Sun Java Enterprise System 5 Upgrade Guide for Microsoft Windows

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–5700–10
February 2007

070227@16599

Contents

Preface

The *Sun™ Java™ Enterprise System 5 Upgrade Guide for Microsoft Windows* contains the information you need to upgrade Sun Java Enterprise System (Java ES) software in a Windows Operating System. The Guide covers upgrade from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5).

This preface contains the following sections:

Who Should Use This Book

This book is intended for system administrators, or software technicians who wants to upgrade Java ES software. This guide assumes you are familiar with the following:

- Installation of enterprise-level software products
- Java ES components currently deployed in your environment
- System administration and networking on your supported Java ES platform

How This Book Is Organized

This guide includes the following chapters:

- Chapter 1 provides information for planning the upgrade of the Java ES software to Java ES 5.
- Chapter 2 provides information for upgrading Directory Server.
- Chapter 3 provides information for upgrading Directory Proxy Server.
- Chapter 4 provides information for upgrading Web Server.
- Chapter 5 provides information for upgrading Message Queue.

- Chapter 6 provides information for upgrading Application Server.
- Chapter 7 provides information for upgrading Service Registry.
- Chapter 8 provides information for upgrading Web Proxy Server.
- Chapter 9 provides information for upgrading Access Manager.

Related Books

The http://docs.sun.comweb site enables you to access the Sun technical documentation online. You can browse the archive or search for a specific book title or subject.

Books in This Documentation Set

The Sun Java Enterprise System manuals are available as online files in Portable Document Format (PDF) and Hypertext Markup Language (HTML) formats. Both formats are readable by assistive technologies for users with disabilities.

The Sun Java Enterprise System documentation includes information about the system as a whole and information about its components. This documentation can be accessed at http://docs.sun.com/prod/entsys.05q1.

The following table lists the system-level manuals in the Sun Java Enterprise System documentation set. The left column provides the name and part number location of each document and the right column describes the general contents of the document.

TABLE P–1 Java ES Documentation

Document Title	Contents
Sun Java Enterprise System 5 Release Notes for Microsoft Windows	Contains the latest information about Java ES, including known problems. In addition, components have their own release notes.
Sun Java Enterprise System 5 Technical Overview	Introduces the technical and conceptual foundations of Java ES. Describes components, the architecture, processes, and features.
Sun Java Enterprise System 5 Deployment Planning Guide	Provides an introduction to planning and designing enterprise deployment solutions based on Java ES. Presents basic concepts and principles of deployment planning and design, discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Java ES.

TABLE P–1 Java ES Documentation *(Continued)*

Document Title	Contents
Sun Java Enterprise System 5 Installation Planning Guide	Helps you develop the implementation specifications for the hardware, operating system, and network aspects of your Java ES deployment. Describes issues such as component dependencies to address in your installation and configuration plan.
Sun Java Enterprise System 5 Installation Guide for Microsoft Windows	Guides you through the process of installing Java ES on the Microsoft Windows operating system. Also shows how to configure components after installation, and verify that they function properly.
Sun Java Enterprise System 5 Installation Reference for UNIX	Gives additional information about configuration parameters, provides worksheets to use in your configuration planning, and lists reference material such as default directories and port numbers.
Sun Java Enterprise System Glossary	Defines terms that are used in Java ES documentation.

Accessing Sun Documentation

For product downloads, professional services, service packs and support, and additional developer information, refer to the following online resources:

- Download Center (`http://www.sun.com/software/download/`)
- Professional Services (`http:// http://www.sun.com/service/sunps/sunone/index.html`)
- Sun Enterprise Services, Windows Service Packs, and Support (`http://sunsolve.sun.com/`)
- Developer Information (`http://developers.sun.com/prodtech/index.html`)

The following location contains information about Sun Java Enterprise System and its components:

`http://www.sun.com/software/learnabout/enterprisesystem/`

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at `http://www.sun.com/documentation`.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–2 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.

TABLE P–2 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–3 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a seven digit or nine digit number that can be found on the title page of the guide or at the top of the document. For example, the part number of this book is **819–5700**.

◆ ◆ ◆ CHAPTER 1

Planning for Upgrades

This chapter provides information used for planning the upgrade of Sun Java Enterprise System (Java ES) software to Java ES 5 (Release 5) in a Windows operating system. This chapter contains the following sections:

Java ES 5 Components

As an introduction to planning the upgrade of Java ES software, this section reviews the components included in Release 5. Depending on your upgrade scenario, you might need to upgrade one or more of these components to their Release 5.

Java ES components are grouped into different types, as described in the *Java Enterprise System Technical Overview*:

- **Product Components.** Java ES product components consist of:
  - System service components, which provide the main Java ES infrastructure services
  - Service quality components, which enhance system services

  Product components are selectable with in the Java ES installer.
- **Shared Components.** Java ES shared components are locally shared libraries upon which Java ES product components depend. Shared components are installed automatically by the Java ES installer. Installation of shared components depends upon product components.

Release 5 Product Components

Release 5 product components are shown in the following table, listed alphabetically. For the service quality components among them, the table includes the type of service enhancement they provide.

TABLE 1–1 Java ES 5 Product Components

Product Component	Abbreviation	Version	Type
Access Manager	AM	7.1	System service component
Application Server	AS	8.2	System service component
Directory Proxy Server	DPS	6.0	Service quality: access component
Directory Server	DS	6.0	System service component
High Availability Session Store	HADB	4.4.3	Service quality: availability component
Java DB	JavaDB	10.1	System Service Component
Message Queue	MQ	3.7 UR1	System service component
Portal Server	PS	7.1	System service component
Portal Server Secure Remote Access	PSRA	7.1	Service quality: access component
Service Registry	SR	3.1	System service component
Web Proxy Server	WPS	4.0.4	Service quality: access component
Web Server	WS	7.0	System service component

Release 5 Shared Components

Java ES shared components, upon which the product components installed on a single computer depend, cannot be selected or deselected within the Java ES installer. When installing Java ES product components, the Java ES installer automatically installs the shared components needed by the installed product components.

Release 5 shared components are listed in the following table:

TABLE 1–2 Java ES 5 Shared Components

Shared Component	Version	Abbreviation
Apache Common Logging	1.0.3	ACL
Jakarta ANT Java/XML-based build tool	1.6.5	ANT
Common Agent Container	1.1 and 2.0	CACAO
FastInfoSet	1.0.2	FIS
International Components for Unicode	3.2	ICU
Java 2 Platform, Standard Edition	5.0 Update 7	J2SE™
JavaBeans™ Activation Framework	1.0.3	JAF
Java Studio Web Application Framework	2.1.5	JATO
JavaHelp™ Runtime	2.0	JHELP
JavaMail ™ Runtime	1.3.2	JMAIL
Java Architecture for XML Binding Runtime	2.0.3	JAXB
Java API for XML Processing	1.3.1	JAXP
Java API for XML Registries Runtime	1.0.8	JAXR
Java APIs for XML-based Remote Procedure Call Runtime	1.1.3_01	JAX-RPC
Java API for Web Services Runtime	2.0	JAXWS
Java Dynamic Management™ Kit Runtime	5.1.2	JDMK
Java Security Services	4.2.4 and 3.1.11	JSS and JSS3

TABLE 1–2 Java ES 5 Shared Components *(Continued)*

Shared Component	Version	Abbreviation
JSP Standard Library Template	1.0.6	JSTL
KT Search Engine	1.3.4	KTSE
LDAP C SDK	6.0	LDAP C SDK
LDAP Java SDK	4.19	LDAP J SDK
Mobile Access Core	6.2	MA Core
Netscape Portable Runtime	4.6.4	NSPR
Network Security Services	3.11.4	NSS
SOAP Runtime with Attachments API for Java	1.3	SAAJ
Simple Authentication and Security Layer	2.19	SASL
Sun Java Monitoring Framework	2.0	MFWK
Sun Java Web Console	3.0.2	SJWC
Web services Common Library	2.0	WSCL
XML Web Services Security	2.0	XWSS

About Java ES Upgrades

No single system utility upgrades all Java ES components. Instead, the upgrade of Java ES product components to Release 5 is performed component-by-component, computer-by-computer, using component-specific upgrade procedures documented in this guide.

The upgrade of a component can range from a major upgrade, which might not be compatible with the previous version of the component, to a fully compatible upgrade that simply provides bug fixes. Because of dependencies between Java ES components, the nature of the upgrade can affect whether other components need to be upgraded as well.

Product Component Upgrades

Java ES product component upgrades involve two basic operations that mirror the initial installation and configuration of Java ES product components:

- **Installation of software upgrades**. Upgrade software enhances or fixes existing software or replaces existing software. Software installation can be achieved through the application of patches to existing software packages, the selective replacement of existing packages, the installation of new packages, or a full reinstallation of component software.
- **Reconfiguration**. Reconfiguration encompasses any change in configuration data, user data, or dynamic application data needed to support the upgraded software. A change in data can mean additional data, a change in data format, whether in property files or database schema, or a change in data location. Sometimes reconfiguration requires that you perform an explicit procedure and sometimes it takes place automatically without your involvement. In some cases, reconfiguration also requires redeployment of component software to a web container.

In addition, Java ES product component upgrades normally involve pre-upgrade tasks and, in some cases, post-upgrade procedures before the upgrade is operational.

Product Component Upgrade Approaches

The upgrade of each product component involves one of the upgrade approaches described in the following sections:

- "Performing a Fresh Install of the Component" on page 19
- "Patching Existing Component Packages" on page 19

Performing a Fresh Install of the Component

Java ES 5 product components are upgraded by performing a fresh install of the components using the Java ES installer. You should install Release 5 in a parallel path and leave the previous version intact. You can reconfigure the product component by migrating the previous version's configuration data to the new installation, or by performing a new configuration, or by doing a combination of both. For some product components a utility is provided for reconfiguring or migrating configuration data for the component.

Patching Existing Component Packages

Web Proxy Server upgrade is performed by manually patching the existing software packages. For more information, see "To Upgrade Web Proxy Server" on page 80.

Upgrade Approach Used for Each Product Component

The upgrade approach used to upgrade each product component to Release 5 is shown in the following table:

TABLE 1–3 Java ES Product Component Upgrade Approaches

Component	Upgrade Approach	Reconfiguration
Access Manager	Perform fresh install in a parallel path using Java ES installer	Use `amconfig.bat` and `amupgrade.bat` files to reconfigure and redeploy to web container
Application Server	Perform fresh install in a parallel path using Java ES installer	None
Directory Proxy Server	Perform fresh install in a parallel path using Java ES installer	Manual reconfiguration
Directory Server	Perform fresh install in a parallel path using Java ES installer	Use `dsmig` command to migrate Directory Server data
Message Queue	Perform fresh install in a parallel path using Java ES installer	None
Service Registry	Perform fresh install in a parallel path using Java ES installer	Manual reconfiguration
Web Proxy Server	Patch binaries	None
Web Server	Perform fresh install in a parallel path using Java ES installer	Use `wadm migrate-server` command to migrate server instance configuration

Shared Component Upgrades

Java ES shared component upgrades are a necessary part of upgrading the product component that depend on them. Shared components for Release 5 need to be installed using the Release 5 installer in a parallel path. Release 5 installer does not upgrade Release 4 shared components.

Upgrade Process

The Java ES upgrade process involves a number of phases, which are normally carried out first in a staging environment, before being executed in a production environment. The use of a staging environment allows you to test each phase as well as write scripts to be used by IT personnel for upgrading complex Java ES deployments.

When you have tested the upgrade process in a staging environment, and have confidence that the upgrade is working properly, you can reproduce the process in your production environment. The process involves the phases shown in the following table and documented in this Upgrade Guide. The phases apply to individual component upgrades as well as to your Java ES deployment as a whole.

TABLE 1–4 Phases in the Upgrade Process

Upgrade Phase	Description
Plan	You develop an upgrade plan. In the development plan, you specify the Java ES components to be upgraded and the sequence by which you need to upgrade those components on the different computers or operating system instances in your deployment.
Pre-upgrade preparation	You back up configuration and application data, perform any patching of the operating system, upgrade any required dependencies, and perform other tasks in preparation for upgrading any individual component.
Upgrade	You obtain all the necessary packages, patches, and tools needed for the upgrade. You install upgraded software and reconfigure each component as prescribed, including the migration of data to the upgraded system.
Verification	You verify that the upgrade has been successful using prescribed verification tests, including starting the upgraded software components and testing various usage scenarios.
Rollback and restoration	Roll back the upgrade and verify that the rollback is successful. Testing the rollback of the upgrade is important in case you have to restore the production environment to its previous state for some reason.

Upgrade Plan Considerations

In an upgrade plan you specify the Java ES components you will upgrade to Release 5 and the sequence by which you will upgrade those components on the different computers or operating system instances in your Java ES deployment.

Your plan will depend on your upgrade objectives and priorities, as well as the scope and complexity of your deployment architecture.

For example, your Java ES deployment architecture might consist of a single Java ES component running on a single computer, and your upgrade objective is to fix some bug in the previous software release. On the other hand, your Java ES deployment architecture might consist of a number of interdependent Java ES components deployed across a number of different computers, and your upgrade objective is to achieve some new functionality by upgrading the minimum number of components required to achieve that end with minimal downtime.

In general, the greater the number of Java ES components and the greater the number of computers in your deployment architecture, the more complex your upgrade plan will be.

However, your upgrade plan will depend on a number of considerations other than the scope and complexity of your deployment architecture. These considerations include the following factors:

- Upgrade Dependencies
- Upgrade All
- Supported Upgrade Paths and Strategies

Upgrade Dependencies

One of the main issues in planning the upgrade of any given Java ES component is that component's dependencies on other Java ES components. You should evaluate whether such other components also need to be upgraded to support the upgrade of the dependent component.

The two types of upgrade dependencies are:

- **Hard upgrade dependency**. Upgrade of a product component requires you to upgrade a component upon which it depends. This requirement can be due to new functionality, new interfaces, or bug fixes needed by the dependent component. With a hard upgrade dependency, you cannot successfully upgrade and use the dependent component without first upgrading the component upon which it depends.
- **Soft upgrade dependency**. Upgrade of a product component does not require you to upgrade the component upon which it depends. With a soft upgrade dependency, you can successfully upgrade and use the dependent component without upgrading the component upon which it depends.

Upgrading a Java ES product component requires you to upgrade all the components upon which it has hard upgrade dependencies, but, with some exceptions noted in this book, allows you to not upgrade components upon which it has soft upgrade dependencies. When multiple interdependent components are involved in an upgrade, you have to upgrade a component if only one of the Java ES components being upgraded has a hard upgrade dependency on that particular component.

In a few special cases, due to incompatibilities that are introduced, upgrade of a component requires you to also upgrade a component that it supports. These special cases are noted in this book.

Upgrade All

- **Upgrade All**. In this approach you upgrade all deployed Java ES product components to Release 5. In some cases, due to the complexity of a deployment, upgrading an entire system at one time is not feasible for business reasons.

  The two approaches to performing upgrades are compared in the following table:

TABLE 1–5 Upgrade All

Upgrade Approach	Advantages	Disadvantages
Upgrade All	Maintains a consistent version for all components in your deployed system	Maximizes the number of components to upgrade

The choice between Selective Upgrade and Upgrade All is not rigid. For example, you might choose to selectively upgrade the product components on a particular computer, but wish to upgrade all shared components needed to support the selected product components.In fact, for upgrades from Release 4 to Release 5, selectively upgrading product components, while upgrading all of the corresponding shared components, is often the preferred approach.

Supported Upgrade Paths and Strategies

Your upgrade plan depends on the Java ES release you wish to upgrade to Release 5. Java ES installer only support upgrade from Java ES 2005Q4 (Release 4). The following table describes the different upgrade paths to Release 5, their characteristics, and the upgrade strategies to be used in performing the upgrade.

TABLE 1–6 Upgrade Paths and Strategies

Release	Java ES Release	System Characteristics	Upgrade Strategies
2005Q4	Release 4	Java ES 5 supports a mixture of Release 4 and Release 5 product components on a single computer. Interoperability between Release 4 and Release 5 product components has been tested, and known interface incompatibilities are noted in the *Sun Java Enterprise System 5 Release Notes for Microsoft Windows*.	The coexistence of Release 4 and Release 5 product components provides for the possibility of selectively upgrading Release 4 product components to Release 5 on a single computer or within a deployment architecture consisting of multiple computers. Release 5 installer automatically installs all required shared components.

Java ES Component Dependencies

One of the most important considerations in an upgrade plan is the dependencies between the various Java ES components in your deployed system. The sequence in which you perform the component upgrades is affected by the nature of the dependencies between them.

- "Dependencies on Shared Components" on page 24
- "Dependencies On Product Components" on page 26

Each of these factors is discussed briefly in the following sections.

Dependencies on Shared Components

Table 1–7 shows the dependencies of Release 5 product components on Java ES shared components. The abbreviations for product components in the table are taken from Table 1–1. The abbreviations for shared components are spelled out in Table 1–2.

Within the matrix of the following table hard upgrade dependencies for Release 4 to Release 5 upgrades are marked "H," and soft upgrade dependencies are marked "S."

TABLE 1–7 Shared Component Dependencies of Java ES 5 Product Components

Shared Component	AM	AS	DPS	DS	MQ	SR	WS	WPS
ANT		S				H		
ACL	S					H		
BDB	S							
Common Agent Container	H	H	H	H				
ICU		S	H	H			S	S
J2SE	S	S	H	H	S	H	S	S
JAF	S	S				H		
JATO	S	S						
JavaHelp	S	S			S		S	
JavaMail	S	S				H	S	
JAXB	S	S					S	
JAXP	S	S				H	S	
JAXR	S	S				H	S	
JAX-RPC	S	S				H	S	
JAXWS							S	
JCAPI								
JDMK	H	S	H	H			S	
JSS	S						S	S
KTSE							S	S
LDAP C SDK	H			H			S	S
LDAP J SDK	S							
MA Core	S							
MFWK	H							
NSPR	S	S	H	H	H		H	S
NSS	S	S		H	H		H	S
SAAJ	S	S				H		

TABLE 1–7 Shared Component Dependencies of Java ES 5 Product Components *(Continued)*

Shared Component	AM	AS	DPS	DS	MQ	SR	WS	WPS
SASL				H			S	S
SJWC	S	S	H	H				
WSCL	S	S				H	S	
XWSS						H		

Dependencies On Product Components

Dependencies on product components fall into two general categories: runtime dependencies and configuration dependencies.

- **Runtime Dependencies.** The functioning of a software system is based on the interactions between its deployed components. The infrastructure dependencies between Java ES components are discussed in the *Java Enterprise System 5 Technical Overview*. If a Release 5 product component has a hard upgrade dependency on another product component, the dependent component con only be successfully upgraded and used if the component upon which it depends is also upgraded.
- **Configuration Dependencies**. In some cases a Java ES component must be installed, configured, and running in order for another component to be configured. For example, a Directory Server user directory must be running for an Access Manager service to be registered. Component upgrade procedures often involve reconfiguration of upgraded components or migration of configuration data. Configuration dependencies can impact the sequence of upgrade procedures.

For runtime dependencies, the relationship between product components can be of the following three types:

- **Mandatory.** The component cannot operate without the supporting component.
- **Optional.** The component can operate without the supporting component, but a subset of its functionality requires the supporting component.
- **Co-dependency.** Both components can operate without the support of the other, but the components together can provide certain enhanced functionality or performance.

The following table shows the dependencies between the Java ES product components listed in Table 1–1. The information can be used to determine the hard upgrade dependencies that impact your upgrade plan.

The first column alphabetically lists Release 5 product components, the second column shows other Java ES components upon which a Release 5 component has a dependency relationship, the third column provides the Java ES release versions that support the Release 5 dependency, the fourth column characterizes the dependency relationship, and the last column indicates

special characteristics of the dependency, such as whether the supporting component must be local as opposed to remote or whether other third-party products can support the dependency.

If a product component you are upgrading to Release 5 has a dependency on Release 5 of a supporting component then the supporting component represents a hard upgrade dependency: the supporting component must also be upgraded to Release 5.

TABLE 1–8 Java ES Product Component Dependencies

Product Components	Dependencies	Nature of Dependencies	Must be Local
Access Manager	Directory Sever	Mandatory: Stores configuration data and enable lookup of user data	
	Java 2 Enterprise Edition (J2EE™) web container: ■ Application Server ■ Web Server	Mandatory: Provides web container runtime services	Yes
Access Manager SDK	Access Manager	Mandatory: Provides Access Manager services	
	J2EE web container: ■ Application Server ■ Web Server	Mandatory: Provides web container runtime services	Yes
Access Manager Distributed Authentication	Access Manager	Mandatory: Provides Access Manager services	
	J2EE web container: ■ Application Server ■ Web Server	Mandatory: Provides web container runtime services	Yes
Access Manager Session Failover	Access Manager	Mandatory: Provides Access Manager services	
	J2EE web container: ■ Application Server ■ Web Server	Mandatory: Provides web container runtime services	Yes
Application Server	Message Queue	Mandatory: Provides reliable asynchronous messaging	Yes
	Web Server	Optional: Provides load balancing between instances	Yes
	High Availability Session Store	Mandatory: Stores session state needed to support failover between instances	Yes

TABLE 1–8 Java ES Product Component Dependencies *(Continued)*

Product Components	Dependencies	Nature of Dependencies	Must be Local
Directory Proxy Server	Directory Server	Co-dependency: Results in improved security and performance for directory requests. Supplies data to Directory Proxy Server	No
Directory Server	Directory Proxy Server	Co-dependency: Results in improved security and performance for directory requests. Distributes load and caches data to Directory Server	
Message Queue	Directory Server	Optional: Stores administered objects and persistent data	
	J2EE web container: ■ Application Server ■ Web Server	Optional: Supports HTTP transport between client and Message Queue broker	
Service Registry	Application Server	Mandatory: Provides container runtime services	Yes
	Java DB	Mandatory: Provides default database for string services and related meta data	Yes
	Service Registry Client	Mandatory: Provides required client libraries	Yes
Web Proxy Server	Directory Server	Optional: Provides LDAP-based authentication	
	Web Server	Co-dependency: Results in improved security and performance for HTTP requests. Supplies data to Web Proxy Server	Yes
Web Server	Directory Server	Optional: Provides LDAP-based authentication	
	Web Proxy Server	Co-dependency: Results in improved security and performance for HTTP requests. Distributes load and caches data from Web Server	

Upgrade Sequencing Guidelines

The following listing provides the order in which Java ES components can be successfully upgraded on a single computer or in a deployed system. When you plan your upgrade, you can omit those components that are not part of your deployment architecture.

The chapters in this guide are arranged according to the order in which components appear in the following listing.

1. **Directory Server** (Chapter 2)

   Many components store user data or configuration data in Directory Server, so upgrades to Directory Server should generally be performed before upgrading the components that have runtime or configuration dependencies on Directory Server.

2. **Directory Proxy Server** (Chapter 3)

   Directory Proxy Server has a hard upgrade dependency on Directory Server and is therefore upgraded after Directory Server. Other components might access Directory Server through Directory Proxy Server.

3. **Web Server** (Chapter 4)

   A number of Java ES components require the support of a web container, which, if upgraded, should be upgraded before the components requiring web container services. Normally web container services are provided by Web Server, but if your architecture contains both, upgrade Web Server first.

4. **Message Queue** (Chapter 5)

   Message Queue, if upgraded, is best upgraded before Application Server, which requires Message Queue to be Java 2 Enterprise Edition (J2EE) compliant.

5. **Application Server** (Chapter 6)

   Application Server depends on Web Server for its load balancing plug in, so if you are using that capability, Application Server should be upgraded after Web Server.

6. **Service Registry** (Chapter 7)

   Service Registry can be upgraded anytime after Application Server is upgraded because it depends upon Application Server for runtime container services.

7. **Web Proxy Server** (Chapter 8)

   Web Proxy Server can be upgraded anytime, though generally it would be upgraded after the Web Server or Application Server component for which it provides a proxy service. Web Proxy Server is a new Java ES Release 5 component that can be upgraded from its previous non-Java ES release.

8. **Access Manager** (Chapter 9)

Access Manager plays a central role in authentication and authorization, including single sign-on. If upgraded, Access Manager should be upgraded before the components that depend on it for those services. In addition, Access Manager requires specific Directory Server schema (Schema 2), which affects how other components use Directory Server.

◆ ◆ ◆ CHAPTER 2

Directory Server

This chapter describes how to upgrade Directory Server to Java ES 5: Sun Java System Directory Server 6.0.

The chapter provides an overview of upgrade issues and procedures for the different upgrade paths supported by Release 5. This chapter covers the following topics:

Note – File locations in this chapter are specified with respect to a directory path referred to as *Server-Root* (Directory Server 5.*x*) and *DirectoryServer-base* (Directory Server 6.0). At least part of this path might have been specified as an installation directory when Directory Server was initially installed. If not, the Java ES installer assigned a default value.

The default value of *Server-Root* is *Install-dir*`\Server_Root` and *DirectoryServer-base* is `C:\Program Files\Sun\JavaES5\DSEE`.

Overview of Directory Server Upgrade

This following sections describe general aspects of Directory Server that affect upgrading to Java ES 5:

About Java ES 5 Directory Server

Java ES 5 versions of Directory Server represents a major release, with a variety of new features and improvements. For more information, see the *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*.

Java ES 5 Directory Server Upgrade Roadmap

Because the Java ES Release 5 version of Directory Server represents a major release, you install the new Directory Server version and migrate user and configuration data from the previous Directory Server version. For migrations instructions to the new version, see the *Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide*.

The following table shows the supported Directory Server upgrade path to Java ES 5:

TABLE 2–1 Upgrade Paths to Java ES 5 Directory Server

Java ES Release	Directory Server Release	General Approach	Reconfiguration Required as Part of Upgrade
Release 4	Sun Java System Directory Server 5.2 2005Q4	Direct upgrade: Fresh install and migration of all data.	Configuration data migrated from previous version to newly installed Directory Server
	Sun Java System Administration Server 5.2 2005Q4	Administration Server functionality replaced by Directory Service Control Center and Directory Server EE command-line utilities	

Directory Server Data

Directory Server 5.*x* versions made use of Directory Server itself for storing configuration data. The data was stored in a specific tree structure within the directory. The Directory Server instance that hosted the configuration was referred to as the configuration directory. The configuration directory could reside on the same computer as other Directory Server instances. However, in most deployment architectures, the configuration directory was remote from the other components that use it to store configuration information.

Directory Server 6.0 no longer stores configuration data in a configuration directory. Configuration is performed using the Directory Service Control Center or the Directory Server EE command-line utilities. The configuration should be accessed through this interface.

The following table shows the type of data that could be impacted by an upgrade of Directory Server software.

TABLE 2–2 Directory Server Data Usage

Type of Data	Location	Usage
Directory Server configuration data	Directory Server 5.x: configuration directory Directory Server 6.0: accessed through Directory Service Control Center and Directory Server EE command-line utilities	Configuration of Directory Server
Directory Server schema		Define structure and semantics of data in the directory
Security data	Directory Server 5.*x*: SSL configured through Directory Server Console. Directory Server 6.0: SSL configured through Directory Service Control Center and Directory Server EE command-line utilities	Server certificates
User data	Directory Server	Server certificates

Directory Server Compatibility Issues

Release 5 Directory Server does not introduce any interface changes and is therefore backwardly compatible with earlier versions. However, Directory Server supports all components supported by Release 4 Directory Server and earlier versions.

Release 5 introduces changes to private administrative interfaces that make Release 5 incompatible with earlier releases of Directory Server. Details can be found in the *Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide*. In particular, the Administration Server, used to configure earlier Directory Server instances, has been replaced by the Directory Service Control Center and Directory Server EE command-line utilities. The `o=NetscapeRoot` directory suffix for storing Directory Server configuration information has been eliminated.

Directory Server Dependencies

Dependencies on other Java ES components can affect the procedure for upgrading Directory Server software. Directory Server has dependencies on specific Java ES shared components, as listed in Table 1–7.

Upgrading Directory Server from Java ES Release 4

This section includes information about upgrading Directory Server from Release4 to Release 5. This section covers the following topics:

Introduction to Upgrading Directory Server

When upgrading Release 4 Directory Server to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is achieved by performing a fresh install of Release 5 Directory Server and then using migration tools to recreate the previous Directory Server instances in new, distinct Release 5 Directory Server instances.
- **Upgrade Dependencies**. Directory Server has dependencies on a number of Java ES shared components, as listed in Table 1–7. Shared components are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Directory Server.
- **Backward Compatibility**. Release 5 Directory Server is not backwardly compatible with the Release 4 version. However, you can use the migration tools to migrate the `o=NescapeRoot` suffix if you continue to maintain a set of Directory Server instances relying on the Directory Server 5.*x* administration framework.
- **Upgrade Rollback.** You can do a rollback of the Release 5 upgrade is achieved by reverting to the previous version, which is left intact by the upgrade.

Directory Server Upgrade

This section describes how to perform an upgrade of Directory Server from Release 4 to Release 5. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Directory Server, perform the tasks described below.

Upgrade Directory Server Dependencies

All Java ES components on a computer system and in a computing environment should be upgraded to Release 5. However, all shared components required by Directory Server are upgraded automatically when you perform an upgrade of Directory Server to Release 5.

Create Directory Server Image (Optional)

In cases where Release 5 Directory Server is being installed on a computer different from where the Release 4 version resides, an image of the Release 4 version should be created on the computer where Release 5 Directory Server is being installed. The image is needed to automate data migration (using the `dsmig.bat`) to the new Release 5 Directory Server instances.

Release 4 image includes all schema files, configuration files, security files, and database files, in an identical layout to the original Directory Server 5.x file structure. The image is needed to perform data migration to the new Release 5 Directory Server instances.

▼ To Verify Current Version Information

1. **Go to** *Server-Root*`\bin\slapd-`*hostname*`\server`

2. **Type the following command:**

   ```
   ns-slapd -v
   ```

   The message that indicate the Java ES release version are:

Release 5	Sun Java System Directory Server 6.0
Release 4	Sun Java System Directory Server 5.2_Patch_4

Upgrade Considerations

When upgrading Directory Server software to Java ES Release 5 you should take into account the following considerations:

- In a deployment architecture in which there are multiple instances of Directory Server running on a single computer and all instances correspond to the same installed Directory Server image, upgrading the Directory Server image will upgrade all the instances. In such architectures, there is only one Administration Server instance per installed Directory Server image.
- In many Release 4 deployment architectures the configuration directory is a separate Directory Server instance. These instances do not need to be upgraded because the configuration directory has been deprecated in Release 5. On the other hand, the upgrade might entail the deployment of the Release 5 Directory Server administrative console (the Directory Service Control Center) to a separate computer from which you remotely manage Directory Server instances.

To Upgrade Directory Server to Java ES Release 5

Before You Begin

- Obtain the Directory Server administrator user ID and password for your currently installed version. Other configuration information is preserved through the upgrade process.
- Any Java ES components using a Directory Server instance should be shut down and re-configured to access the corresponding new Release 5 instance.

1. **Log in as administrator.**
2. **Shut down the Release 4 Directory Server (5.2) instances.**
3. **For remote install of Release 5, create a Release 4 image and transfer it to the remote computer, see "Create Directory Server Image (Optional)" on page 35.**

   Note – If the old and the new versions of the Directory Server have been installed on two different machines, an image of the old Directory Server must be created on the same machine where the new Directory Server is installed.

4. **Make sure you have upgraded any Java ES components upon which Directory Server and Administration Server have hard upgrade dependencies.**

   For more information, see "Upgrade Directory Server Dependencies" on page 35
5. **Install Release 5 Directory Server.**

   For Java ES 5 Directory Server installation instructions, see the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.
6. **Create a Directory Server instance.**

   For information about creating a Directory Server instance, see the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.
7. **Migrate Release 4 data to the Release 5 Directory Server instance.**

   For more information about the migration process, see the *Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide*.

To Verify the Upgrade

1. **Log in as administrator.**
2. **Start the new Directory Server instance.**

   DirectoryServer-base`\ds6\bin\dsadm -V`

3 **Check the startup messages in the** *instance-path*`\logs\errors` **Directory Server error log.**

The message that indicate the Java ES release version are:

Release 5 Sun Java System Directory Server 6.0

Release 4 Sun Java System Directory Server 5.2_Patch_4

Post-Upgrade Tasks

All Java ES components dependent on Directory Server need to be re-configured to point to the new Directory Server instances.

Rolling Back the Upgrade

A rollback of the Release 5 upgrade is achieved by reverting to the previous version, which is left intact by the upgrade to Release 5.

◆ ◆ ◆ CHAPTER 3

Directory Proxy Server

This chapter describes how to upgrade Release 4: Directory Proxy Server 5.2 to Release 5: Directory Proxy Server 6.0.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Release 5. This chapter covers the following topics:

Note – File locations in this chapter are specified with respect to directory paths referred to as*Server-Root* (Directory Proxy Server 5.*x*) and *DirectoryProxyServer-base* (Directory Proxy Server 6.0). At least part of these paths might have been specified as installation directories when Directory Proxy Server was installed. If not, the Java ES installer assigned a default value.

The default value of *Server-Root* is *Install-dir*`\Server_Root` and *DirectoryProxyServer-base* is `C:\Program Files\Sun\JavaES5\DSEE`.

Overview of Directory Proxy Server Upgrades

The following sections describe general aspects of Directory Proxy Server that affect upgrading to Release 5:

About Java ES 5 Directory Proxy Server

Java ES Release 5 Directory Proxy Server represents a major release. Release 5 Directory Proxy Server is still an LDAP proxy, but with new, extensible routing capabilities. Release 5 also enables the Virtual Directory feature, the ability to aggregate multiple data views in a single view. These data views can represent LDAP or SQL accessible data stores.

For more information, see the *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*.

Java ES 5 Directory Proxy Server Upgrade Roadmap

Table 3–1 shows the supported Directory Proxy Server upgrade paths to Release 5:

TABLE 3–1 Upgrade Paths to Java ES 5: Sun Java System Directory Proxy Server 6

Java ES Release	Directory Proxy Server Release	General Approach	Reconfiguration Required
Release 4	Sun Java System Directory Proxy Server 5.2 2005Q4	Direct upgrade: Replace Release 4 with a fresh install and configuration of Release 5.	If backward compatibility desired, manually map previous configuration to new configuration

Directory Proxy Server Data

Directory Proxy Server no longer uses Directory Server for storing configuration data. Configuration is performed using the new Directory Service Control Center or Directory Server EE command-line utilities.

The following table shows the type of data that could be impacted by an upgrade of Directory Proxy Server software.

TABLE 3–2 Directory Proxy Server Data Usage

Type of Data	Location	Usage
Directory Proxy Server configuration data	Directory Proxy Server 5.*x*: configuration directory Directory Proxy Server 6.0: Accessed through Directory Service Control Center and Directory Server EE command-line utilities	Configuration of Directory Proxy Server

TABLE 3–2 Directory Proxy Server Data Usage *(Continued)*

Type of Data	Location	Usage
Security data	Directory Proxy Server 5.*x*: SSL configured through Directory Proxy Server Console Directory Proxy Server 6.0: SSL configured through Directory Service Control Center and Directory Server EE command-line utilities	Server certificates

Directory Proxy Server Compatibility Issues

Release 5 Directory Proxy Server introduces interface changes that make it incompatible with earlier Directory Proxy Server releases. Release 5 Directory Proxy Server is based on a completely new Java-based implementation and its configuration differs fundamentally from Release 4 Directory Proxy Server, as well as earlier releases.

However, you can to configure Release 5 Directory Proxy Server to be backwardly compatible, that is, to behave like Release 4 Directory Proxy Server and earlier releases. This configuration requires you to manually map previous configuration attributes to Release 5 configuration properties. For more information, see the *Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide*.

Directory Proxy Server Dependencies

Dependencies on other Java ES components can impact the procedure for upgrading and reconfiguring Directory Proxy Server software. Directory Proxy Server has dependencies on specific Java ES shared components, as listed in Table 1–7. Directory Proxy Server provides front-end access to Directory Server but has no dependency on Directory Server beyond this functional relationship.

Upgrading Directory Proxy Server from Java ES Release 4

This section includes information about upgrading Directory Proxy Server from Release 4 to Release 5. This section covers the following topics:

Introduction to Upgrading Directory Proxy Server

When upgrading Release 4 Directory Proxy Server to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach**. The upgrade is achieved by performing a fresh install of Release 5 Directory Proxy Server and then configuring new Directory Proxy Server instances using the Directory Service Control Center or Directory Server EE command-line utilities.
- **Upgrade Dependencies**. Directory Proxy Server has dependencies on a number of Java ES shared components, see Table 1–7, all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Directory Proxy Server.
- **Backward Compatibility.** Release 5 Directory Proxy Server can be configured to be backwardly compatible with Release 4.
- **Upgrade Rollback.** A rollback of the Release 5 upgrade is achieved by reverting to the previous version, which is left intact by the upgrade to Release 5.

Directory Proxy Server Upgrade

This section describes how to perform an upgrade of Directory Proxy Server from Release 4 to Release 5. This section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Directory Proxy Server, you should perform the tasks described below.

Upgrading Directory Proxy Server Dependencies

All Java ES components on a computer system and in a computing environment should be upgraded to Release 5.

When upgrading Directory Proxy Server dependencies, you should do so in the order below , before you upgrade Directory Proxy Server. Upgrade of shared components is normally achieved automatically by the Java ES installer.

- **Shared Components**. Release 5 installer automatically installs all required shared components in the parallel path.
- **Directory Server (optional)**. Instructions for upgrading Directory Server to Release 5 are provided in Chapter 2. However, Release 5 Directory Proxy Server is supported by Release 4 Directory Server.

▼ To Verify Current Version Information

1 **Start the new Directory Proxy Server instance.**

DirectoryServer-base`\dps6\bin\dpadm start` *instancePath*

2 **Check for the Directory Proxy Server version.**

DirectoryServer-base`\dps6\bin\dpadm --version`

The outputs that indicate the Java ES Directory Proxy Server release version are:

Release 5 Sun ONE Directory Proxy Server Version 6.0

Release 4 Sun ONE Directory Proxy Server Version 5.2_Patch_4

Upgrade Considerations

When upgrading Directory Proxy Server software to Java ES Release 5 you should take into account the following considerations:

- In a deployment architecture in which there are multiple instances of Directory Proxy Server running on a single computer and all instances correspond to the same installed Directory Proxy Server image, upgrading the Directory Proxy Server image will require you to create new Directory Proxy Server instances.
- In Release 4 deployment architectures involving Directory Proxy Server, an Administration Server was used to configure and manage Directory Proxy Server instances. In Release 5 the upgrade of Directory Proxy Server might entail deployment of the Directory Service Control Center, used to configure and manage Directory Proxy Server instances.

▼ To Upgrade Directory Proxy Server

Before You Begin

- Directory Proxy Server must run as the same user as Directory Server and Administration Server.
- Any Java ES components using a Directory Proxy Server instance should be shut down and re-configured to access the corresponding new Release 5 instance.

1 **Log in as administrator.**

2 **Shut down all Java ES components dependent on the Directory Proxy Server instances that are to be upgraded.**

This step might depend on how Directory Proxy Server is replicated within your deployment architecture.

For information about how to shut down a Java ES component, see its respective administration guide.

3. **Install Release 5 Directory Proxy Server.**

   For more information on Java ES 5 Directory Proxy Server installation instructions, see the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

4. **Create a Directory Proxy Server instance.**

5. **(Optional) Map Release 4 configuration attributes to the Release 5 Directory Proxy Server properties.**

   For more information on mapping procedure, see the *Directory Server Enterprise Edition 6 Migration Guide*.

6. **All Java ES components dependent on Directory Proxy Server need to be re-configured to point to the new Directory Proxy Server instances.**

▼ To Verify the Upgrade

1. **Change to the** *Server-Root*`\bin\dps\server\bin` **directory.**

2. **Type the following command:**

   ```
   ldapfwd -v
   ```

   The outputs that indicate the Java ES Directory Proxy Server release version are:

   Release 5 Sun ONE Directory Proxy Server Version 6.0

   Release 4 Sun ONE Directory Proxy Server Version 5.2_Patch_4

Rolling Back the Upgrade

A rollback of the Release 5 upgrade is achieved by reverting to the previous version, which is left intact by the upgrade to Release 5.

◆ ◆ ◆ CHAPTER 4

Web Server

This chapter describes how to upgrade Web Server software from previous Java ES versions to Java ES 5 (Release 5): Web Server 7.0.

This chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Release 5. This chapter covers the following sections:

Note – File locations in this chapter are specified with respect to a directory path referred to as *WebServer-base*. At least part of this path might have been specified as an installation directory when Web Server was initially installed. If not, the Java ES installer assigned a default value.

The default values of these directory paths are:

WebServer6-base	`C:\Sun\WebServer`
WebServer7-base	`C:\Program Files\Sun\JavaES5\webserver7`
WebServer7Config-base	`C:\Program Files\Sun\JavaES5\webserver7`

Overview of Web Server Upgrades

The following sections describe general aspects of Web Server that impact upgrading to Release 5:

About Java ES 5 Web Server

Java ES Release 5 Web Server represents a major release with respect to Release 4. It has a number of new features and interface enhancements.

Release 5 Web Server has a new administrative infrastructure with new administrative tools. The administrative infrastructure includes an Administration Server instance which hosts configuration information for any number of Web Server instances. A new command line interface (`wadm`) and new graphical user interface are used to create Web Server instances, either locally or on remote computers, and to configure and manage these instances. The new administrative tools require an administrator user name and password.

For more information about the new administrative infrastructure, see the *Web Server 7.0 Administrator's Guide*.

Web Server Upgrade Roadmap

The following table shows the supported Web Server upgrade paths to Java ES Release 4.

TABLE 4–1 Upgrade Paths to Java ES 5 Web Server 7.0

Java ES Release	Web Server Release	General Approach	Reconfiguration Required
Release 4	Sun Java System Web Server 6.1 SP 5 2005Q4	Direct upgrade: Fresh install followed by data migration	Migration of instance configuration to new instance

Web Server Data

The following table shows the type of data that could be impacted by an upgrade of Web Server software.

TABLE 4–2 Web Server Data Usage

Type of Data	Location	Usage
Configuration data	Web Server 6.x (Java ES 4): *WebServer6-base*\`https-`*instanceName* `\config` Web Server 7.0 (Java ES Release 5): *WebServer7Config-base*\`https-`*configName* `\config` Web Server 7.0 (Java ES Release 5): Central Configuration Store. This is not a public interface that is managed by the `admin-server` instance.	Configuration of Web Server instance

Web Server Compatibility Issues

Java ES Release 5 Web Server does not introduce any changes in public interfaces and is backwardly compatible with earlier versions. However, the new administrative interfaces for deploying web applications impact the upgrade and redeployment of web applications including, for example, Java ES components.

In particular, Release 5 Web Server uses different defaults for instance directories and virtual server names, as shown in the following table.

TABLE 4–3 Web Server Instance Directories and Virtual Server Names

Item	Java ES 4 Web Server 6.*x* Default	Java ES 5 Web Server 7.0 Default
Configuration name		*hostName.domainName*
Instance directory path	*WebServer6-base*\`https-`*hostName.domainName*	*WebServer7Config-base*`https-`*hostName.domainName*
Virtual server name	https-*hostName.domainName*	*hostName.domainName*

Web Server Dependencies

Web Server has no dependencies on other Java ES components other than on Java ES shared components, as listed in Table 1–7.

Upgrading Web Server from Java ES Release 4

This section includes information about upgrading Web Server from Release 4 to Release 5. This section covers the following topics:

Introduction to Upgrading Web Server

When upgrading Release 4 Web Server to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach**. The upgrade is performed by doing a fresh install of Release 5 Web Server, migrating Release 4 Web Server instance configuration information to a Release 5 configuration, and then creating Release 5 Web Server instances that correspond to the Release 4 instances.
- **Upgrade Dependencies**. Web Server has dependencies on a number of Java ES shared components, all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Web Server. These dependencies are listed in Table 1–7. Web Server has hard upgrade dependencies only on NSS and NSPR shared components.

  Backward Compatibility. Release 5 Web Server is backwardly compatible with the Release 4 version.

  Upgrade Rollback. Rollback of the Release 5 upgrade is achieved by reverting to the Release 4 installation, which remains intact.

Web Server Upgrade

This section describes how to perform an upgrade of Web Server from Release 4 to Release 5. This section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Web Server you should perform the tasks described below.

Upgrading Web Server Dependencies

All Java ES components on a computer system and in a computing environment should be upgraded to Java ES Release 5. However, all shared components required by Web Server are upgraded automatically by the Java ES installer when you perform an upgrade of Web Server to Release 5.

Backing Up Web Server Data

The Web Server upgrade from Release 4 to Release 5 does not modify the existing configuration data, which is left intact. You do not need to back up current data.

▼ To Verify Current Version Information

◗ **Type the following command:**

WebServer6-base`\https-`*hostName.domainName*`\start -version`

This command returns the Web Server version information.

Release 5 Sun Java System Web Server 7.0

Release 4 Sun Java System Web Server 6.1 SP5

Upgrade Considerations

When upgrading Web Server software to Java ES Release 5 you should take into account the following considerations:

Note – There are some issue with migration of Java ES 4 data to Java ES 5. For more information about Web Server migration issues, see the *Sun Java System Web Server 7.0 Release Notes for Microsoft Windows*.

- **Migration of a default Release 4 instance configuration.** When performing an upgrade, you migrate configuration data for each Release 4 Web Server instance to a central configuration store maintained by the Web Server Administration Server. The migration is achieved using the `wadm migrate-server` command or the Release 5 Administration Console.

  If an instance being migrated is a default Release 4 Web Server (6.*x*) instance, it has the same name (`hostName.domainName`) as the default Release 5 Web Server (7.0) configuration, which is automatically created by the Configure automatically during installation option.

  There are three approaches to performing the migration of a default Release 4 instance configuration to Release 5, each of which results in a different configuration name. The approach you choose can impact the subsequent upgrade of deployed web applications. For example, the upgrade of deployed Java ES components (such as Access Manager and Portal Server) and Sun Java Communications Suite components (such as Communications

Express, Instant Messaging, and Delegated Administrator) generally requires that the person performing such upgrades know the name of the Release 5 configuration to which the Release 4 instance configuration has been migrated.

The three approaches are the following:

- Specify a new configuration name different from the default name. The `wadm migrate-server` command will then create a new configuration. Subsequently upgraded web applications would need to be redeployed to the new configuration.
- Do not specify a new configuration name, but delete the new default Release 5 instance and configuration (`hostName.domainName`) before running the `migrate-server` command. The `migrate-server` command will then create a new configuration with the default name (`hostName.domainName`). The sequence would be as follows:

  1. `wadm delete-instance`
  2. `wadm delete-config`
  3. `wadm migrate-server`
  4. `wadm create-instance`

  Subsequently upgraded web applications would need to be redeployed to the `hostName.domainName` configuration, which is the same as the default Release 4 instance name.
- Do not specify a new configuration name, and do not delete the new default Release 5 configuration. The `migrate-server` command will then create a new configuration with the following name: `hostName.domainName-1`.

- **Migration of configuration data.** When migrating Release 4 instance configurations, the following information is automatically migrated:
  - All the configuration information in the Release 4 Web Server instance directory: `WebServer6-base\https-instanceName\config`. This includes configuration information for all web applications deployed in the Release 4 instance. For example, Java ES components such as Access Manager and Portal Server.
  - `acl` information from `WebServer6-base\httpacl`
  - `auth-db` information from `WebServer6-base\userdb`
  - Scheduler information from `WebServer6-base\https-admserv\config`
  - Certificate information from `WebServer6-base\alias`
  - Search collection information and index files, as specified when you perform the migration.

  The automatic migration does not include the following data:
  - The new configuration will point to the old `docroot` and a log message will be recorded in the migration log.

- Webdav collection information will be migrated. Instead, they will point to the Release 4 file and a log message will be recorded in the migration log.
- 3rd party NSAPI plug-ins.
- Log files.
- Changes to the search collection `docroot`.
- Command line utilities, `startsvr`, `startsvr.bat`, `stopsvr`, `stopsvr.bat`, `restart`, `reconfig`, and `reconfig.bat`.

- **Creation of Release 5 Web Server instances.** After migration, you have to explicitly create a Release 5 Web Server instance corresponding to the migrated Release 4 instance. This operation is not done automatically.

Note – Windows does not allow two services with same name on a single host. For this reason, if you install Java ES 5 Web Server on a host that already has Java ES 4 Web Server installed on it then the default instance name for Java ES 5 Web Server instance will be `https-hostname-1` but the document root will be created under *WebServer7–base*`\http-instancename`.

▼ To Upgrade Web Server

Before You Begin

- You need to log in as administrator to perform the upgrade and the user account performing migration should have permission to access the existing Web Server installation directories.
- Any J2EE components running in an Web Server instance should be shut down before you upgrade that instance.
- All Web Server instances corresponding to the same installed Web Server image are upgraded at the same time. All such instances should be shut down.
- After performing a fresh install of Release 5 Web Server, you have to migrate configuration data for each Release 4 Web Server instance to a central configuration store maintained by the Web Server Administration Server.

  The migration is achieved using the `wadm migrate-server` command or the Release 5 Administration Console. The Release 4 Web Server instance should be shut down before migrating its instance configuration data.

1 Log in as administrator

2 Stop all running instances of Web Server and the Administration Server.

WebServer6-base`\https-`*instanceName*`\stop`

WebServer6-base`\https-admserv\stop`

3. **Install Java ES 5 Web Server.**

   For Java ES 5 Web Server installation instructions, see the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

4. **Start the Web Server Administration Server service.**

   WebServer7Config-base`\admin-server\bin\startserv`

5. **Migrate Release 4 Web Server instance configurations to Release 5 configurations.**

   You can use either the command-line (`wadm migrate-server`) or graphical user interface administration tools. For more information about data migration, see the *Sun Java System Web Server 7.0 Installation and Migration Guide*.

   For more information on `wadm` command and options, see the *Sun Java System Web Server 7.0 CLI Reference Manual*.

6. **Create Release 5 Web Server instances.**

   You must create a new Release 5 instance for each migrated Release 4 instance. If you have migrated to the default configuration, a corresponding default instance already exists.

   - **If you have not migrated to the default configuration:**

     a. **Before creating a new instance, verify the migration log and fix any issues in the migrated configuration.**

     b. **Type the `create-instance` command to create Web Server instances.**

        WebServer7-base`\bin\wadm create-instance --config=`*configName nodehost1* [*nodehost2 ...nodehostN*]

   - **If you have not migrated to the default configuration:**

     Deploy the corresponding configuration without first having to create the instance.

     WebServer7-base`\bin\wadm deploy-config` *configName*

7. **Start each Release 5 instance.**

   WebServer7Config-base`\https-`*configName*`\bin\startserv`

▼ To Verify the Upgrade

◗ **Type the following command:**

WebServer6-base`\https-`*hostName.domainName*`\start -version`

This command returns the Web Server version information.

Release 5	Sun Java System Web Server 7.0
Release 4	Sun Java System Web Server 6.1 SP5

Post-Upgrade Tasks

The main post-upgrade task concerns performing manual migration, if needed, of certain Release 4 data. This is data normally associated with one or more virtual servers configured for Release 4 and specified in the `server.xml` configuration file.

For more information about data migration, see the *Web Server 7.0 Installation and Migration Guide*.

▼ To Roll Back the Upgrade

1. **Log in as administrator.**
2. **Stop all running Web Server instances one by one.**

   WebServer7Config-base`\https-`*configName*`\bin\stopserv`
3. **Remove the Release 5 Web Server installation.**

   This step includes deleting all Release 5 instances and configurations.
4. **Restart the Web Server instances that were stopped when upgrading Web Server.**

   For more information, see "To Upgrade Web Server" on page 51.

◆ ◆ ◆ CHAPTER 5

Message Queue

This chapter describes how to upgrade Message Queue software from previous Java ES versions to Java ES 5: Sun Java System Message Queue 3.7 UR 1.

The chapter provides a general overview of Message Queue upgrade issues and procedures for the different upgrade paths supported by Java ES 5. This chapter covers the following topics:

Note – File locations in this chapter are specified with respect to a fixed directory path referred to as *MessageQueue-base*. The default value of *MessageQueue-base* is `C:\Program Files\Sun\JavaES5\mq`.

Overview of Message Queue Upgrades

The following sections describe general aspects of Message Queue that affect upgrading to Release 5:

About Java ES 5 Message Queue

Java ES 5 Message Queue represents minor code fixes with no new features, enhancements, or compatibility issues.

With Java ES 5 Message Queue, the Platform Edition is deprecated and Message Queue includes all Enterprise Edition features. An upgrade from an earlier Java ES version to Java ES 5 converts any installed Platform Edition to full Message Queue enterprise-level features.

Message Queue Upgrade Roadmap

The following table shows the supported Message Queue upgrade paths to Java ES 5.

TABLE 5–1 Upgrade Paths to Java ES 5 Message Queue 3.7 UR1

Java ES Release	Message Queue Release	General Approach	Reconfiguration Required
Release 4	Sun Java System Message Queue 2005Q4 (3.6 SP3) Enterprise Edition only	Direct upgrade:Performed using coexistence	Need manual steps to migrate data

Message Queue Data

Message Queue, like other Java ES components, makes use of various kinds of data that for any specific upgrade might need to be migrated to an upgraded version. Table 5–2 shows the type of data that could be impacted by an upgrade of Message Queue software.

In the table, *Instance-Name* identifies the name of the Message Queue broker instance with which the data is associated and *MessageQueue-base* is the installation directory for Message Queue.

For Java ES 5, `C:\Program Files\Sun\JavaES5\mq` is the default installation location for Message Queue . For Java ES 4, `C:\Sun\MessageQueue` is the default installation location.

TABLE 5–2 Message Queue Data Usage

Data Category	Location	Usage
Broker instance configuration properties	*MessageQueue-base*`\var\instances\`*Instance-Name*`\config.properties`	Broker and related services configurations

TABLE 5–2 Message Queue Data Usage *(Continued)*

Data Category	Location	Usage
Persistent store for dynamic application data	Release 4: *MessageQueue-base*\var\instances*Instance-Name*\fs350 Release 5: *MessageQueue-base*\var\instances*Instance-Name*\fs370 or accessible through the Java Database Connectivity (JDBC™) API	Stores messages, destinations, durable subscriptions, transactions, and other dynamic data
Administered objects (object store)	Local directory of your choice or an LDAP Directory Server	Objects used to configure client/broker connections
Security: user repository	*MessageQueue-base*\var\instances*Instance-Name*\etc\passwd	Stores user data used for authentication and authorization
Security: access control file (default location)	*MessageQueue-base*\var\instances*Instance-Name*\etc\accesscontrol.properties	Sets the rules that authorize user access to destinations and related capabilities
Security: passfile directory (default location)	*MessageQueue-base*\var\instances*Instance-Name*\etc\	Stores encrypted password information.
Security: broker's keystore file location	*MessageQueue-base*\etc	Stores encrypted certificate information for secure messaging.

Message Queue Dependencies

Message Queue dependencies on other Java ES components can impact the procedure for upgrading and reconfiguring Message Queue software. Changes in Message Queue interfaces or functions, for example, could require upgraded version of components upon which Message Queue depends. The need to upgrade such components depends upon the specific upgrade path.

Message Queue has dependencies on the following Java ES components:

- **Shared components**. Message Queue has dependencies on specific Java ES shared components, as listed in Table 1–7.
- **Directory Server (optional)**. If you want to configure Message Queue to store administered objects and/or user data in an LDAP directory rather than locally, you can use Directory Server for that purpose.

- **Web Container (optional)**. If you need HTTP messaging between client and broker, then Message Queue requires web container support from Java ES Web Server or from Java ES Application Server.
- **Databases (optional)**. You can configure Java DB or a third-party database as a data store for the Message Queue persistence layer.

Upgrading Message Queue from Java ES Release 4

This section includes information about upgrading Message Queue from Java ES 4 to Java ES 5. The section covers the following topics:

Introduction to Upgrading Message Queue

When upgrading Java ES 4 Message Queue to Java ES 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach**. Java ES 5 does not provide any in place upgrade support.
- **Upgrade Dependencies**. Message Queue has dependencies on a number of Java ES shared components, as listed in Table 1–7. Java ES 5 Message Queue is compatible with the Java ES 4 versions of all the components except NSS and NSPR. For this reason, you must upgrade these components while upgrading to Java ES 5.

  In addition, Java ES 5 Message Queue is optionally dependent on Directory Server and Web Server (or Application Server), as described in "Message Queue Dependencies" on page 57. However, upgrade of these components is optional.
- **Backward Compatibility**. Java ES 5 Message Queue is fully compatible with Java ES 4 with respect to protocols, broker compatibility, administered objects, administration tools, and client applications.
- **Upgrade Rollback**. Java ES 5 does not provide any utility to roll back the Message Queue upgrade to Java ES 4. You have to remove the upgraded components and manually restore the previous version and configuration data.

Message Queue Upgrade

This section describes how to perform a Message Queue upgrade from Java ES 4 to Java ES 5. This section contains the following sections:

Pre-Upgrade Tasks

Before you upgrade Message Queue, perform the procedures described in the following sections.

Upgrading Message Queue Dependencies

All Java ES 4 components on a computer system and in a computing environment should be upgraded to Java ES 5. However, because Message Queue does not require upgrading the Java ES 4 components upon on which it depends, this task is optional.

However, if you choose to upgrade all Message Queue dependencies, they should be upgraded in the following order, all before you upgrade Message Queue. You can skip any items that might already have been upgraded.

- **Shared Components**. Release 5 installer automatically installs all required shared components in the parallel path..
- **Directory Server (optional)**. Instructions for upgrading Directory Server to Java ES 5 are provided in Chapter 2.
- **Web Container Software (optional)**. Instructions for upgrading Web Server or Application Server are provided in Chapter 4 and Chapter 6 respectively.

▼ To Verify Current Version Information

◗ **Start the Message Queue broker with `-version` option.**

```
imqbrokerd -version
```

The outputs that indicate the Message Queue version are:

Release 5	Sun Java System Message Queue 3.7 UR1
Release 4	Sun Java System Message Queue 3 2005Q4 Version: 3.6 SP3

▼ To Upgrade Message Queue

Before You Begin Back up application data in a production environment before performing an upgrade. For Message Queue data locations, refer Table 5–2.

1 **Stop any running Java ES 4 Message Queue instances.**

If Message Queue is being used in an Application Server environment, shut down Application Server, as well.

2 **Stop any running brokers.**

3 **Install the Java ES 5 Message Queue.**

For Java ES 5 Message Queue installation instructions, see the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

4 **(Optional) Migrate the Release 4 Message Queue data, as listed in Table 5–2, to the Release 5 Message Queue installation location.**

You need to perform this step only if you wish to preserve any Message Queue broker data from Release 4.

5 **(Optional) Uninstall Release 4 Message Queue.**

Note – Uninstall of Release 4 Message Queue is optional because Java ES versions 4 and 5 can coexist on a system.

▼ To Verify Message Queue Upgrade

◗ **Start the Message Queue broker with `-version` option.**

```
imqbrokerd -version
```

The outputs that indicate the Message Queue version are:

Release 5 Sun Java System Message Queue 3.7 UR1

Release 4 Sun Java System Message Queue 3 2005Q4 Version: 3.6 SP3

Post-Upgrade Tasks

If you have upgraded the web container and are using the Message Queue HTTP tunneling servlet, you may need to redeploy the servlet in the new web container. Otherwise, you do not need to redeploy the servlet after upgrading Message Queue. For more information about HTTP support, see the *Sun Java System Message Queue 3.7 UR1 Administration Guide*.

▼ To Roll Back the Upgrade

1 **Stop any running Release 5 Message Queue instances.**

If Message Queue is being used in an Application Server environment, shut down Application Server as well.

2 **Stop any running brokers.**

3 **Migrate back the Release 4 Message Queue data, as listed in Table 5–2, to the Release 4 Message Queue installation location.**

4 Start the Release 4 Message Queue.

◆ ◆ ◆ CHAPTER 6

Application Server

This chapter describes how to upgrade Application Server from previous Java ES versions to Java ES 5: Sun Java System Application Server Enterprise Edition 8.2.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES 5. The chapter covers the following topics:

Note – File locations in this chapter are specified with respect to a directory path referred to as *ApplicationServer-base*. At least part of this path might have been specified as an installation directory when Application Server was initially installed. If not, the Java ES installer assigned a default value.

The default value of *ApplicationServer-base* is `C:\Program Files\Sun\JavaES5\appserver`

Overview of Application Server Upgrades

The following sections describe general aspects of Application Server that affect upgrading to Java ES 5:

About Java ES 5Application Server

Java ES 5 Application Server represents selected bug fixes to the Java ES 4 version. Functionally Java ES 5 Application Server is the same as Java ES 4.

Application Server Upgrade Roadmap

The following table shows the supported Application Server upgrade paths to Java ES 5.

TABLE 6–1 Upgrade Paths to Java ES 5: Sun Java System Application Server Enterprise Edition 8.2

Java ES Release	Application Server Release	General Approach	Reconfiguration Required
Release 4	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	Direct upgrade:Performed using coexistence	None

Application Server Data

The following table shows the type of data that could be affected by an upgrade of Application Server software.

TABLE 6–2 Application Server Data Usage

Type of Data	Location	Usage
Environment variables	*ApplicationServer-base*`\config\asenv.bat`	Global variables
Configuration data	*ApplicationServer-base*`\domains\domainName\config`	Configuration of Application Server instances
Deployment data	*ApplicationServer-base*`\domains\domainName\applications`	Configuration of J2EE container for specific J2EE components and applications

Application Server Compatibility Issues

Java ES 5 Application Server does not introduce any interface changes with respect to Java ES 4.

Application Server Dependencies

Application Server dependencies on other Java ES components can affect the procedure for upgrading and reconfiguring Application Server software. Changes in Application Server interfaces or functions, for example, could require upgraded versions of components upon which Application Server depends. The need to upgrade such components depends upon the specific upgrade path.

Application Server has dependencies on the following Java ES components:

- **Shared components**. Application Server has dependencies on specific Java ES shared components, as listed in Table 1–7.
- **Message Queue.** Application Server depends on Message Queue to provide J2EE Java Message Service-compliant asynchronous messaging support.
- **HADB.** Application Server depends on High Availability Database (HADB) for high availability storage of HTTP session and stateful session bean data. HADB is designed to support up to 99.999% service and data availability with load balancing, failover, and state recovery.
- **NSS.** Application Server depends on Network Security Service (NSS) for managing security.
- **Web Container (optional)**. Application Server depends upon web container services for its optional load balancing plug-in. This support can be provided by Java ES Web Server.
- **High Availability Session Store (optional).** Application Server depends upon High Availability Session Store to maintain session state information needed to support failover between instances.

Upgrading Application Server from Java ES 4

This section includes information about upgrading Application Server from Java ES 4 to Java ES 5. This section covers the following topics:

- "Introduction to Upgrading Application Server" on page 65
- "Application Server Upgrade" on page 66

Introduction to Upgrading Application Server

When upgrading Java ES 4 Application Server to Java ES 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach**. Java ES 5 does not provide any in place upgrade support.
- **Upgrade Dependencies**. Application Server has dependencies on a number of Java ES shared components, as listed in Table 1–7. Java ES 5 Application Server is compatible with the Release 4 version of all these components.

- **Backward Compatibility**. Java ES 5 Application Server is backwardly compatible with the Java ES 4 provided the shared component dependency is not broken.
- **Upgrade Rollback**. Rollback of the Java ES 5 upgrade to Java ES 4 is achieved by uninstallling the Java ES 5 and reinstalling Java ES 4.

Application Server Upgrade

This section describes how to perform an upgrade of Application Server from Java ES version 4 to Java ES version 5. This section covers the following topics:

Pre-Upgrade Tasks

Upgrading Application Server Dependencies

All Java ES components on a computer system and in a computing environment should be upgraded to Java ES version 5.

▼ To Verify Current Version Information

◗ **Type the following command.**

ApplicationServer-base`\bin\asadmin.bat version`

The outputs that indicate the Application Server version are:

Release 5 Sun Java Enterprise System Application Server Enterprise Edition 8.2

Release 4 Sun Java Enterprise System Application Server Enterprise Edition 8.1_02

▼ To Stop Application Server Component

1 **Stop all running node agents.**

ApplicationServer-base`\bin\asadmin.bat stop-node-agent`

2 **Stop all domains.**

ApplicationServer-base\bin\asadmin.bat stop-domain

3 **Stop the pointbase database server if one is installed and running.**

ApplicationServer-base\pointbase\tools\stopserver.sh

▼ To Back Up Application Server Data

1 **Back up Application Server domains.**

Create a directory outside the Application Server installation directories and back up the content of the `domains` directory in this new directory. The default location for the Java ES 4 domains directory is *ApplicationServer-base*\domains.

2 **Back up Application Server node agents.**

Create a directory outside the Application Server installation directories and back up the content of the nodeagents directory in this new directory. The default location for the Java ES 4 `nodeagents` directory is *ApplicationServer-base*\nodeagents.

3 **Back up pointbase binaries.**

Create a directory outside the Application Server installation directories and back up the content of the `pointbase` directory in this new directory. The default location for the Java ES 4 `pointbase` directory is *ApplicationServer-base*\pointbase.

4 **Back up pointbase data files.**

Create a directory outside the Application Server installation directories and back up the pointbase data files in this new directory. The default location for the Java ES 4 pointbase data files is: *ApplicationServer-base*\pointbase.

5 **Back up the Application Server configuration file.**

The default location for Application Server configuration file is *ApplicationServer-base*\config\asenv.conf.

6 **Back up the sample configuration file.**

The default location for sample configuration file is *ApplicationServer-base*\samples\common.properties.

▼ To Upgrade Application Server

Before You Begin

- Obtain Application Server administrator user ID and password for your currently installed version.

- Shut down all J2EE components running in an Application Server instance before upgrading the instance. However, if load balancing provides for high availability or scalability, this requirement can be relaxed.
- Shut down all instances of Application Server running on a single computer (all corresponding to the same installed Application Server image.
- Back up all domain, node agent, and configuration data.

In multiple node deployments, perform the upgrade procedure on each node or computer that hosts Application Server instances.

1. **Install Java ES 5 Application Server with the Configure Automatically During Installation option.**

   The new installed domain is started automatically.

2. **Stop the domain.**

   You can also delete the domain created under new installation directory.

▼ To Verify the Upgrade

◗ **Type the following command.**

ApplicationServer-base`\bin\asadmin.bat version`

The outputs that indicate the Application Server version are:

Release 5	Sun Java Enterprise System Application Server Enterprise Edition 8.2
Release 4	Sun Java Enterprise System Application Server Enterprise Edition 8.1_02

Post-Upgrade Tasks

To verify a successful upgrade, you need to perform the following post-upgrade tasks:

- "To Restore Application Server Data" on page 68
- "To Validate Restored Domains" on page 69
- "To Validate Restored Node Agents" on page 69
- "To Validate Pointbase Data Files" on page 69

▼ To Restore Application Server Data

1. **Restore Application Server domains.**
2. **Restore Application Server node agents.**
3. **Restore pointbase binaries.**

4. **Restore pointbase data files.**

5. **Restore the configuration settings from the Java ES 4 Application Server configuration file.**
   Ensure that the new entries from the Java ES 5 configuration file are retained in this file.

6. **Restore the configuration settings from the Java ES 4 sample configuration file.**
   Ensure that the new entries from the Java ES 5 sample configuration file are retained in this file.

▼ To Validate Restored Domains

1. **Start the restored Java ES 4 Application Server domains.**
   ApplicationServer-base`\bin\asadmin.bat start-domain --user` *userName* `--domaindir` *JES4-Domain-location domainName*

2. **Check for the deployed applications.**

3. **Check for the configuration settings.**

▼ To Validate Restored Node Agents

- **Start the restored Java ES 4 node agents.**
  ApplicationServer-base`\bin\asadmin.bat start-node-agent --user` *userName* `--agentdir` *JES4-node-agent-location nodeAgentName*

▼ To Validate Pointbase Data Files

1. **Start the restored Java ES 4 Application Server domains.**
   ApplicationServer-base`\pointbase\tools\startpointbase.sh`

2. **Verify that the data files are accessible to the applications.**

▼ To Roll Back the Upgrade

1. **Uninstall Java ES 5.**

2. **Start domain and node agent from Java ES4 Application Server install location.**

◆ ◆ ◆ CHAPTER 7

Service Registry

This chapter describes how to upgrade Service Registry to Java ES 5 (Release 5): Service Registry 3.1.

This chapter covers the following topics

Note – File locations in this chapter are specified with respect to directory paths referred to as *ServiceRegistryR4-base* and *RegistryDomainR4-base* (Java ES Release 4 Service Registry), and *ServiceRegistryR5-base* and `RegistryDomainR5-base` (Java ES Release 5 Service Registry). At least part of these paths might have been specified as installation directories when Service Registry was installed. If not, the Java ES installer assigned a default value. The default values of these directory paths are:

ServiceRegistryR4-base	`C:\Sun\ServiceRegistry`
DomainRegistryR4-base	`C:\Sun\ServiceRegistry`
ServiceRegistryR5-base	`C:\Program Files\Sun\JavaES5\srvc-registry`
DomainRegistryR5-base	`C:\Program Files\Sun\JavaES5\srvc-registry`

Overview of Service Registry Upgrade

The following sections describe general aspects of Service Registry that affect upgrading to Java ES 5:

- "Service Registry Compatibility Issues" on page 73
- "Service Registry Dependencies" on page 73

About Java ES 5 Service Registry

Release 5 Service Registry represents a minor release with respect to Java ES 2005Q4 (Release 4) Service Registry. It includes some improved functionality, updated interfaces, and selected bug fixes.

Service Registry Upgrade Roadmap

The following table shows the supported Service Registry upgrade paths to Java ES Release 5.

TABLE 7–1 Upgrade Paths to Java ES 5: Sun Java System Service Registry 3.1

Java ES Release	Service Registry Release	General Approach	Reconfiguration
Release 4	Sun Java System Service Registry 3.0 2005Q4	Direct upgrade: Replace Release 4 with a fresh install and transfer registry data to Release 5.	None

Service Registry Data

The following table shows the type of data that could be impacted by an upgrade of Service Registry software.

TABLE 7–2 Service Registry Data Usage

Type of Data	Location	Usage
Installation Parameters	*ServiceRegistryR4-base*\install\install.properties	Configuration of Service Registry
Trusted certificates	*ServiceRegistryR4-base*\install\cacerts	Certificates trusted by Service Registry that are not part of the Application Server installation
Configuration data	*RegistryDomainR4-base*\domains\registry\applications\j2ee-modules\soar\WEB-INF\classes*.properties	Configuration of Service Registry instance

TABLE 7–2 Service Registry Data Usage *(Continued)*

Type of Data	Location	Usage
Registry/repository data	*RegistryDomainR4-base**3.0*\`data`	Database and user certificates
Web interface configuration	*RegistryDomainR4-base*\`3.0`\`jaxr-ebxml`	Configuration of web interface

Service Registry Compatibility Issues

Release 5 Service Registry is backwardly compatible with Release 4 Service Registry.

Service Registry Dependencies

Service Registry dependencies on other Java ES components can affect the procedure for upgrading and reconfiguring Service Registry software. Changes in Service Registry interfaces or functions, for example, could require upgraded version of components upon which Service Registry depends. The need to upgrade such components depends upon the specific upgrade path.

Service Registry has dependencies on the following Java ES components:

- **Shared Components.** Service Registry has dependencies on specific Java ES shared components, as listed in Table 1–7.
- **Application Server.** Service Registry depends on Application Server to provide a container for the Service Registry application and, in Release 5, to manage connections to the networked registry and repository database.
- **Java DB.** Service Registry depends upon Java DB as the default database for storing services and the meta data describing them.

Upgrading Service Registry from Java ES Release 4

This section includes information about upgrading Service Registry from Release 4 to Release 5. The section covers the following topics:

- "Introduction to Upgrading Service Registry" on page 73
- "Release 4 Service Registry Upgrade" on page 74

Introduction to Upgrading Service Registry

When upgrading Release 4 Service Registry to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach**. The upgrade is achieved by performing a fresh install of Release 5 Service Registry, migrating the Release 4 data and configuration to Release 5, and then removing Release 4 to conserve disk space.
- **Upgrade Dependencies**. Service Registry has dependencies on a number of Java ES shared components, as listed in Table 1–7. All of these shared components are automatically upgraded to Release 5 by the Java ES installer when you perform an Service Registry upgrade.

  In addition, Service Registry has a hard upgrade dependency on Application Server and Java DB.
- **Backward Compatibility**. Release 5 Service Registry is backwardly compatible with Release 4.
- **Upgrade Rollback**. A rollback of the Release 5 upgrade is achieved by reverting to Release 4 after restoring the saved database and configuration data.

Release 4 Service Registry Upgrade

This section describes how to perform a Java ES 4 Service Registry upgrade to Java ES 5. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Service Registry, perform the following procedures:

Backing Up Service Registry Data

The Service Registry upgrade from Release 4 to Release 5 does not modify configuration data or the registry and repository database. You do not have to back up current data.

Upgrading Service Registry Dependencies

All Java ES components on a computer system and in a computing environment should be upgraded to Java ES Release 5. Service Registry has hard upgrade dependencies on a number of shared components and on Application Server.

When upgrading Service Registry dependencies, you should do so in the order below, before you upgrade Service Registry. Upgrade of shared components is normally achieved automatically by the Java ES installer.

1. **Shared Components.**Release 5 installer automatically installs all required shared components in the parallel path.
2. **Application Server.** Instructions for upgrading Application Server to Release 5 are provided in Chapter 6

▼ To Upgrade the Service Registry

Before You Begin

- Obtain the user IDs, passwords, domain name, and port number for your Release 4 Service Registry.

1 Log in as administrator.

2 Stop the Release 4 Service Registry (Application Server) domain.

```
cd ServiceRegistryR4-base\install

ant -f build-install.xml appserver.domain.stop
```

3 Install Release 5 Service Registry.

Use the Configure Manually After Installation option to install Java ES 5 Service Registry. For Service Registry installation instructions, see the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

4 Update the Release 5 `install.properties` file.

a. Set the `jes.data.home property` property in the `install.properties` file to point to the Release 4 install location.

b. Set the `soar.data.home` property in the `install.properties` file to point to *ServiceRegistryR4-base*.

5 Perform post-install upgrade and configuration of the Release 5 Service Registry.

```
cd ServiceRegistryR5-base\install

ant -f build-install.xml upgrade
```

The upgrade utility creates a new Application Server domain for Service Registry, starts the Service Registry domain and deploys the Service Registry. Each Service Registry is associated with its own Application Server domain.

6 Stop the Release 5 Service Registry (Application Server) domain.

```
cd ServiceRegistryR5-base\install

ant -f build-install.xml appserver.domain.stop
```

7. **Update the Release 5 configuration data files with changes made to the Release 4 configuration data files.**

8. **Start the Release 5 Service Registry (Application Server) domain.**

   ```
   cd ServiceRegistryR5-base\install
   ant -f build-install.xml appserver.domain.start
   ```

▼ To Roll Back the Upgrade

1. **Log in as administrator.**

2. **Stop and delete the Release 5 Service Registry (Application Server) domain.**

   ```
   cd ServiceRegistryR5-base\install
   ant -f build-install.xml appserver.domain.stop appserver.domain.delete
   ```

3. **Run the Release 5 uninstaller to uninstall Release 5 Service Registry.**

4. **Start the Release 4 Service Registry (Application Server) domain.**

   ```
   cd ServiceRegistryR4-base\install
   ant -f build-install.xml appserver.domain.start
   ```

◆ ◆ ◆ CHAPTER 8

Web Proxy Server

This chapter describes how to upgrade Web Proxy Server to Java ES 5 (Release 5): Sun Java System Web Proxy Server 4.0.4.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5.

Note – File locations in this chapter are specified with respect to a directory path referred to as *WebProxyServer-base*. At least part of this path might have been specified as an installation directory when Web Proxy Server was initially installed. If not, the Java ES installer assigned a default value.

The default value of *WebProxyServer-base* is `C:\Program Files\Sun\JavaES5\webproxyserver`.

Overview of Web Proxy Server Upgrade

The following sections describe general aspects of Web Proxy Server that affect upgrading to Java ES 5:

About Java ES 5 Web Proxy Server

Java ES 5 Web Proxy Server represents a minor bug-fix release with respect to Release 4. However, Release 5 Web Proxy Server includes better performance, more scalable architecture, better standards compliance, and a new administration interface as compared to Sun ONE Web Proxy Server 3.6 before its inclusion in the Java Enterprise System software.

Web Proxy Server Upgrade Roadmap

The following table shows the supported Web Proxy Server upgrade paths to Java ES Release 5.

TABLE 8–1 Upgrade Paths to Java ES Release 5: Sun Java System Web Proxy Server 4.0.4

Java ES Release	Web Proxy Server Relase	General Approach	Reconfiguration Required
Release 4	Sun Java System Web Proxy Server 4.0.1 2005Q4	Direct upgrade: Performed using patches	None

Web Proxy Server Data

The following table shows the type of data that could be impacted by an upgrade of Web Proxy Server software.

TABLE 8–2 Web Proxy Server Data Usage

Type of Data	Location	Usage
Configuration data	*WebProxyServer-base*\`proxy-serverid\config` This directory contains files, such as `server,xml`, `magnus.conf`, and `obj.conf`	Stores configuration information for the server, cache, filters, routing, and other functional aspects of Web Proxy Server

Web Proxy Server Compatibility Issues

Release 5 Web Proxy Server does not introduce any new public interfaces and is backwardly compatible with Release 4 Web Proxy Server. Release 5 Web Proxy Server is also compatible with release 3.6, except that plug-ins developed using the NSAPI interface supported by release 3.6 must be recompiled with the NSAPI interface supported by Release 5.

Web Proxy Server Dependencies

Web Proxy Server dependencies on other Java ES components can affect the procedure for upgrading and reconfiguring Web Proxy Server software.

Web Proxy Server has dependencies on specific Java ES shared components, as listed in Table 1–7, but has no dependencies on other Java ES product components. Web Proxy Server can be used with Directory Server, Web Server, and Application Server, but beyond this functional relationship to these components it has no dependencies on them.

Upgrading Web Proxy Server from Java ES Release 4

This section includes information about upgrading Web Proxy Server from Java ES release 4 to Java ES release 5. The section covers the following topics:

Introduction

When upgrading Java ES Release 4 Web Proxy Server to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach**. The upgrade is performed using patches. No additional reconfiguration is required.
- **Upgrade Dependencies**. Web Proxy Server has dependencies on a number of Java ES shared components, as listed in Table 1–7. However Web Proxy Server has hard upgrade dependencies only on NSS and NSPR shared components.
- **Backward Compatibility**. Release 5 Web Proxy Server is backwardly compatible with Release 4.
- **Upgrade Rollback**. Release 5 Web Proxy Server upgrade rollback is performed by removing the upgrade patches.

Release 4 Web Proxy Server Upgrade

This section describes how to perform a Java ES 4 Web Proxy Server upgrade to Java ES 5. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Web Proxy Server, perform the procedures described in the following sections.

▼ To Verify Current Version Information

◗ **Type the following command.**

WebProxyServer-base`\proxy-admserv\start -version`

The outputs that indicate the Web Proxy Server version are:

Release 5 Sun Java System Web Proxy Server 4.0.4

Release 4 Web Proxy Server 4.0.1

Upgrading Web Proxy Server Dependencies

All Java ES components on a computer system and in a computing environment should be upgraded to Java ES Release 5.

Backing Up Web Proxy Server Data

The Web Proxy Server upgrade to Release 5 does not modify Release 4 configuration data. You do not need to back up current data.

▼ To Upgrade Web Proxy Server

Before You Begin All Web Proxy Server instances corresponding to the same installed Web Proxy Server image are upgraded at the same time. All such instances should be shut down when patches are being applied to the installed image.

1 **Log in as administrator.**

2 **Stop all running instances of Web Proxy Server and the Administration Server.**

WebProxyServer-base`\proxy-`*instancename*`\stopsvr`

WebProxyServer-base`\proxy-admserv\stopsvr`

3 **If you have not already done so, upgrade the NSS and NSPR shared components to Release 5 and any other shared components you wish to upgrade.**

4 **Install Release 5 Web Proxy Server upgrade patches.**

Patches can be downloaded from `http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage`. The patch ID for Web Proxy Server core is 123533-01.

Note – Patch revision numbers are the minimum required for upgrade to Java ES Release 5. If newer revisions become available, use the newer patches instead of the old patches.

---

a. **Download** *Patch-ID*`.zip`**.**

b. **Unzip the zip file.**

c. **Execute** *Patch-ID*`.exe`**.**

5 **Confirm that the patch upgrades were successful.**

You can confirm patch upgrade by checking the patch installation log.

6 **Restart the Web Proxy Server instance.**

WebProxyServer-base`\proxy-`*instancename*`\startsvr`

▼ To Verify Web Proxy Server Upgrade

◗ **Type the following command.**

WebProxyServer-base`\proxy-admserv\start -version`

The outputs that indicate the Web Proxy Server version are:

Release 5 Sun Java System Web Proxy Server 4.0.4

Release 4 Web Proxy Server 4.0.1

▼ To Remove Web Proxy Server Patches

1 **Log in as administrator.**

2 **Stop all running instances of Web Proxy Server and the Administration Sever.**

WebProxyServer-base`\proxy-`*instancename*`\stopsvr`

WebProxyServer-base`\proxy-admserv\stopsvr`

3 **Remove the Web Proxy Server patches by typing the following command:**

`Uninstall_`*Patch_ID*`.bat`.

4 **Restart the Web Proxy Server instances.**

◆ ◆ ◆ CHAPTER 9

Access Manager

This chapter describes how to upgrade Access Manager software from previous Java ES releases to Java ES 5: Sun Java System Access Manager 7.1.

The chapter provides a general overview of Access Manager upgrade issues and procedures for the different upgrade paths supported by Java ES 5 (Release 5).

Note – File locations in this chapter are specified with respect to a directory path referred to as *AccessManager-base*. At least part of this path might have been specified as an installation directory when Access Manager was initially installed. If not, the Java ES installer assigned a default value.

The default value of *AccessManager-base* is `C:\Program Files\Sun\JavaES5\identity`.

Overview of Access Manager Upgrades

The following sections describe general aspects of Access Manager that affect upgrading to Java ES release 5:

About Java ES 5 Access Manager

Java ES Release 5 Access Manager 7.1 is a minor release that contains several bugs fixes and RFE's to the Java ES 4 Access Manager 7.0 release. Access Manager 7.1 includes the Java ES Monitoring Framework, which creates are new shared component dependencies. No major changes occurred in terms of functionality. In order to maintain backward compatibility with other components in the Java ES Stack, Access Manager 7.1 can be run in the legacy mode.

Access Manager Upgrade Roadmap

The following table shows the supported Access Manager upgrade paths to Java ES Release 5.

TABLE 9–1 Upgrade Paths to Java ES Release 5: Sun Java System Access Manager 7.1

Java ES Release	Access Manager Release	General Approach	Reconfiguration Required
Release 4	Sun Java System Access Manager 7.0 2005Q4	Direct upgrade: Performed by doing a full installation and reconfiguration of Release 5	Configuration data Web container Customized JavaServer Pages™ (JSP™) for Access Manager console and authentication UI Directory schema

Note – Before starting the upgrade process for Access Manager, the upgrade for Web container and Directory Server should have been completed.

Access Manager Data

Access Manager, like other Java ES components, makes use of various kinds of data that for any specific upgrade might need to be migrated to an upgraded version. The following table shows the type of data that could be impacted by an upgrade of Access Manager software.

TABLE 9–2 Access Manager Data Usage

Type of Data	Location	Usage
Configuration data	*AccessManager-base*\config\AMConfig.properties *AccessManager-base*\config\serverconfig.xml Java Archive (JAR) files for authentication and customized modules *AccessManager-base*\lib	Configuration of Access Manager and its integration with a back-end data store
Web container configuration	Web Server: server.policy and server.xml files in *WebServer-base*\https-*hostname*\config Application Server: server.policy and domain.xml files in *ApplicationServer-base*\domain\domain1\config	Configuration of Access Manager web container instance
Customization data (Web container customized JSP files)	Admin Console: *AccessManager-base*\web-src\applications Authentication UI: *AccessManager-base*\web-src\services	Configuration of Access Manager administration interfaces
Directory schema Services configuration User data	Directory Server	Access Manager provides authentication and authorization services for end users, based on services configuration, user, and policy data that is stored in a directory
Dynamic application data	None	Access Manager does not persistently store application data such as session state

Access Manager Compatibility Issues

Release 5 Access Manager is backwardly compatible with Release 4 Access Manager. Note that Release 4 Access Manager was a major release that, except when configured to run in Legacy

mode, broke compatibility with earlier releases. Likewise, Release 5 Access Manager, unless configured to run in Legacy mode, is not backwardly compatible with the Java ES 4 Access Manager running in Legacy mode.

Legacy mode is necessary to support other Java ES components, as well as older versions of Access Manager policy agents that which cannot interoperate with Access Manager in Realm mode. This incompatibility is an important upgrade consideration, and means in most Java ES deployments that Access Manager should be upgraded to Release 5 Legacy mode even when configured to run in Legacy mode.

Access Manager Dependencies

Access Manager dependencies on other Java ES components can affect the procedure for upgrading and reconfiguring Access Manager software. Changes in Access Manager interfaces or functions, for example, could require upgraded version of components upon which Access Manager depends. The need to upgrade such components depends upon the specific upgrade path.

Access Manager has dependencies on the following Java ES components:

- **Shared components**. Access Manager has dependencies on specific Java ES shared components, as listed inTable 1–7. Access Manager upgrades might depend upon upgraded versions of these shared components.
- **Web Container**. Access Manager depends upon web container services, which can be provided either by Java ES Web Server and Java ES Application Server. Access Manager upgrades must therefore be reconfigured for a web container instance. In addition, any customized JSPs for the Access Manager console or for the authentication UI need to be migrated to the upgraded Access Manager environment.
- **Directory Server**. Access Manager stores configuration data and also accesses user data stored in Directory Server. As a result, Access Manager upgrades might require extensions of directory schema.

Web Container Upgrade Scenarios

Access Manager can be deployed in a web container provided by either Web Server or Application Server. As a result, the upgrade of Access Manager to Release 5 can be complicated by the possibility of also having upgraded to Release 5 the web container in which it is deployed. In this regard, there are a number of web container upgrade scenarios possible, enumerated in the following table:

TABLE 9–3 Web Container Upgrade Scenarios for Access Manager Upgrade

Scenario	Web Container in which Access Manager is Originally Deployed	Web Container in which Access Manager is Deployed After Upgrade	Applicable Access Manager Upgrade Paths: Upgrades From
Scenario 1	Web Server 6.*x*	Web Server 7.0	Release 4
Scenario 2	Application Server 8.1	Application Server 8.2	Release 4

You must be careful when upgrading Access Manager (for example when using the `amconfig.bat`) to provide values appropriate to the upgrade scenario in Table 9–3that applies, especially when there is a major version upgrade of the web container.

Upgrading Java ES Release 5 Access Manager from Java ES Release 4

This section includes information about upgrading Access Manager from Java ES 4 to Java ES 5. The section covers the following topics:

Introduction to Upgrading Access Manager

When upgrading Java ES Release 4 Access Manager to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach**. The upgrade is performed by installing Release 5. Reconfiguration of Access Manager is subsequently performed using the `amconfig.bat` file, and directory schema is migrated using the `amupgrade.bat` file.
- **Upgrade Dependencies**. Access Manager has dependencies on a number of Java ES shared components and other products, as listed in Table 1–7.

  In addition, Release 5 Access Manager is dependent upon Directory Server, Web Server, or Application Server, as described in "Access Manager Dependencies" on page 86. Because these are soft upgrade dependencies, upgrade of these components is optional with respect to upgrade of Access Manager to Release 5.
- **Backward Compatibility**. Release 5 Access Manager is compatible with Release 4. Access Manager 7.1 will continue to work with Release 4 Access Manager configuration for existing features. New features that access new configuration data required upgrading of Access Manager data. These features include Java ES monitoring and feature improvements in authentication, policy, and delegation.

- **Upgrade Rollback**. No utility exists for rolling back the Access Manager upgrade. Do not to uninstall Release 4 Access Manager. In order to restore Release 4 Access Manager, you should manually backup the Directory Server data. The best approach to rollback is to create a parallel installation using backed-up configuration files, and testing this parallel installation before performing the upgrade. This process enable you to revert to the parallel installation if necessary.

Access Manager Upgrade

This section describes how to perform a Access Manager upgrade from Java ES 4 to Java ES 5. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Access Manager, perform the procedures described in the following sections:

Upgrading Access Manager Dependencies

All Java ES components on a computer system and in a computing environment should be upgraded to Java ES Release 5. Access Manager has hard upgrade dependencies on only a couple of shared components.

If you choose to upgrade Access Manager product component dependencies, you should do so in the order below,skipping any components that might already have been upgraded, before you upgrade Access Manager. Upgrade of shared components is normally achieved automatically by the Java ES installer.

1. **Shared Components.** All shared components required by Access Manager are upgraded automatically by the Java ES installer when you perform an upgrade of Access Manager to Release 5.
2. **Directory Server (optional).** Instructions for upgrading Directory Server to Release 5 are provided in Chapter 2.

   Make sure that the Release 5 Directory Server uses the same port as of Release 4 Directory Server.

3. **Web Container Software (optional).** Instructions for upgrading Web Server or Application Server are provided in Chapter 4 and Chapter 6 respectively.

   If web container software is not upgraded before Access Manager, the upgrade procedure will configure and redeploy Access Manager to the existing web container.

   Make sure that the Release 5 web container uses the same port as of Release 4 web container.

Backing Up Directory Server Data

The Access Manager upgrade process uses scripts that modify Directory Server schema. Therefore, before you upgrade Access Manager, back up your Directory Server data using the Directory Server Console or a command-line utility such as `db2bak`.

For more information about backing up Directory Server, see the *Sun Java System Directory Server Administration Guide*.

Back Up Release 4 Access Manager Configuration Information

Because the reconfiguration of Release 5 Access Manager software requires the reconfiguration of the Release 4 version, you should back up configuration files to a known location. The following Web container configuration files should be backed up:

- For Web Server: `server.policy` and `server.xml` files located in *WebServer-base*`\https-`*hostname*`\config`
- For Application Server: `server.policy` and `domain.xml` files located in *ApplicationServer-base*`\domain\domain1\config`

▼ To Verify Current Version Information

◗ **Type the following command.**

AccessManager-base`\bin\amadmin --version`

The outputs that indicate the Access Manager version are:

Release 5 Access Manager 7.1

Release 4 Access Manager 7 2005Q4

▼ To Upgrade Access Manager

Before You Begin

- Obtain Access Manager administrator user ID and password, LDAP user ID and password, and Directory Manager name and password for the Directory Server instance that Access Manager is using.
- Before uninstalling all other Java ES components, back up the required data. For more information about backing up other components see the upgrade guides of the respective components.

1. **Log in as administrator to the machine where Java ES 4 Access Manager is installed.**

2. **Manually backup the Access Manager DIT (Directory Server data).**

3. **Stop the following Java ES 4 services:**

   - Web Server
   - Directory Server
   - Directory Proxy Server
   - Application Server
   - Instant Messaging
   - Calender Server
   - Messaging Server

4. **Install the Java ES 5 Access Manager.**

   For Java ES 5 Access Manager installation instructions, see the *Sun Java Enterprise System 5 Installation Guide for Microsoft Windows*.

   Note – Restart the machine after installing Java ES 5 Access Manager.

5. **Re-customize JavaServer Pages for Access Manager.**

   a. **Re-apply the Release 4 customization to JavaServer Pages for the Access Manager Console and authentication user interface (UI) present in the Release 4 installation location.**

   b. **Copy the customized JSP files to the correct directories.**

      - Console: *AccessManager-base*`\web-src\applications\console`
      - Authentication UI:

        AccessManager-base`\web-src\services\config\auth\default` or *AccessManager-base*`\web-src\services\config\auth\default_`*locale* (where *locale* is a locale indicator like `ja`)

      For more information, see the *Sun Java System Access Manager Developer's Guide*.

6. **Configure Access Manager.**

   Configure Access Manager for your specific web container by running the `amconfig.bat` file. The `amconfig.bat` file and the associated `AMConfigurator.properties` input file resides in the *AccessManager-base*`\setup` directory.

   For information about the `amconfig.bat` file and the `AMConfigurator.properties` file, see the *Sun Java System Access Manager Administration Guide*.

Perform the steps to reconfigure and redeploy Access Manager to the web container as described in "To Reconfigure and Redeploy Access Manager" on page 91.

7 Update the directory structure and schema.

Release 5 Access Manager coexists with the Release 4 directory structure, but the structure must be modified to support Release 5 capabilities. Update the Access Manager directory structure and schema to Release 5 by running the `amupgrade.bat` file, which is installed in the *AccessManager-base*`\upgrade`directory.

a. Obtain the values of the following parameters to be requested by the `amupgrade.bat`:

Parameter	Value
Directory Server Host	Set the fully qualified name: *hostname.domain.*
Directory Server Port	Specify a non-SSL port number Default: `389`.
Directory Manager DN	Default: `cn=Directory Manager`.
Directory Manager Password	
Access Manager Administrator User ID Default: *amadmin*	Default: `amadmin`.
Access Manager Administrator Password	
Enable Realm Mode	`Y/N: Yes` means Realm Mode is enabled and services data is migrated to new Realm tree. `No` (default) means services data remain in Legacy Mode.

b. Run the *AccessManager-base*`\upgrade\amupgrade.bat` file.

If the upgrade is successful, the script displays "Upgrade completed."

c. Check the following upgrade log file for information about the directory schema extensions:

`AccessManager-base\setup\AccessManager_upgrade_`*num*`.log`

8 Enable the components disabled during reconfiguration of Access Manager.

9 Start Access Manager.

Restart the web container in which Access Manager is deployed.

▼ To Reconfigure and Redeploy Access Manager

1 If you chose to upgrade your web container software, as described in "Upgrading Access Manager Dependencies" on page 88, make sure the upgrade is complete

2 **Make sure that the administrative instance of your web container is running, and is in a mode supported by the `amconfig.bat` file, as indicated in the table below:**

Web Container	Supported Mode	Default Port Number
Application Server (8.*x*): Java ES 4 and 5	SSL (secure) non-SSL	4849
Web Server (7.0): Java ES 5	SSL (secure)	8989
Web Server (6.*x*): Java ES 4	non-SSL	8888

3 **If the web container is running in SSL mode, make sure that the container's SSL certificates have not expired and are still valid.**

4 **If Access Manager is deployed in Release 5 Web Server, disable all Java ES components depending on Access Manager that are running in the same instance as Access Manager.**

These components would likely be components such as Portal Server or Sun Java Communications Suite; Communications Express, Instant Messaging, or Delegated Administrator. The procedure is as follows:

a. **Log in as administrator at** *https://host:8989*

b. **Go to Edit Virtual Server.**

c. **Select the Web Applications tab.**

d. **Select all Access Manager dependent applications.**

e. **Click Disable.**

f. **Click Save.**

g. **Click deployment pending | Deploy Config.**

The configuration change will propagate to the Web Server instance

5 **Check that Directory Server and the appropriate web container are running.**

6 **Set the configuration parameters in the `AMConfigurator.properties` file.**

Some of the parameter values can be migrated from the `AMConfig.properties` file and others are more specific to the upgrade procedure, as shown in the following table.

Parameter	Value
Upgrade Parameters	
DEPLOY_LEVEL	Set to 26 for undeploy Set to 1 for reconfigure and deploy
DIRECTORY_MODE	Set to 5 (Existing Upgrade)
AM_REALM	Set to `disabled`. Because Realm Mode is disabled, Legacy Mode is therefore enabled
JAVA_HOME	Set to the JDK Release 5 directory
WEB_CONTAINER	Set to the value appropriate to the web container type you are using and fill out only the corresponding section of the configuration file.
WS61_INSTANCE (If using Web Server as the web container)	Set to `https-`*hostname.domain* where the value above matches the instance name in *install-dir*`\webserver` The value is case-sensitive.
AS81_INSTANCE (Using Application Server 8.*x* as the web container)	Set to Application Server.*x instanceName* Default: `server`
AS81_INSTANCE _DIR (Using Application Server 8.*x* as the web container)	Set to Application Server.*x* `domain` directory for the instance Default: *AppServer8Config-base*`\domains\domain1`
AS81_DOCS_DIR (Using Application Server 8.*x* as the web container)	Set to Application Server.*x* `docroot` directory for the instance Default: *AppServer8Config-base*`\domains\domain1\docroot`
AS81_ADMIN_IS_SECURE (Using Application Server 8.*x* as the web container)	Set to `false` Default: `true`
Migrated from `AMConfig.properties`	
SERVER_PROTOCOL	`com.iplanet.am.server.protocol`
SERVER_PORT	`com.iplanet.am.server.port`
SERVER_HOST	`com.iplanet.am.server.host`
DS_HOST	`com.iplanet.am.directory.host`
DS_PORT	`com.iplanet.am.directory.port`
ROOT_SUFFIX	`com.iplanet.am.defaultOrg`
CONSOLE_DEPLOY_URI	`com.iplanet.am.console.deploymentDescriptor`
SERVER_DEPLOY_URI	`com.iplanet.am.services.deploymentDescriptor`
PASSWORD_DEPLOY_URI	`com.sun.identity.password.deploymentDescriptor`

Parameter	Value
AM_ENC_PWD	`am.encryption.pwd`

For other parameters, provide the same values that were used in the Release 4 configuration that you are upgrading, unless you are changing web container or passwords. For example, if you have upgraded Web Server to Release 5, provide the values from the following table.

Parameter	Value
WS_CONFIG	The name of the Web Server configuration: *configName*
WS_INSTANCE	`https-`*configName*
WS_HOME	*WebServer7-base*
WS_PROTOCOL	`http` or `https`
WS_HOST	Fully qualified host name on which Web Server instance is listening for connections
WS_PORT	Port on which Web Server instance is listening for connections
WS_ADMINPORT	Port on which Web Server administration instance is listening for connections
WS_ADMIN	Web Server administrator user ID
WS_ADMINPASSWORD	Web Server administrator password

7 Run *AccessManager-base*`\setup\amconfig.bat` **to undeploy Access Manager.**

a. Set the value of *DEPLOY_LEVEL* **to 26 in the** *AccessManaget-base*`\setup\AMConfigurator.properties` **file.**

b. Change to the setup directory.

`cd` *AccessManager-base*`\setup`

c. Run `amconfig.bat` **file.**

8 Run *AccessManager-base*`\setup\amconfig.bat` **to reconfigure Access Manager and deploy into web container.**

a. Set the value of *DEPLOY_LEVEL* **to 1 in the** *AccessManager-base*`\setup\AMConfigurator.properties` **file.**

b. Change to the setup directory.

`cd` *AccessManager-base*`\setup`

c. Run `amconfig.bat` **file.**

▼ To Verify the Access Manager Upgrade

◗ **Type the following command.**

AccessManager-base`\bin\amadmin --version`

The outputs that indicate the Access Manager version are:

Release 5 Access Manager 7.1

Release 4 Access Manager 7 2005Q4

Post-Upgrade Tasks

If you are using the Security Assertion Markup Language (SAML) service, you must add and enable a SAML authentication module using the Access Manager console. For information on creating a SAML authentication module instance, see the *Sun Java System Access Manager Administration Guide*.

Coexistence of Release 5 with Earlier Directory Data

In some deployment architectures Access Manager is deployed on multiple computer systems to provide for high availability and scalability. The Access Manager instances access the same Directory Server. It is often desirable to upgrade the Access Manager instances sequentially without interrupting service. This section discusses the procedure for performing such rolling upgrades.

Note – Upgrading multiple instances of Access Manager installed on the same host system is not supported in the current release. If you have multiple instances on the same host, after you upgrade the main instance, you must then recreate the additional instances.

The procedure for upgrading Access Manager from Release 4 includes a step for updating directory schema to support Release 5. Release 4 Access Manager does not support Release 5 directory schema, however Release 5 Access Manager does support Release 4 directory schema.

Java ES Release 5 Access Manager and Release 4 Access Manager instances can coexist and run concurrently against the same Directory Server only if the directory schema has not yet been updated to Release 5. Therefore, in rolling upgrades, the directory schema should not be updated to Release 5 until all other steps in the upgrade procedure have first been performed for all Access Manager instances.

That is, in performing rolling upgrades, upgrade each instance of Access Manager as described "Upgrading Java ES Release 5 Access Manager from Java ES Release 4" on page 87, except for the step on updating directory structure and schema. When all instances have been upgraded, then that step can be performed.

Index