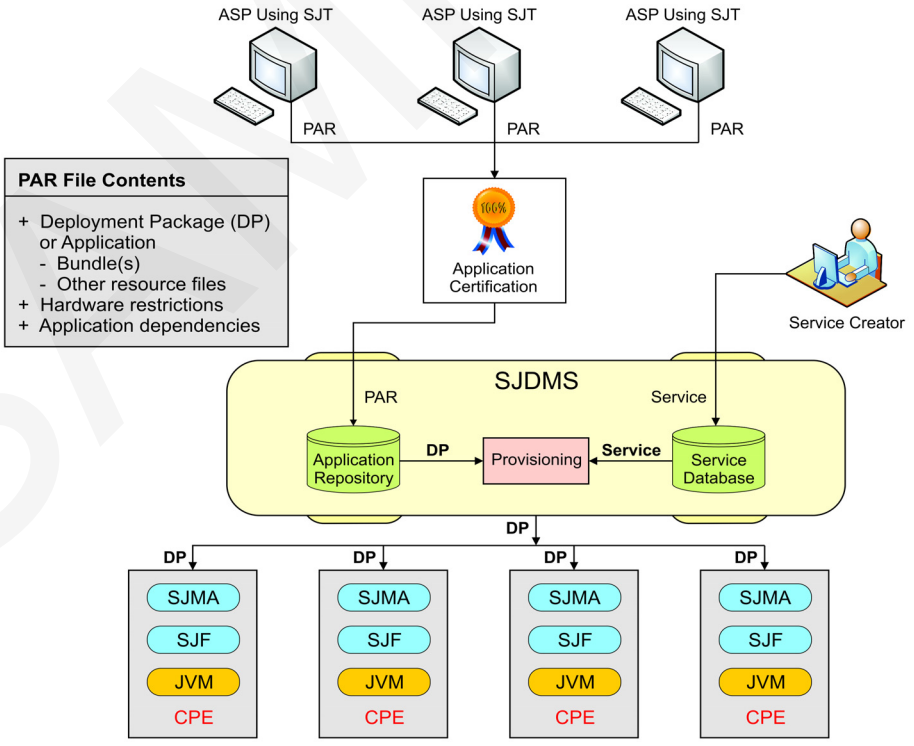

Figure 1-1: Workflow for OSGi solution

SuperJ Deployment Management System

Hitachi's SuperJ Deployment Management System (SJDMS) is a service provider-class system for managing applications and services on R4.x-compliant OSGi Frameworks in a scalable, redundant, and secure manner.

The SJDMS provides user-friendly GUIs for operators as well as an NBI (Northbound Interface) for service provider operations, business, and web services systems integration. Adminstrators can manage the framework through the SuperJ Management Agent (SJMA) bundle resident on the OSGi framework in the customer premises equipment (CPE).

The architecture of the SJDMS is shown in Figure 1-2:

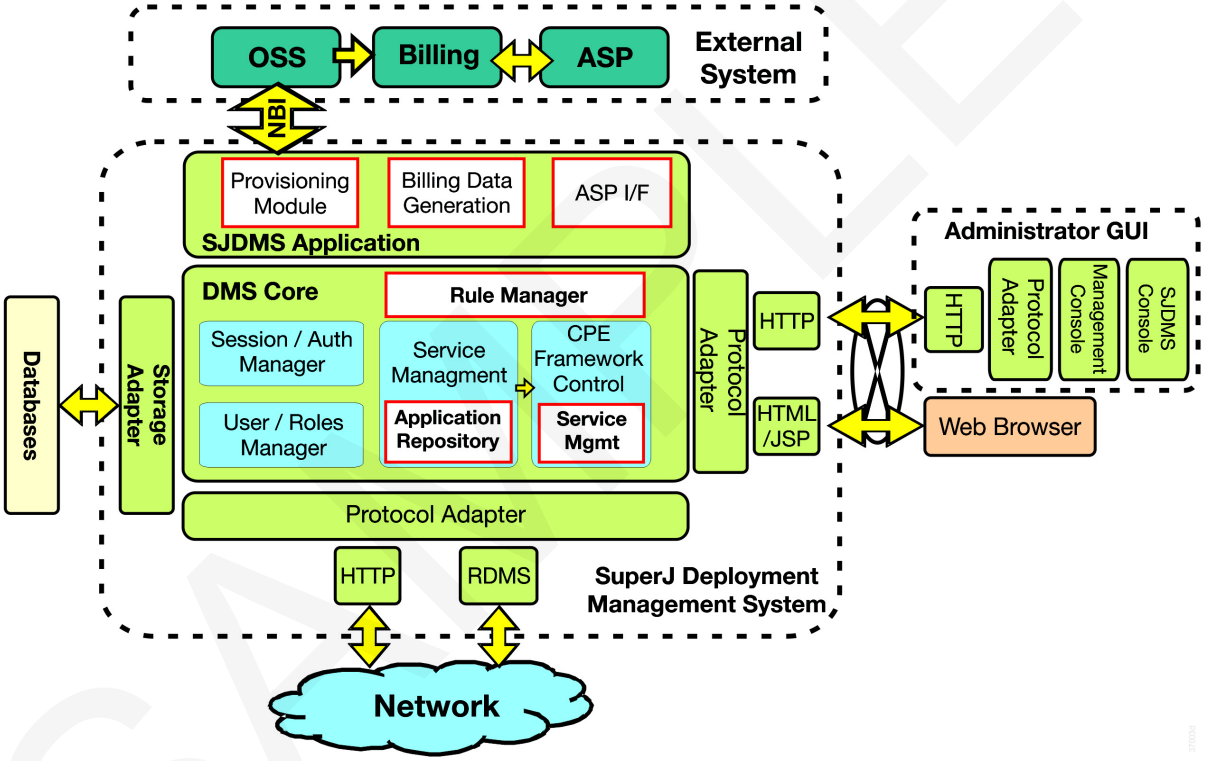

Figure 1-2: SJDMS architecture

continued...

SuperJ Deployment Management System, continued

The following table highlights the features of the SJDMS:

Feature	Description
OSGi application management	The SJDMS supports the complete lifecycle of bundles and applications and services from registration to end of life. • Manages R4.x compliant OSGi Frameworks via the SJDMS Agent bundle. • Supports the OSGi R4.x compliant operations: - Install - Start - Stop - Upgrade - Uninstall • Manages a repository of OSGi bundles and applications. • Supports groupings of *application*s into *services*. • Ensures delivery of applications to the R4.x compliant OSGi Framework. • Manages application dependencies.
Framework Resource Monitoring[a]	The SJDMS supports the retrieval and display of statistics for monitoring the health of the OSGi framework. Statistics include: • JVM Summary - JVM Name/Version - JVM Vendor - Number of active threads - Number of deamon threads - Total Java heap memory size - Free Java heap memory size - Operation System name - CPU architecture - Class path - Library path - Boot class path - Uptime • Resource usage per bundle - Number of total loaded classes - Number of current loaded classes - Number of active threads - Number of deamon threads - Number of created threads - Size of file storage used - Uptime
Rules Manager (manifest file)	The Hitachi SJDMS uses the OSGi standard approach through an XML based provisioning descriptor file based on JSR124. These rules are extracted by the SJDMS when the application package is input to the SJDMS. The rules are bound to the Applicaton Service Provider (ASP) application via the manifest file. The SJDMS dynamically imports the rules from the ASP's application package (deployment package) when the application is registered with the SJDMS. The rules are validated against the XML schema.

Feature	Description
Flow Setup	The SJDMS supports the following setup processes: 1. ASP creates application with the appropriate rules via the SuperJ Tools (SJT) or other OSGi-compliant J2ME SDK. 2. After verification of the application by the service provider, a digital certificate can be added to the application. 3. Service provider registers the application with the SJDMS. 4. The SJDMS parses the rules and adds the application to the repository (database). 5. The SJDMS is now ready for a provisioning system or a service portal to request application install and activation.
Diagnostics and logging	NBI for OSS integration • Log file • Alert file (error reporting) • State (active, inactive) monitoring • Self-diagnostic test[a] • Resource (thread, memory) monitoring[a]
Communications	The SJDMS uses http/https for the communication protocol. Since there is no standard protocol for OSGi management at this time, the SJDMS uses a proprietary protocol. Hitachi is tracking the various standards working to define to a management protocol and will support the standard protocol after it has been approved.
User Security	The SJDMS provides strict user and NBI security through the use of: • Roles • User Groups • Access control restrictions • Password and authentication control • Logging
Platform	The SJDMS has been certified on: • Solaris 10 10/09 • Oracle Database 11g Release 2 Enterprise Edition

a. This feature is being developed for a future release.

SJDMS Server Tier Architecture

Hitachi's SuperJ™ Deployment Management System (SJDMS) allows you to start with one or more applications to match your current needs, then add new applications and adapters as your network management requirements evolve.

The multi-tiered server architecture is flexible and highly scalable. All the service tiers can be installed on one physical computer (server machine) using Solaris zones (virtual machines), or distributed across multiple physical computers. Regarding fault management, if a business server fails, it will automatically fail over to other business servers. Other processes must be restarted manually

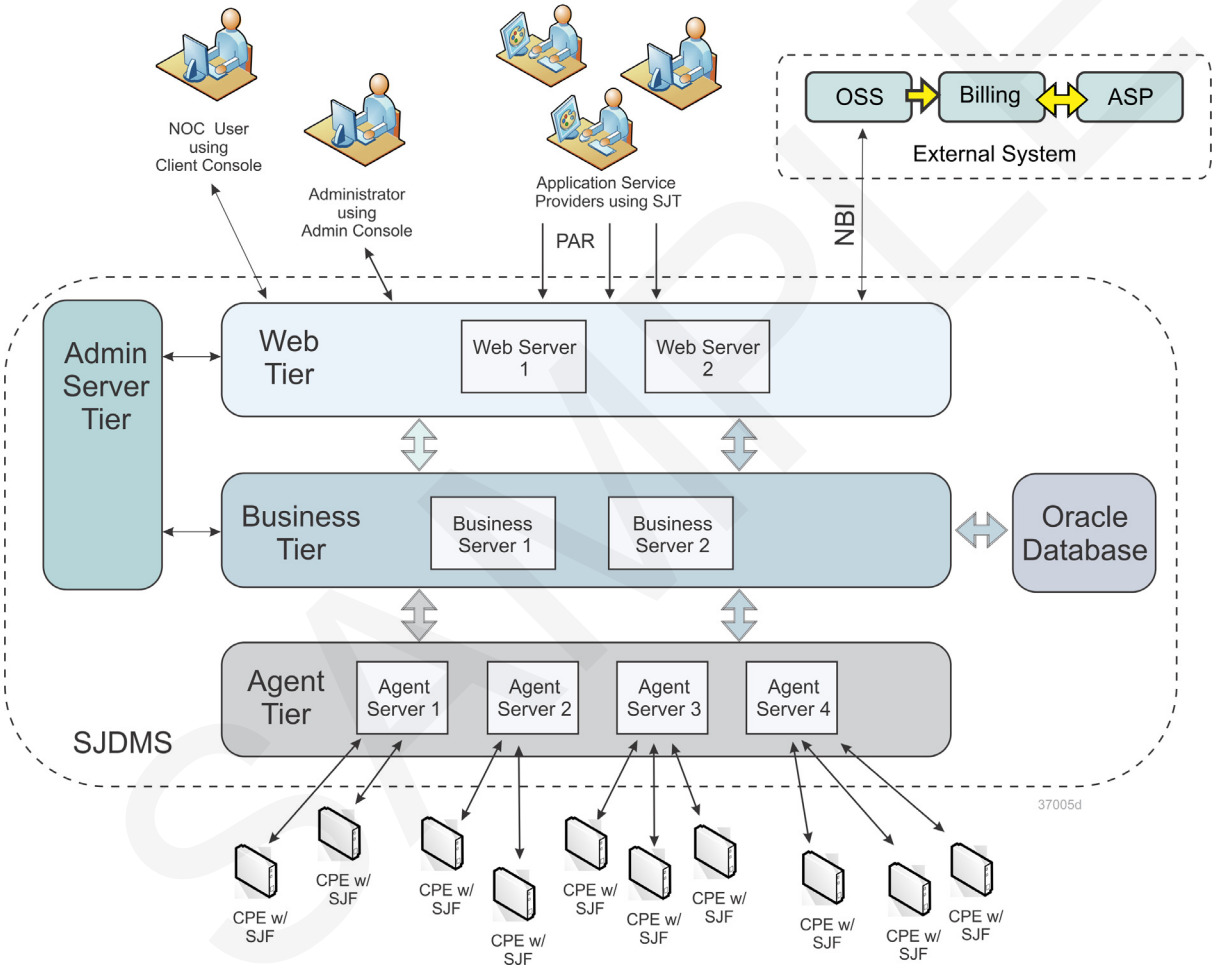

Figure 1-3: SJDMS Server Tier Architecture

As shown in Figure 1-3, each tier of the system architecture provides a specific function that interacts only with adjacent tier. This architecture allows the tiers to execute collectively on a single server in separate zones (virtual machines) or to be distributed across multiple servers depending on the size of the network.

The SJDMS architecture consists of the following service tiers:

Admin server tier

The Admin tier provides the Oracle WebLogic application server. The SJDMS web and business servers are deployed as WebLogic applications. Through the WebLogic Administrator web interface, you can monitor, start, and stop each of the web and business tier servers in your SJDMS system.

Web server tier

The Web tier presents the SJDMS Admin Client, SJDMS Client Console, and SJDMS NBI to operators using an HTTPS interface. The web tier is often collocated with the business logic tier for ease of administration and improved performance, but it can also be distributed in clusters.

Business server tier

The business server tier contains the core application logic of the Admin Console and it provides framework services such as transaction and database persistence. The business tier has a global view of the managed network and interfaces with the agents and the database. The business logic tier presents service interfaces to the web tier, and directs the agents to perform specific network layer tasks. Web client applications connect through the web tier to interact with the business tier to perform framework management tasks.

Agent tier

The agent tier is responsible for communicating directly with OSGi frameworks and can be remotely located close to the frameworks being managed. The agent tier is responsible for communication with the frameworks to be managed.

Database

The SuperJ™ Deployment Management System supports connectivity to your Oracle database to store all of the network and business data for the system.

It is highly recommended that you perform regular database backups according to the procedures outlined in the database documentation.

Java Virtual Machine

The Java Virtual Machine, or JVM, is a simulated computer that runs compiled Java programs. The JVM runs as a normal application inside a host operating system and supports a single process. Its purpose is to provide a platform-independent programming environment that abstracts away details of the underlying hardware or operating system and allows a program to execute in the same way on any platform, from servers to resource-limited embedded devices.

A JVM is distributed along with Java Class Library, a set of standard class libraries (in Java bytecode) that implement the Java application programming interface (API). These libraries, bundled together with the JVM, form the Java Runtime Environment (JRE).

Programs intended to run on a JVM must be compiled into Java bytecode, a standardized portable binary format which typically comes in the form of .class files (Java class files). A program may consist of many classes in different files. For easier distribution of large programs, multiple class files may be packaged together in a .jar file (short for Java archive).

SuperJ Framework

Hitachi's SuperJ Framework (SJF) is an R4.x-compliant OSGi Java program that can be deployed on CPE devices to manage and execute bundle(s) of applications, providing the bundle runtime environment.

The SJF is managed by the SJDMS and its management agent (SJMA) through the SJDMS API. OSGi bundles are wrapped as application services and deployed on the SJF OSGi framework. The SJF is a collaborative environment in which bundles run in the same VM and can actually share code. The SJF uses explicit imports and exports to group applications so they do not have to handle class loading.

The SJF also provides an API through which bundles can start, stop, and update other bundles, as well as enumerate the bundles and their service usage. This API is also used by the SJDMS and the SJMAto control OSGi frameworks.

The basic architecture of Hitachi's OSGi CPE solution (JVM, SJF, and SJMA bundle) is shown in the following figure:

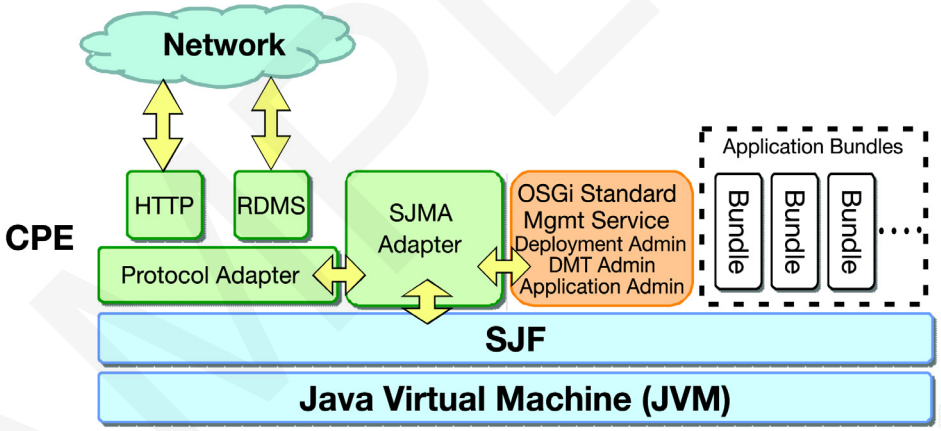

Figure 1-4: OSGi CPE solution architecture

Hitachi's SJF is optimized for smaller memory footprint and better performance (ex: start-up time). The SJF can run not only on Oracle's JVM but also on other Java ME CDC/Foundation level runtime environments, such as IBM J9.

The R4.x-compliant SJF supports:

- Remote Management via the SJMA
- OSGi-compliant CPE devices (type-agnostic)

Feature	Description
Communication protocol	The SJF uses HTTP (Hypertext Transfer Protocol)/HTTPS (Hypertext Transfer Protocol over Secure Socket Layer) for the communication protocol.
Management protocol	Since there is no standard protocol for OSGi management at this time, the SJF uses a proprietary protocol. Hitachi is tracking the various standards working to define a management protocol and will support the standard protocol after it has been approved.
Framework Resource Monitoring[a]	The SJF supports the retrieval and display of statistics for monitoring the health of the OSGi framework. The statistics include: • JVM Summary - JVM Name/Version - JVM Vendor - Number of active threads - Number of deamon threads - Total Java heap memory size - Free Java heap memory size - Operation System name - CPU architecture - Class path - Library path - Boot class path - Uptime • Resource usage per bundle - Number of total loaded classes - Number of current loaded classes - Number of active threads - Number of deamon threads - Number of created threads - Size of file storage used - Uptime

a. This feature is being developed for a future release.

SuperJ Tools

Hitachi's SuperJ Tools (SJT) is a J2ME Software Development Kit (SDK) that operates in an Ecplise-based environment for the development of OSGi bundles and applications.

Features of the SJT include:

- JVM emulation
- OSGi framework (SJF) emulation
- SJDMS management agent bundle emulation
- Permits emulation of specific end devices or CPE
- Wrapper required by the SJDMS for interoperability with the SJMA on the CPE's OSGi framework
- Debugging tools to help expedite development
- Resource monitoring tools
- Tools to group bundles into applications (packages) along with associated rules for dependencies and restrictions
- Rules configuration (dependencies and restrictions definitions)
- Application profiler (package creation including dependencies and restrictions)
- Libraries, sample code and documentation
- SJT installation package
- Application deployment package (groups bundles, defines rules and dependencies, etc.)
- SJDMS management agent bundle
- Optional library of common bundles available for ASPs

2 REQUIREMENTS

This chapter describes the requirements that you must meet to install the SuperJ™ Deployment Management System server software.

Contents

Topics covered in this document include: **Page**

SJDMS Server Requirements

This following sections describe the requirements for installing and running the SuperJ™ Deployment Management System on Sun Solaris:

Note: The requirements in this section are recommendations only. Operating system and SJDMS server requirements are dependent on the size of the network, the application load (the type of applications and management traffic generated), the user load (the type and number of clients and user traffic generated), and the required level of redundancy. Each system is unique and must be properly engineered and configured. Please contact Hitachi-CTA for your specific engineering requirements.

Operating System and Hardware Requirements

This section contains the minimum operating system and hardware requirements for installing and running the Client Console.

Minimum operating system requirements

Hitachi has tested its products with the Oracle Solaris 11 Express 2010.11, which was the most current release available at the time of testing.

Some operating system patches may be required if an Oracle 11g R2 database will be installed on the same computer.

IMPORTANT

If your requirements are to use an older version of Solaris, contact your Hitachi support representative for configuration assistance.

Minimum hardware requirements

The Client Console must be installed on Sun servers with the following specifications or better:

- 8 physical cores
- UltraSPARC T1, 1.2 GHz processor
- 32 GB of RAM

Server zoning and configuration requirements

The following table describes the zoning and configuration requirements that apply to the Sun servers on which you install the Client Console.

Table 2-1: Server zoning and configuration requirements (Page 1 of 2)

Requirement	Description
File system	ZFS
RAID	RAID-1 for each physical server
Ethernet	Minimum of one Ethernet connection set to 100MB/s full duplex or faster and configured in redundant mode. To avoid auto-negotiation problems, hard coding the speed to 100MB full duplex is highly recommended.

Table 2-1: Server zoning and configuration requirements (Page 2 of 2)

Requirement	Description
DNS	To enable NOC, SNAC and NIC client users to log in using either a DNS name or IP address, ensure that Client Consoles are added to the DNS.
Global zone	• 8 GB of swap space minimum • Enable the Fair Share Scheduler (FSS) • Make sure the Pools utility is running and set each pool to use the FSS subsystem. • Configure Automatic Lights Out Management (ALOM).
Non-global zone	• Configure one zone (non-global) for each Client Console instance to be installed. Figure 3-1 on page 3-11 shows a sample zone configuration. • CPU Share: equal weighting for all zones

Table 2-2 lists the resource budget guidelines for core, disk and Random Access Memory (RAM) requirements for each server tier instance. These are for planning purposes only.

Table 2-2: Guidelines for each server tier instance (zone)

Server instance	Cores	Disk size	Memory (RAM)
admin server	0.75	20 GB	4 GB
business server	0.75	20 GB	6 GB
web server	0.75	20 GB	4 GB
agent server	0.75	20 GB	4 GB
Oracle database server	2+	50 GB (for zone, database) additional requirement for data	4 GB

SJDMS Server port hardening requirements

The ports used by the SJDMS Server are listed in Table 2-3. If you reconfigure any of the services to use different ports, you must modify the OS hardening requirements accordingly.

Table 2-3: Port requirements (Page 1 of 2)

Port	Requirement
161	The default port number that an SNMP agent uses to listen to for incoming requests from SNMP managers. Can be configured during installation to any available port between 1024 and 65535.The SNMP agent is located on the admin server. See the *SJDMS Administrator's Guide* for the configuration procedure.
162	The default port that the agents bind to for SNMP traps from the NEs.
1521	The default Database connection port on the Oracle server (must be open in the direction from the business server to the Oracle server). This port value can be configured to your Oracle port value during installation of the admin server (NAKadmin) package. See the *SJDMS Administrator's Guide* for the procedure.
6666	Node manager process required by the web and business servers (must be open in the direction from the admin server to the managed servers)
7001	WebLogic console HTTP port on the admin server
7002	WebLogic console secure HTTPS port on the admin server
8011	Web server HTTP listening port
8012	Web server secure HTTPS port
1521	Default Oracle listener port. If you change any of the default SJDMS Server ports to use port 8080, you must disable the Oracle listener port.
9001	Business server HTTP listening port
9002	Business server secure HTTPS port (must be open in the direction from web to business server, and from each agent to the business server).
8998 (configurable)	secure HTTPS listening port on the agent
9082	RMI over SSL communication between server and the agents

Table 2-3: Port requirements (Page 2 of 2)

Port	Requirement
9090 to 9100	OSI stack services on the agent servers
between 1024 and 49151	Provisionable Session Broker FTP Data Connection ports. The actual provisioning of the ports is done through the Network Integrity Framework client interface using the Session Broker application. This can be configured to any available port between 1024 and 49151. For details, see the *SJDMS Administrator's Guide*.
provisionable file transfer ports	You must make ports available for file transfer (FTP, FTPS, or TFTP) from the agent servers. The actual provisioning of the ports is done through the SJDMS File Servers tool.

User account and permissions

To perform the installation of the Client Console, you can create an account to run (own) the installation prior to installing the packages, or you can let the installer create the account during the installation. You can choose the default account, “nakadmin”, or you can create a different account name to own the installation.

IMPORTANT

If you allow the installer to create the user account, the user account that gets created is not locked, so when the installation is complete, the system administrator must assign a password to the account.

Subnet requirements

Ensure that the following requirements are met:

- Admin server, business servers, web servers and agents can be deployed across multiple physical or logical servers.
- All business servers must be within one WebLogic cluster. All web servers must be within a second single WebLogic cluster.
- Business servers, web servers, and agent servers can reside in the same subnet, or run in different subnets.

Confirm routing between computers/virtual machines to be used in the SJDMS installation, as follows:

Business tier <-> Web tier

Business tier <-> Agent tier

Business tier <-> Admin tier

Web tier <-> Admin tier

Business tier <-> Oracle tier

Certificates for SSL

Agent tier server are installed with demo certificates that are used for Secure Socket Layer (SSL) communication. If you want to replace the demo certificates with your own, contact your Hitachi support representative for details.

Oracle WebLogic Licensing Requirements

The SJDMS Server installs Oracle WebLogic 11g (10.3.4) for its tiered server architecture.

Contact your administrator or your Hitachi sales representative for WebLogic licensing details.

Oracle Database Requirements

The SJDMS Client Console requires the use of an Oracle database. This section describes the Oracle database requirements in detail.

IMPORTANT

You are responsible for managing the database as described in the Oracle documentation.

Hardware and Oracle version recommendation

The SJDMS Client Console requires:

- Solaris 10 10/09 or later
- Oracle 11g Release 2

As illustrated in Figure 3-1 on page 3-11, Hitachi highly recommends the installation of the Oracle database on a separate physical server from the Client Console instances.

The Oracle server must have high-bandwidth connectivity to the business servers.

If the Oracle software is installed and maintained properly by a database administrator, in some configurations, Hitachi does support running Oracle in a separate zone on the same physical server as the SJDMS instances.

For more information on specific Oracle co-residency configurations that are supported for networks that do not require the full Hitachi solution scalability, please contact your Hitachi representative.

Database name (system ID) requirements

The Oracle database must use Locally Managed Tablespaces.

Create an Oracle System ID (SID) to uniquely identify the SJDMS Server database. The installation scripts use `oss` as the default SID. A different database name can be used, but make sure to record the new ORACLE_SID and have it available when prompted for the *Oracle database name to use* during the installation of the admin server.

Temp and init.ora settings

If you use the Database Creation Assistant (dbca), select the *General Purpose* template.

For a single SJDMS Server deployment with auto-extend turned off, the TEMP tablespace should be set to 24 GB. For a single SJDMS Server deployment with auto-extend turned on, the TEMP tablespace should be set to 12 GB. If you have multiple SJDMS Server systems accessing the same Oracle instance, multiply the TEMP tablespace requirement by the number of systems.

continued...▶

Temp and init.ora settings, continued

Table 2-4 shows the values for some of the required database parameters. The default values shown are for a General Purpose Database configuration.

Table 2-4: Values for parameters in init.ora

Parameter	Default value	Recommended value	Description
open_cursor	300	4000	Sets the limit of open cursors per session.
processes	150	1500	Limits the number of users that can concurrently access the instance.
Undotbs01.dbf	200 MB	500 MB	Contains one or more undo segments that maintain transaction history which is used to roll back, or undo, changes to the database.
Number of JDBC connections		200	The required number of Java Database Connectivity (JDBC) connections.
OPTIMIZER_INDEX_CACHING		90	Controls the costing of an index probe in conjunction with a nested loop.
OPTIMIZER_INDEX_COST_ADJ		25	Adjusts the cost of index probes

Tablespace and permission requirements

The Oracle database that you use for the SJDMS Server must use Locally Managed Tablespaces.

The SuperJ Deployment Management System requires two Oracle 11g tablespaces and two database users. For instructions, see “Configuring Oracle Tablespaces and User Accounts” on page 3-7.

You must have a user account with administrative permissions to set up the database.

By default, the installation scripts use user account `ossadmin` and password `ossadmin` for the SJDMS administration tablespace. You can use a different user and password, but if you do so, you should record this information. You will be prompted for the **Oracle user account** and **Oracle password** during the installation of the admin server.

continued...▶

Tablespace and permission requirements, continued

Table 2-5 provides tablespace configuration values.

Table 2-5: Tablespace configuration parameters

Parameter	Default value	Recommended value	Description
SJDMS administration tablespace (example: `ossadmin.dbf`)		20 GB minimum	For large networks contact your customer service representative for sizing requirements.
User role for ossadmin (or the tablespace you created)		Roles: DBA	Database user `ossadmin` for the tablespace you created must be granted the roles described on page 3-8.

For command details, see the *Oracle Performance Tuning Guide*.

The Oracle database users for the SJDMS must have the permissions listed in “Creating an Oracle user for the SJDMS Admin Console tablespace” on page 3-8 and “Creating an Oracle user for the SJDMS Client Console tablespace” on page 3-9.

Web Browser Requirements

Note: The Admin Console supports Internet Explorer 6 or higher, or Mozilla Firefox 3 or higher.

To use the WebLogic console for the Admin Console administration procedures, you must install the Java Runtime Environment (JRE) on the workstation running the browser. Ensure that the JRE is enabled in your browser.

3 PREPARING FOR INSTALLATION

This chapter describes how to meet the requirements for installing the SuperJ™ Deployment Management System server software. It also depicts an example installation using server zones.

Contents

Topics covered in this document include: **Page**

Overview of Preparation Steps

Before you install the SuperJ DMS Server, you must take steps to ensure that your installation environment is properly configured for the system. Use the following table as a guide.

Table 3-6: Installation Preparation Steps

Step	Refer to
Ensure that your server hardware, operating system, and database meet SJDMS requirements.	Chapter 2, Requirements
Ensure that you have access to the `root` user or to an administrator account with full read, write, and execute privileges in the applicable file structure on each server/zone on which you will install SJDMS.	

Ensure that domain resolution is set up for each SJDMS server/server zone.
• If using the `/etc/hosts` file (not DNS), set up IP address - domain mapping in that file.
• If using DNS, ensure that DNS is configured properly.
	Example:
	`10.133.9.40 server1 server1.acme.com`
	`10.133.9.41 oracle oracle.acme.com`
	`10.133.9.42 admin admin.acme.com`
	`10.133.9.43 bus1 bus1.acme.com`
	`10.133.9.44 web1 web1.acme.com`
	`10.133.9.45 agent0 agent0.acme.com`
	`10.133.9.46 agent1 agent1.acme.com`

Ensure that any previously installed SJDMS server processes are not running.	"Stopping the SJDMS Servers (if Previously Installed)" on page 3-3
Run the uninstall scripts to remove the installed files for each component.	"Uninstalling the SJDMS Server and Client Administration Software" on page 3-5
Copy the software installation files.	"Copying the Installer Files to the Hosts" on page 3-6
Create the two Oracle database tablespaces required for the SJDMS. Note: This should only be necessary in the initial installation.	"Configuring Oracle Tablespaces and User Accounts" on page 3-7
Remove the two Oracle users associated with the SJDMS tablespaces if they exist. This clears the SJDMS data from the database.	
Create the two Oracle users associated with the SJDMS tablespaces	

Stopping the SJDMS Servers (if Previously Installed)

This section provides procedures for stopping the agent, web, business, and admin servers in the SJDMS Server system. If you need to shut down all of the SJDMS servers, perform the shutdown sequence in the following order:

- shut down each of the agent servers
- shut down each of the web servers
- shut down each of the business servers
- shut down the admin server

Stopping an agent server

Agent server are not managed from the WebLogic console. Use this procedure to stop an agent server at the command line.

1. Log in as root to the server or zone hosting the agent server.
2. Change to the `/etc/init.d` directory.
3. Enter the following command to check the status of the server:

   ```
   ./nakinardmsAgent status
   ```

4. If the status report shows **`enabled`** `true` and **`state`** `online`, the server is running. Enter the following command:

   ```
   ./nakinardmsAgent stop
   ```

5. You can now log out of the agent server or zone.

Shutting down a web server

To shut down an SJDMS web server:

1. If you haven't already done so launch the Oracle WebLogic Administration Console by logging into the console using a browser at `https://<admin-server>:7002/console`

   with the username `Nakina` and the Password `NakinaSys`
2. In the left pane of the Console, expand **Environment** and select **Servers**.
3. Under **Summary of Servers**, select the **Control** tab.
4. In the **Summary of Servers** table, in the **Name** column, select the checkbox for the web server to shut down.
5. Click the **Shutdown** button above or below the table, and then select the **Force Shutdown Now** option.
6. Click the **Shutdown** button, then select the **Force Shutdown** option.
7. Wait for the **State** of the selected server to change to SHUTDOWN. Note: If the server was in the UNKNOWN state prior to SHUTDOWN, it will remain as UNKNOWN.
8. If you are shutting down additional servers, repeat from Step 4 for each web server.

Shutting down a business server

To shut down an SJDMS business server:

1. If you haven't already done so launch the Oracle WebLogic Administration Console by logging into the console using a browser at `https://<admin-server>:7002/console`

   with the username `Nakina` and the Password `NakinaSys`
2. In the left pane of the Console, expand **Environment** and select **Servers**.
3. Under **Summary of Servers**, select the **Control** tab.
4. In the **Summary of Servers** table, in the **Name** column, select the checkbox for the business server to shut down.
5. Click the **Shutdown** button above or below the table, and then select the **Force Shutdown Now** option.
6. Click the **Shutdown** button, then select the **Force Shutdown** option.
7. Wait for the **State** of the selected server to change to SHUTDOWN. Note: If the server was in the UNKNOWN state prior to SHUTDOWN, it will remain as UNKNOWN.
8. If you are shutting down additional business servers, repeat from Step 4 for each business server.

Stopping the admin server

To stop the SJDMS admin server, use the following procedure:

1. Log in as root to the server or zone hosting the admin server.
2. Change to the `/etc/init.d` directory.
3. Enter the following command to check the status of the server:

   ```
   ./nakinaadmin status
   ```
4. If the status report shows **`enabled`** `true` and **`status`** `online`, the server is running. Enter the following command:

   ```
   ./nakinaadmin stop
   ```
5. You can run the status command again to confirm that `enabled` is false and `state` is disabled.