Sun Java™ System

Sun Java Enterprise System 2005Q4 Upgrade Guide

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

Part No: 819-2331-13

Contents

List of Tables

Preface

The *Java Enterprise System Upgrade Guide* contains the information you need to upgrade Sun Java™ Enterprise System (Java ES) software in a Sun Solaris™ Operating System (Solaris OS) or Linux operating system environment. The Guide covers upgrades from Java ES 2004Q2 (Release 2) and Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4).

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 book 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
- Clustering model (if you are installing clustering software)

Conventions Used in This Book

The tables in this section describe the conventions used in this book.

Typographic Conventions

The following table describes the typographic changes used in this book.

Table 1 Typographic Conventions

Typeface	Meaning	Examples
AaBbCc123 (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, onscreen computer output, sample code.	Edit your.login file. Use ls -a to list all files. % You have mail.
AaBbCc123 (Monospace bold)	What you type, when contrasted with onscreen computer output.	% **su** Password:
AaBbCc123 (Italic)	Book titles, new terms, words to be emphasized. A placeholder in a command or path name to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file. The file is located in the *install-dir*/bin directory.

Symbols

The following table describes the symbol conventions used in this book.

Table 2 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
>	Indicates menu item selection in a graphical user interface.	File > New > Templates	From the File menu, choose New. From the New submenu, choose Templates.

Shell Prompts

The following table describes the shell prompts used in this book.

Table 3 Shell Prompts

Shell	Prompt
C shell on UNIX or Linux	*machine-name*%
C shell superuser on UNIX or Linux	*machine-name*#
Bourne shell and Korn shell on UNIX or Linux	$
Bourne shell and Korn shell superuser on UNIX or Linux	#
Windows command line	C:\

Related Documentation

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

Books in This Documentation Set

The Java ES 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™ documentation web site can be accessed here:

http://docs.sun.com

The Java ES documentation includes information about the system as a whole and information about its components. This documentation can be accessed here:

http://docs.sun.com/prod/entsys.05q4

The following table lists the system-level manuals in the Java ES 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 4 Java Enterprise System Documentation

Document	Contents
Java Enterprise System Release Notes http://docs.sun.com/doc/819-2329	Contains the latest information about Java Enterprise System, including known problems. In addition, components have their own release notes.
Java Enterprise System Roadmap http://docs.sun.com/doc/819-2327	Provides descriptions of all documentation related to Java Enterprise System, both as a system and for the individual components.
Java Enterprise System Technical Overview http://docs.sun.com/doc/819-2330	Introduces the technical and conceptual foundations of Java Enterprise System. Describes components, the architecture, processes, and features.
Java Enterprise System Deployment Planning Guide http://docs.sun.com/doc/819-2326	Provides an introduction to planning and designing enterprise deployment solutions based on Java Enterprise System. 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 Enterprise System.

Table 4 Java Enterprise System Documentation *(Continued)*

Document	Contents
Java Enterprise System Installation Planning Guide http://docs.sun.com/doc/819-3933	Helps you develop the implementation specifications for the hardware, operating system, and network aspects of your Java Enterprise System deployment. Describes issues such as component dependencies to address in your installation and configuration plan.
Java Enterprise System Installation Guide for UNIX http://docs.sun.com/doc/819-2328	Guides you through the process of installing Java Enterprise System on the Solaris™ Operating System or the Linux operating system. Also shows how to configure components after installation, and verify that they function properly.
Java Enterprise System Installation Reference http://docs.sun.com/doc/819-3765	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.
Java Enterprise System Deployment Example Series: Evaluation Scenario http://docs.sun.com/doc/819-0059	Describes how to install Java Enterprise System on one system, establish a set of core, shared, and networked services, and set up user accounts that can access the services that you establish.
Java Enterprise System Upgrade Guide http://docs.sun.com/doc/819-2331	Provides instructions for upgrading Java Enterprise System on the Solaris™ Operating System or the Linux operating environment.
Java Enterprise System Glossary http://docs.sun.com/doc/819-3875	Defines terms that are used in Java Enterprise System documentation.

Accessing Sun Resources Online

For product downloads, professional services, patches and support, and additional developer information, go to the following:

- Download Center
  http://www.sun.com/software/download/
- Client Solutions
  http://www.sun.com/service/sunjavasystem/sjsservicessuite.html
- Sun Enterprise Services, Solaris Patches, and Support
  http://sunsolve.sun.com/
- Developer Information
  http://developers.sun.com

The following location contains information about Java ES and its components:

http://www.sun.com/software/javaenterprisesystem/index.html

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in the product documentation, go to http://www.sun.com/service/contacting.

Related Third-Party Web Site References

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.

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 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 book or at the top of the document.

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 2005Q4 (Release 4). It contains the following sections:

Java ES 2005Q4 (Release 4) Components

As an introduction to planning the upgrade of Java ES software, this section reviews the components included in Java ES Release 4. Depending on your upgrade scenario, you might need to upgrade one or more of these components to their Release 4 version.

Java ES components are grouped into different types, as described in the *Java Enterprise System Technical Overview* (http://docs.sun.com/doc/819-2330). Accordingly, system service components provide the main Java ES infrastructure services, while service quality components enhance those system services. These two types of Java ES components are together referred to here as *product* components, components that are selectable within the Java ES installer.

Each product component depends on one or more locally shared libraries known as Java ES *shared* components. Shared components are installed automatically by the Java ES installer during product component installation, depending on the product components that are being installed.

Release 4 Product Components

The Java ES Release 4 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 Release 4 Product Components

Product Component	Version	Type	Short Name
Access Manager	7.0	System service component	AM
Administration Server	5.2	Service quality: administrative component	ADS
Application Server	8.1	System service component	AS
Calendar Server	6.2	System service component	CS
Communications Express	6.2	Service quality: access component	CX
Delegated Administrator	6.3	Service quality: administrative component	DA
Directory Preparation Tool	6.3	Service quality: administrative component	DPT
Directory Proxy Server	5.2	Service quality: access component	DPS
Directory Server	5.2	System service component	DS
High Availability Session Store	4.4.2	Service quality: availability component	HADB

Table 1-1 Java ES Release 4 Product Components *(Continued)*

Product Component	Version	Type	Short Name
Instant Messaging	7.0.1	System service component	IM
Message Queue	3.6 SP3	System service component	MQ
Messaging Server	6.2	System service component	MS
Portal Server	6.3	System service component	PS
Portal Server Secure Remote Access	6.3	Service quality: access component	PSRA
Service Registry	3.0	System service component	SR
Sun Cluster	3.1 8/05	Service quality: availability component	SC
Web Proxy Server	4.0.1	Service quality: access component	WPS
Web Server	6.1 SP%	System service component	WS

Release 4 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.

The Java ES Release 4 shared components are listed in the following table.

Table 1-2 Java ES Release 4 Shared Components

Shared Component	Version	Abbreviation
Apache Commons Logging	1.0.3	ACL
Jakarta ANT Java/XML-based build tool	1.6.2	ANT
Berkeley Database	4.2.52	BDB
Common agent container	1.1	CAC
International Components for Unicode	3.2	ICU
Instant Messenger SDK	6.2.8	IM-SDK
Java 2 Platform, Standard Edition	5.0 Update 3	J2SE™
JavaBeans™ Activation Framework	1.0.3	JAF
Java Studio Enterprise Web Application Framework	2.1.5	JATO

Table 1-2 Java ES Release 4 Shared Components *(Continued)*

Shared Component	Version	Abbreviation
JavaHelp™ Runtime	2.0	JHELP
JavaMail™ Runtime	1.3.2	JMAIL
Java Architecture for XML Binding Runtime	1.0.4	JAXB
Java API for XML Processing	1.2.6	JAXP
Java API for XML Registries Runtime	1.0.7	JAXR
Java API for XML-based Remote Procedure Call Runtime	1.1.2	JAX-RPC
Java Calendar API	1.2	JCAPI
Java Dynamic Management™ Kit Runtime	5.1	JDMK
Java Security Services	4.1	JSS
KT Search Engine	1.3.2	KTSE
LDAP C SDK	5.11	LDAP C SDK
LDAP Java SDK	4.18	LDAP J SDK
Mobile Access Core	1.0.6	MA Core
Netscape Portable Runtime	4.5.2	NSPR
Network Security Services	3.10	NSS
SOAP Runtime with Attachments API for Java	1.2.1	SAAJ
Simple Authentication and Security Layer	2.18	SASL
Sun Explorer Data Collector	4.3.1	SEDC
Sun Java Enterprise System Monitoring Framework	1.0.1	MFWK
Sun Java Web Console	2.2.4	SJWC
Web services Common Library	1.0	WSCL

About Java ES Upgrades

The upgrade of Java ES software to Release 4 is not generally performed using the Java ES installer or any other system utility. It is performed component-by-component, computer-by-computer, using component-specific upgrade procedures.

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 impact 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:

- **Installing upgraded software.** The new software can enhance or fix existing software, or replace existing software. In general, the new software is achieved through the application of patches to existing software packages, the replacement of existing packages, the installation of new packages, or a full re-installation of a component using the Java ES installer.
- **Re-configuration.** Re-configuration 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 re-configuration requires that you perform an explicit procedure and sometimes it takes place automatically without your involvement.

These two aspects of component upgrades are described in this *Upgrade Guide* for each of the Java ES product components.

The *Upgrade Guide* also covers other important aspects of product component upgrades, including:

- dependencies that impact an upgrade
- operations you might need to perform before you upgrade a component
- operations you perform to verify successful upgrade
- operations you perform if you need to roll back an upgrade

Shared Component Upgrades

Java ES shared component upgrades are often a necessary part of upgrading the product components that depend on them.

The upgrading of shared components is typically more straightforward than the upgrading of product components. In general, the upgrade is achieved through the application of patches to existing packages or the replacement of existing packages. As compared to upgrading product components, there is normally no re-configuration required, nor pre or post upgrade procedures to be performed.

While shared components can be upgraded one by one, Java ES Release 4 allows you to collectively upgrade a number of shared components in one operation. For more information, see Chapter 2, "Upgrading Java ES Shared Components."

Upgrade Technologies

The upgrade of both product components and shared components, as described in this *Upgrade Guide*, involves the modification or replacement of currently installed software packages and, in some cases, the installation of new packages. Solaris and Linux platforms employ similar technologies for managing installed software packages and tracking changes through a package registry.

- **Solaris platform.** Java ES packages can be installed and removed through the Solaris `pkgadd` and `pkgrm` commands, using packages found on the Java ES software distribution. Package contents, once installed, can be modified using patches that are applied or removed through the `patchadd` and `patchrm` commands. Patches to Solaris packages are distributed through the SunSolve website at:
  `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

  Solaris patches can patch one or more packages. The `patchadd` command saves a backup of the package being patched to facilitate the removal of the patch using the `patchrm` command. Patches are identified by a patch ID, which consists of a patch number followed by a revision number that is incremented as the patch is modified over time.

  Solaris patches can also be collected into a patch cluster. The patch cluster lets you collectively download and apply all the patches in the cluster. Patch clusters are provided for upgrading Java ES shared components (see Chapter 2, "Upgrading Java ES Shared Components").

- **Linux platform.** Java ES RPM (Red Hat Package Manager) packages can be installed or updated through the `rpm` command, using packages found on the Java ES software distribution. Package contents, once installed, however, cannot be modified using patches. Rather, RPM packages are updated using the `rpm -U` command option, which replaces the current package with a newer package.

  As a convenience, many RPM package upgrades are distributed not only on the Java ES software distribution, but also through the SunSolve website at: `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

  For distribution through SunSolve, RPM packages can be wrappered as patches and assigned a patch ID and revision number similar to Solaris patches. These Linux patches can include one or more RPM packages, each identified by a unique RPM name, RPM number, and a revision number that is incremented as the RPM package is modified over time.

Operating System Issues

A number of operating system issues impact the upgrading of Java ES software, as described below.

Required Operating System Patches

In some situations, the successful upgrade of a Java ES product component can require you to first patch the operating system or apply specific fixes. Rather than applying the specific operating system patch required in each case, it is generally preferable to simply bring the operating system up to date before performing Java ES upgrades.

- Solaris platform patches are available through the SunSolve website as a patch cluster, a collection of operating system patches that can be collectively applied. The operating system patch clusters for Solaris 8, 9, and 10 are available at:
  `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`
- Linux platform update releases are available at:
  `https://www.redhat.com/apps/download/`

Minor Release Upgrades

A significant number of Java ES shared components have Solaris release-specific packages. The release-specific packages might not function correctly on other Solaris platforms. For example, packages that are released for the Solaris 8 operating system cannot be expected to work for Solaris 9 or Solaris 10 operating systems.

When upgrading the operating system from one minor release to another, the various installed Java ES shared components will be affected. When shared components have release-specific packages, these packages will also need to be upgraded after the operating system is upgraded, to match the newly upgraded operating system.

Upgrades to Non-Supported Platforms

Java ES 2004Q2 (Release 2) is supported on Solaris 8 and 9 operating systems and on Red Hat Enterprise Linux (RHEL) 2.1. If you wish to upgrade your operating system platform to Solaris 10 or RHEL 3.0, which are not supported by Java ES Release 2, you will also need to upgrade your Java ES Release 2 to a Java ES Release that supports the upgraded platform, preferably to Java ES Release 4.

Because upgrade of some Java ES components requires that other Java ES components be running, you cannot, as a general rule, upgrade your operating system platform to Solaris 10 or RHEL 3.0 before upgrading Java ES from Release 2 (Java ES Release 2 does not support these platforms).

Instead, the approach you need to take depends on platform:

- **Linux platform.** You should upgrade Java ES Release 2 to Release 4 first and then perform the upgrade to RHEL 3.0.
- **Solaris platform.** You should uninstall Java ES Release 2, upgrade your operating system to Solaris 10, and then perform a fresh install of Java ES Release 4. Such an operation means that all Java ES components must be freshly configured. In this situation it would be prudent to back up all Java ES Release 2 configuration files and customizations for use in configuring Java ES Release 4 components.

Upgrade Planning

The approach you take to upgrading a deployed Java ES software system to Java ES Release 4 can 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.

These two examples represent upgrade scenarios of very different complexities, requiring substantially different upgrade plans. No one plan works for all deployed Java ES software systems.

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

What is an Upgrade Plan?

An upgrade plan specifies how to approach each stage of the upgrade process. This process involves, at a minimum, the phases shown in the following table.

Table 1-3 Phases in the Upgrade Process

Upgrade Phase	Description
Preparation	You develop an upgrade plan. In it, you specify the Java ES components you need to upgrade and the sequence by which you need to upgrade those components on the various computers in your system. You also plan how to test upgrade procedures in a staging environment before executing them in your production environment. In this step, you also back up your current system and test your ability to restore it to its current configuration.
Execution	You obtain all the necessary packages, patches, and tools needed for the upgrade. You execute the upgrade and re-configuration of your Java ES deployed system in a staging environment. This involves the backup of configuration and application data, the upgrade of system software, and the re-configuration or migration of data to the upgraded system.

Table 1-3 Phases in the Upgrade Process *(Continued)*

Upgrade Phase	Description
Verification	You start the upgraded software components and perform verification tests as you proceed. If verification is not successful, and problems cannot be resolved within a reasonable time frame, you might be forced to roll back the upgrade and restore the system to its previous state.
Rollback/restoration	If necessary, you restore the system to its previous state as specified in the preparation phase. You also perform tests to verify that the rollback is successful.

The following sections provide information that can help in formulating an upgrade plan.

Upgrade Plan Considerations

Your upgrade plan will depend on a number of factors beyond the scope and complexity of your deployment architecture. These factors include the following considerations:

- Your upgrade path
- The dependencies between deployed Java ES components
- The possibility of performing selective upgrade
- Multi-instance upgrades

These factors are discussed in the following sections.

Upgrade Paths

While it is possible to upgrade all previous releases of Java ES software to Java ES 2005Q4 (Release 4), the only certified upgrades are from Java ES 2005Q1 (Release 3) and Java ES 2004Q2 (Release 2). Upgrades from earlier releases are not documented in this *Upgrade Guide*.

The various upgrade paths involve different upgrade strategies, as described in Table 1-4 on page 35.

Because of the different characteristics of the Release 3 to Release 4 and the Release 2 to Release 4 upgrade paths, and because the upgrade procedures for product components often depend on the upgrade path, the separate chapters in this *Upgrade Guide* describing the upgrade of each product component are divided into two sections: one on upgrading from Release 3 to Release 4 and one on upgrading from Release 2 to Release 4.

Table 1-4 Upgrade Paths to Java ES 2005Q4 (Release 4)

Product Number	Java ES Release	System Characteristics	Upgrade Strategies
2005Q1	Release 3	Java ES Release 4 supports a mixture of Release 3 and Release 4 components on a single computer. This includes both product components and shared components. Compatibilities between Release 3 and Release 4 components have been tested, and any known incompatibilities are noted in the *Java Enterprise System Release Notes* (http://docs.sun.com/doc/819-2329).	The coexistence of Release 3 and Release 4 components provides for the possibility of selectively upgrading Release 3 components to Release 4 on a given computer, or within a deployment architecture consisting of multiple computers.
2004Q2	Release 2	Java ES Release 4 does not support a mixture of Release 2 and Release 4 components on a single computer. This includes both product components and shared components. There are known incompatibilities between the release versions, and interoperability between Release 2 and Release 4 components is not certified.	When upgrading components from Release 2 to Release 4 on a given computer, all Release 2 components should be upgraded to Release 4. However, assuming compatibility of components, it is possible to mix Release 2 and Release 4 components residing on *different* computers within a deployment architecture consisting of multiple computers.
2003Q4 and prior	Release 1 and prior	Java ES Release 4 does not support a mixture of Release 1 or prior releases and Release 4 components on a single computer. This includes both product components and shared components. There are known incompatibilities between the release versions, and interoperability between Release 1 or prior components and Release 4 components is not certified.	Java ES Release 4 does not certify the direct upgrade of Release 1 or prior releases to Release 4. In some cases, however, It is possible to perform an upgrade from Release 1 by upgrading first to Java ES Release 3, as documented in the Release 3 *Java Enterprise System Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062). In other cases the upgrade from Release 1 to Release 4 can be performed in the same way as the upgrade from Release 2 or Release 3 to Release 4, and in those cases, the upgrade roadmap for that component in this *Upgrade Guide* notes this possibility.

NOTE Some product components have issued interim releases that fall between the official Java ES releases. In such cases the upgrade of the interim release should be performed using the same procedure as for the previous Java ES release. For example, if an interim release took place between Release 2 and Release 3, the component would be upgraded using the procedure for upgrading from Release 2 to Release 4.

Upgrade Dependencies

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

In this respect, there are two types of upgrade dependencies:

- **Hard upgrade dependency.** A hard upgrade dependency is when an upgraded version of a component requires an upgraded version of some component upon which it has a dependency. This requirement can be due to new functionality, new interfaces, or bug fixes needed by the dependent component. You cannot successfully upgrade and use the component without first upgrading the component upon which it depends.
- **Soft upgrade dependency.** A soft upgrade dependency is when an upgraded version of a component does not require an upgraded version of some component upon which it has a dependency. You can successfully upgrade and use the component without upgrading the component upon which it depends.

Upgrading a Java ES component requires you to upgrade all the components upon which it has hard upgrade dependencies, but allows you to not upgrade components upon which it has soft upgrade dependencies. (This general rule does not apply to upgrades from Release 2 to Release 4 on a single computer.)

This general rule does not necessarily apply, however, when multiple interdependent components are involved in an upgrade. In such cases, you have to upgrade a component if only one of several other Java ES components has a hard upgrade dependency on that particular component.

Selective Upgrade or Upgrade All

The difference between hard and soft upgrade dependencies allows for the possibility of selectively upgrading Java ES components within a deployed system. This possibility only applies to upgrading from Release 3 to Release 4 on a single computer (see upgrade path characteristics in "Upgrade Paths" on page 34). Selective upgrade from Release 2 to Release 4 on a single computer is not supported.

- **Selective Upgrade.** The selective upgrade approach starts with the Java ES component you wish to upgrade to Release 4. Determine the hard upgrade dependencies for that component, which includes dependencies on both product components and shared components. Those components also need to be upgraded. You repeat this process for each successive hard upgrade dependency until no further components need to be upgraded. This exercise specifies all Java ES components that need to be upgraded.

  The selective upgrade approach can be simple or quite complex, depending on your deployment architecture and the hard upgrade dependencies involved.

- **Upgrade All.** As an alternative, you can upgrade all deployed Java ES components to Release 4. The complexity of this approach also depends on your deployment architecture. In some cases, it simply is not feasible for business reasons to upgrade an entire system at one time.

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

Table 1-5 Selective Upgrade Compared to Upgrade All

Upgrade Approach	Advantages	Disadvantages
Selective Upgrade	Minimizes number of components to upgrade	You must track the version of each component in your deployed system
Upgrade All	A consistent version for all components in your deployed system	Maximizes 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 3 to Release 4, selectively upgrading product components, while upgrading all of the corresponding shared components, is often the preferred approach.

Multi-instance Upgrades

The sequence of upgrade procedures can depend on whether and how redundancy is being used in a deployment architecture. Multiple instances of a Java ES component can be used to achieve high availability, scalability, serviceability, or some combination of these service qualities. There are three technologies that make use of redundant components in Java ES deployment architectures: load balancing, high availability techniques (Sun Cluster and High Availability Session Store), and multimaster replication (Directory Server).

In most cases where redundancy is involved, it is desirable to perform upgrades without incurring downtime. These rolling upgrades attempt to successively upgrade redundant instances of a component without compromising the service that they provide.

In most cases the redundant instances are deployed across multiple computers. From an upgrade planning perspective, this might imply isolating the upgrade of such replicated components from other component upgrades in order to achieve minimal downtime. In other words, you might perform all the pre-upgrade tasks for the component on each computer before performing a rolling upgrade of the replicated component.

Each replication technology has configuration or re-configuration procedures that might affect the overall sequence of Java ES component upgrades. For example, components that run in a Sun Cluster environment can require upgrading Sun Cluster before upgrading the components that are running in the Sun Cluster environment.

Java ES Component Dependencies

As mentioned in the previous section, an upgrade plan specifies the Java ES components you need to upgrade and the sequence by which you need to upgrade those components. One of the important considerations in an upgrade plan is the dependencies between the various Java ES components in your deployed system.

Whether you take a selective upgrade approach or you upgrade all components, the sequence by which you perform the component upgrades is affected by the nature of the dependencies between them.

This section provides information about Java ES component dependencies. The following dependency factors impact your upgrade plan.

- Dependencies On Shared Components
- Dependencies On Product Components
- Multi-instance Upgrades

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

Dependencies On Shared Components

When upgrading Java ES product components, you have to take into account dependencies these Java ES components have on Java ES shared components. When a product component has a hard upgrade dependency on a shared component, the shared component also must be upgraded.

Shared Component Dependency Matrix

Table 1-6 on page 40 shows the dependencies of Java ES 2005Q4 (Release 4) product components on Java ES shared components. The abbreviations for product components that head the columns of Table 1-6 are taken from Table 1-1 on page 26. The abbreviations for shared components are spelled out in Table 1-2 on page 27.

Four product components are not included in Table 1-6: Directory Proxy Server (DPS), High Availability Session Store (HADB), and Directory Preparation Tool (DPT) have been omitted because they have no dependencies on shared components. Service Registry (SR) is omitted because it is a new product component for which there is no previous version from which to upgrade. Web Proxy Server (WPS), another new Release 4 product component, however, is included in Table 1-6 because it can be upgraded to Release 4 from its previous release, which was not included in Java ES.

Within the matrix of Table 1-6 hard upgrade dependencies for Release 3 to Release 4 upgrades are marked “H,” while soft upgrade dependencies are marked “S.” For Release 2 to Release 4 upgrades, all shared component dependencies are, by definition, hard upgrade dependencies; all shared components must be upgraded from Release 2 to Release 4.

Table 1-6 Shared Component Dependencies of Java ES Release 4 Product Components

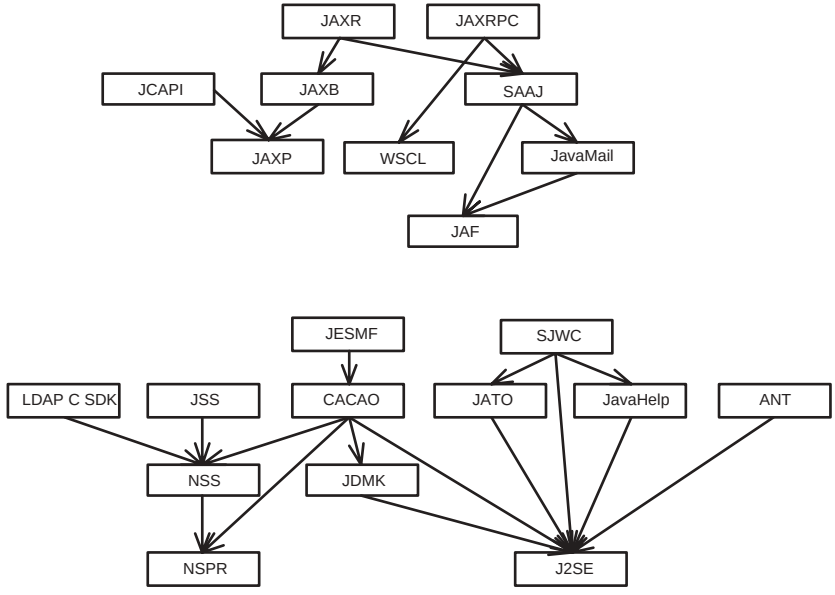
Shared Component	AM	ADS	AS	CS	CX	DA	DPS	DS	IM	MQ	MS	PS	PSRA	SC	WPS	WS
ANT			S													
ACL	S															
BDB	S															
CAC									S					S		
ICU		S	S	S			S	S			H	S			S	S
IM-SDK									H			S				
J2SE™	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
JAF	S		S		S				S	S		S	S			
JATO	S		S		S	S						S				
JavaHelp™	S		S							S						
JavaMail ™	S		S		S				S	S		S	S			
JAXB	S		S													
JAXP	S		S		S				S	S		S	S			
JAXR	S		S													
JAX-RPC	S		S													
JCAPI					S				S							
JDMK			S						S					S		
JSS	S	S		S		S	S	S		S		S	S		S	S
KTSE												S			S	S
LDAP C SDK		S		S			S	S			H				S	S
LDAP J SDK	S	S			S	S	S	S								
MA Core	S											H	H			
MFWK									S							
NSPR	S	S	S	S		S	S	S	S	S	H	S	S	S	S	H

Table 1-6 Shared Component Dependencies of Java ES Release 4 Product Components *(Continued)*

Shared Component	AM	ADS	AS	CS	CX	DA	DPS	DS	IM	MQ	MS	PS	PSRA	SC	WPS	WS
NSS	S	S	S	S		S	S	S	S	S	H	S	S	S	S	H
SAAJ	S		S							S		S	S			
SASL		S		H	H		S	S			H				S	S
SEDC														S		
SJWC	S		S											S		
WSCL	S		S													

The dependencies shown in Table 1-6 for any product component represent both direct and indirect shared component dependencies. In other words, a product component might depend on a specific shared component that, in turn, depends on one or more other shared components. The shared component dependencies shown in Table 1-6 include all such indirect dependencies. The following figure illustrates inter-dependencies among shared components.

Figure 1-1 Shared Component Inter-dependencies



Shared Component Upgrade Guidelines

Table 1-6 lets you determine the shared components to upgrade when upgrading one or more product components on a given computer:

- **Release 2 to Release 4 Upgrades.** If you are performing an upgrade from Release 2 to Release 4, all the shared components marked as “S” or “H” in Table 1-6 for the respective product components must be upgraded.
- **Release 3 to Release 4 Upgrades.** If you are upgrading all product components from Release 3 to Release 4, all the shared components indicated in Table 1-6 for the respective product components should be upgraded.

  Even if you are selectively upgrading product components, however, it is the recommended practice to upgrade the shared components needed by all product components on the computer; Release 4 shared components are certified to support Release 3 product components.

  While selectively upgrading shared components might work in most cases (that is, upgrading only those shared components that support selectively upgraded product components, or upgrading only hard upgrade dependencies compared to soft upgrade dependencies), significantly more risk is involved in adopting this approach.

  If no hard upgrade dependencies are involved, you might not upgrade shared components at all. However, as a general rule, it is a good practice to upgrade your underlying Java ES shared component base to the most current versions.

NOTE The sequence of upgrading shared components can depend upon the shared component inter-dependencies shown in Figure 1-1.

Also, if you plan to upgrade J2SE to J2SE 5.0, you should upgrade this shared component first. J2SE is the base component for many Java ES components.

For information on how to upgrade shared components, consult Chapter 2, “Upgrading Java ES Shared Components.”

Dependencies On Product Components

Dependencies of one product component on another are an important determinant of the Java ES components you need to upgrade and the sequence by which you need to upgrade them. 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 Technical Overview*. In upgrading any Java ES product component, you must take into account such dependencies. If an upgraded version of one component has a hard upgrade dependency on another component, that dependency implies that the dependent component should only be upgraded after the component upon which it depends is upgraded.
- **Configuration Dependencies.** In many cases a Java ES component must be installed, configured, and running for another component to be configured. For example, a Directory Server configuration directory must be running for you to configure Messaging Server components, or a Directory Server user/group directory must be running for an Access Manager service to be registered. Component upgrade procedures often involve re-configuration of upgraded components or migration of configuration data. In fact, some product components' main function is to provide configuration or administrative support to other components. As a result, configuration dependencies can have a strong impact on the sequence of upgrade procedures.

Table 1-7 shows the dependencies between the Java ES product components listed in Table 1-1 on page 26. Using Table 1-7, you can diagram the chain of dependencies in your upgrade set. The left column lists each product component, the middle column shows its dependencies on other product components, the third characterizes each dependency, and the last column indicates whether or not the respective components must be local.

Table 1-7 Java ES Product Component Dependencies

Product Component	Dependencies	Nature of Dependency	Must be Local?
Access Manager	Directory Server	To store configuration data and enable lookup of user data	No
	J2EE web container, one of: - Application Server - Web Server - BEA WebLogic Server - IBM WebSphere Application Server	To provide web container runtime services	Yes
Access Manager SDK	Access Manager	To provide Access Manager services	No
	J2EE web container, one of: - Application Server - Web Server - BEA WebLogic Server - IBM WebSphere Application Server	To provide web container runtime services	Yes
Administration Server	Directory Server	To provide configuration directory	No
Application Server	Message Queue	To provide reliable asynchronous messaging	Yes
	Web Server (optional)	To provide for load balancing between instances	Yes
	High Availability Session Store (optional)	To store session state needed to support failover between instances	Yes
Calendar Server	Directory Server	To store and enable lookup of user data	No
	Directory Preparation Tool	To prepare directory for use by Calendar Server	No
	Access Manager (optional)	To provide single sign-on	No
	Messaging Server (optional)	To provide email notifications	No
	Delegated Administrator (optional)	To provision users for calendar services	No

Table 1-7 Java ES Product Component Dependencies *(Continued)*

Product Component	Dependencies	Nature of Dependency	Must be Local?
Communications Express	J2EE web container, one of: - Application Server - Web Server	To provide web container runtime services	Yes
	Directory Server	To store and enable lookup of user data, such as in address books	No
	Directory Preparation Tool	To prepare directory for use by Communications Express	No
	Access Manager or Access Manager SDK	To provide authentication and authorization services, single sign-on	Yes
	Messaging Server	To enable web-based access to messaging	No
	Calendar Server	To enable web-based access to calendaring	No
Delegated Administrator	J2EE web container, one of: - Application Server - Web Server	To provide web container runtime services	Yes
	Directory Server	To store user data	No
	Directory Preparation Tool	To prepare directory for use by Delegated Administrator	No
	Access Manager or Access Manager SDK	To provide API needed for user provisioning	Yes
Directory Preparation Tool	Directory Server	To provide the user/group directory that it is preparing for use by communications components	Yes
Directory Proxy Server	Administration Server	To configure Directory Proxy Server	No
	Directory Server	To provide access to a directory	No
Directory Server	Administration Server	To configure Directory Server	No
High Availability Session Store	None		
Instant Messaging	J2EE web container, one of: - Application Server - Web Server	To provide web container runtime services	Yes
	Directory Server	To store user data	No
	Access Manager (optional)	To provide single sign-on	No

Table 1-7 Java ES Product Component Dependencies *(Continued)*

Product Component	Dependencies	Nature of Dependency	Must be Local?
Message Queue	None		
Messaging Server Store MTA MMP MEM	Directory Server	To store configuration data and enable lookup of user data	No
	Administration Server	To store configuration data in Directory Server configuration directory	Yes
	Directory Preparation Tool	To prepare directory for use by Messaging Server	No
	Access Manager (optional)	To provide single sign-on	No
	Delegated Administrator (optional)	To provision users for messaging services	No
Portal Server	J2EE web container, one of: - Application Server - Web Server - BEA WebLogic Server - IBM WebSphere Application Server	To provide web container runtime services	Yes
	Directory Server	To store and enable lookup of user profiles	No
	Access Manager or Access Manager SDK	To provide authentication and authorization services, single sign-on	Yes
	Communications Express	To provide messaging and calendar channels	No
Portal Server Secure Remote Access	Portal Server	To provide access to a portal	Yes
	Access Manager or Access Manager SDK	To provide authentication and authorization services, single sign-on	Yes
Sun Cluster	None		
Sun Cluster Agents	Sun Cluster	To provide access to Sun Cluster services	
Web Proxy Server	None		
Web Server	None		

General Sequencing Guidelines

The factors discussed in the previous sections can all impact which Java ES components you plan to upgrade as well as the order in which you upgrade them. These factors also influence your approach to upgrading Java ES components that are deployed across multiple computers. The specific impact of all these factors depends on your deployment architecture.

Nevertheless a few general sequencing guidelines apply, though not in every case. The following list provides the order in which Java ES components can be successfully upgraded on a single computer or in a deployed system. When performing an upgrade, simply omit those components that are not part of your deployment architecture, or, if you are performing a selective upgrade, omit those components which are not part of your upgrade plan.

NOTE The chapters in this *Upgrade Guide* are arranged according to the order you would normally upgrade Java ES components, as indicated by these sequencing guidelines.

1. **Shared Components** (See Chapter 2, “Upgrading Java ES Shared Components” on page 51)

   Shared components should generally be upgraded before the components which depend on them.

2. **Sun Cluster software** (See Chapter 3, “Sun Cluster Software” on page 93)

   If any components run in a Sun Cluster environment, and the Sun Cluster software needs to be upgraded, it should be upgraded before the components that use Sun Cluster services. Sun Cluster agents, if upgraded, should be upgraded as part of the Sun Cluster upgrade.

3. **Directory Server and Administration Server** (See Chapter 4, “Directory Server and Administration Server” on page 103)

   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. Administration Server must be upgraded with Directory Server.

4. **Directory Proxy Server** (See Chapter 5, “Directory Proxy Server” on page 123)

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

5. **Web Server** (See Chapter 6, “Web Server” on page 137)

   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 or Application Server, but if your architecture contains both, upgrade Web Server first.

6. **Message Queue** (See Chapter 7, “Message Queue” on page 149)

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

7. **High Availability Session Store** (See Chapter 8, “High Availability Session Store” on page 167)

   High Availability Session Store, if upgraded, is best upgraded before Application Server, which requires High Availability Session Store for high availability.

8. **Application Server** (See Chapter 9, “Application Server” on page 175)

   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.

9. **Web Proxy Server** (See Chapter 10, “Web Proxy Server” on page 195)

   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 4 component that can be upgraded from its previous non-Java ES release.

10. **Access Manager** (See Chapter 11, “Access Manager” on page 203

    Access Manager plays a central role in authentication and authorization, including single sign-on, and, if upgraded, 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.

11. **Directory Preparation Tool** (See Chapter 12, “Directory Preparation Tool” on page 231)

    Directory Preparation Tool depends on the Directory Server schema and should therefore be run against Directory Server after Access Manager is upgraded. (For an exception to this guideline, see “Upgrading Access Manager from Java ES Release 2” on page 224.) If upgrading Directory Preparation Tool, it should be upgraded before you upgrade the communications components that depend on Directory Preparation Tool to make changes in the directory: Messaging Server, Calendar Server, Communications Express, and Delegated Administrator.

12. **Messaging Server** (See Chapter 13, “Messaging Server” on page 245)

    Messaging Server, if upgraded, should be upgraded only after the preceding upgrades and should be upgraded before Communications Express, which has a dependency on Messaging Server components.

13. **Calendar Server** (See Chapter 14, “Calendar Server” on page 263)

    Calendar Server, if upgraded, should be upgraded after Messaging Server since some of its functions require Messaging Server support. Calendar Server should be upgraded before Communications Express, which has a dependency on Calendar Server.

14. **Communications Express** (See Chapter 15, “Communications Express” on page 275)

    Communications Express, if upgraded, depends on many of the preceding components (Calendar Server, Messaging Server, Directory Preparation Tool, Access Manager, Web Server, and Directory Server) and, if upgraded, should be upgraded after them.

15. **Instant Messaging** (See Chapter 15, “Communications Express” on page 275)

    Instant Messaging, if upgraded, can be upgraded at almost any point after Access Manager has been upgraded.

16. **Portal Server** (See Chapter 17, “Portal Server” on page 311)

    Portal Server, like Communications Express, depends on many of the preceding components, but in particular, it depends on Communications Express to provide messaging and calendar channels, and, if upgraded, should therefore be upgraded after Communications Express

17. **Portal Server Secure Remote Access** (See Chapter 18, “Portal Server Secure Remote Access” on page 343)

    Portal Server Secure Remote Access, if upgraded, can be upgraded anytime after Portal Server has been upgraded.

18. **Delegated Administrator** (See Chapter 19, “Delegated Administrator” on page 367)

    Delegated Administrator, if upgraded, can be upgraded and used to provision users any time after Directory Preparation Tool has been upgraded and run against Directory Server. By convention, users are provisioned after other services have been upgraded and started, however, Delegated Administrator can be upgraded before upgrading the communications components that depend on Delegated Administrator for provisioning users.

Chapter 2

Upgrading Java ES Shared Components

This chapter provides information on upgrading Java ES shared components to Java ES 2005Q4 (Release 4).

This chapter contains the following sections:

NOTE To upgrade shared components in preparation for upgrading Sun Cluster software, follow the procedures for upgrading dependency software in "Upgrading Sun Cluster Software" in *Sun Cluster Software Installation Guide for Solaris OS*, which is available at:

`http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view`

However, install the packages for the security components from the Java ES 2005Q4 (Release 4) distribution rather than from the Sun Cluster 1 of 2 CD-ROM or the Sun Cluster 2 of 2 CD-ROM.

Shared Component Upgrade Overview

Upgrading shared components to Java ES 2005Q4 (Release 4) should be done as part of a larger upgrade plan, as discussed in Chapter 1, “Planning for Upgrades.” To ensure that you have a successful upgrade, read Chapter 1 carefully and prepare an upgrade plan that meets your needs.

About Your Upgrade Plan

Your upgrade plan should cover the following areas:

- **Operating System Issues.** Perform any operating system upgrades, as described in “Operating System Issues” on page 31. For all platforms except for Solaris 10 OS, perform operating system upgrades before you upgrade shared components.
- **Upgrade Path.** Determine which version of Java Enterprise System you currently have and make sure you understand the supported path to upgrade to Java ES Release 4. In most cases, when upgrading shared components, the upgrade procedures you follow are the same whether you are upgrading from Java ES 2004Q2 (Release 2) or Java ES 2005Q1 (Release 3). Procedures that depend on a specific upgrade path are noted in this chapter. For more information on upgrade path, refer to “Upgrade Paths” on page 34.
- **Upgrade Dependencies.** Understand the interdependencies of the product components you are upgrading. Typically, you sequence the upgrade of product components according to their dependencies. For example, before you upgrade a component you upgrade any component upon which it depends. There are various other factors to consider, such as hard and soft upgrade dependencies, as explained in “Upgrade Dependencies” on page 36.

  Use Table 1-6 on page 40 to determine which shared components need to be upgraded before you upgrade product components.
- **Upgrade All or Selective Upgrade.** If you are upgrading all product components on a computer, then you should upgrade all shared components upon which the product components depend. However, when upgrading from Release 3 to Release 4, you can selectively upgrade some product components on a computer without upgrading others. Nevertheless, best practice is to upgrade all shared components upon which all product components on the computer depend. Release 4 shared components are certified to support Release 3 product components. For more information, refer to “Shared Component Upgrade Guidelines” on page 42.

- **Sequencing Guidelines.** Review the sequencing guidelines listed in "General Sequencing Guidelines" on page 47. Typically, shared components are upgraded first. However, you should understand the entire sequence of your upgrade to Java ES Release 4 before beginning your upgrade process.

Technologies for Upgrading Shared Components

There are three technologies for upgrading shared components to Java ES Release 4. The technologies you use depend on the number and type of shared components you are upgrading, according to your upgrade plan.

The three technologies are:

- **Patches.** Most shared components on Solaris platforms can be upgraded to Java ES Release 4 through the application of patches. Patches typically upgrade a single component or a group of related components.

  If your upgrade plan calls for upgrading a few shared components, then you might consider applying individual patches to those components for which upgrade patches are available.

  Table 2-1 on page 56 shows the upgrade patches that are available for each shared component. "Upgrading Shared Components by Applying Individual Patches" on page 59 provides instructions for downloading and applying patches.

- **Patch Clusters.** A patch cluster bundles all the upgrade patches available for shared components. This simplifies the upgrade process because you can upgrade all the corresponding shared components by executing a single upgrade script provided with the patch cluster.

  There is a separate patch cluster for each Solaris platform. A patch cluster is not available for the Linux platform.

  Use a patch cluster if your upgrade plan calls for upgrading several shared components. Even if you are not upgrading all the shared components covered by the patch cluster, the patch cluster might be the most efficient way to upgrade the shared components specified in your upgrade plan.

  For information on downloading and applying a patch cluster refer to "Upgrading Shared Components with Patch Clusters" on page 61.

- **Replacement of Packages.** Some shared components can only be upgraded by replacing existing packages on your system with newer versions of the packages. The newer versions of shared component packages are available with your Java ES Release 4 distribution.

  Because patching technology is not available to upgrade Java Enterprise System on the Linux platform, you typically upgrade Linux shared components by replacement of RPM packages. However, some shared components deliver RPM packages as patches.

  Table 2-1 on page 56 shows the shared components that use replacement of packages when upgrading to Java ES Release 4. "Upgrading Components by Replacing Packages" on page 66 provides details on replacement of packages.

General Upgrade Procedure

The general steps you take to upgrade shared components are discussed below:

1. From your upgrade plan, determine your upgrade path and the shared components you wish to upgrade.

   Review the earlier sections in this overview for information on developing an upgrade plan. You can also refer to "Upgrade Planning" on page 33 for additional information.

2. Determine the upgrade technologies available to upgrade the shared components specified in your upgrade plan.

   You can find this information in the following sections:

   - "Upgrading Release 2 Shared Components" on page 57
   - "Upgrading Release 3 Shared Components" on page 55

3. Depending on your specific needs, follow the procedures in the following sections:

   - "Upgrading Shared Components by Applying Individual Patches" on page 59
   - "Upgrading Shared Components with Patch Clusters" on page 61
   - "Upgrading Components by Replacing Packages" on page 66
   - "Components Requiring Special Upgrade Procedures" on page 75

Upgrading Release 3 Shared Components

Table 2-1 on page 56 shows the upgrade technologies to use when upgrading shared components from Java ES Release 3. Please note the following:

- **Solaris platform.** A number of different upgrade options are available:
  - In many cases, you can apply specific patches to upgrade shared components to Java ES Release 4.
  - Applying a platform-specific patch cluster is typically the most efficient way to upgrade shared components from Release 3. A patch cluster bundles all the patches available to upgrade all shared components to Release 4. For more information, refer to "Upgrading Shared Components with Patch Clusters" on page 61.
  - In some cases, you have to replace Release 3 packages with Release 4 packages provided with the Java ES distribution.
  - For shared components that have not changed since Release 3, no upgrade is necessary.
- **Linux platform.** Shared components must be upgraded by installing or replacing RPM packages. Where a patch ID is listed in Table 2-1, a patch has been provided that bundles the RPM packages needed to upgrade the component, making it easy to download and install the packages. Special instructions are provided with the patch. Additionally, you can use the patch ID to track modifications made to the shared component. Patch clusters are not available for the Linux platform.

NOTE In Table 2-1, the trailing two digits in the patch ID specify the revision number for the patch. A higher revision number indicates a newer version.

Table 2-1 lists the minimum revision required for upgrade. If newer revisions of the patch become available, you should apply those revisions instead of the ones listed in the table.

The full names of shared components listed in Table 2-2 are provided in "Release 4 Shared Components" on page 27.

Table 2-1 Upgrade Technologies to Upgrade Shared Components from Java ES Release 3

Shared Component	Solaris 8 SPARC	Solaris 9 SPARC	Solaris 10 SPARC	Solaris 9 x86	Solaris 10 x86	Linux
ANT	Replace packages					
ACL	Replace packages					
BDB	Replace packages					
CAC	Replace packages					
ICU	116103-08	114677-10	119810-01	114678-10	119811-01	Replace Packages
IM-SDK	118789-09			118790-09		118791-10
J2SE™	Install J2SE 5.0 as described in “Upgrading J2SE for Java ES Release 4” on page 82.					
JAF	Unchanged since Release 3. No upgrade necessary					
JATO	Replace packages (Optional, see “Upgrading JATO” on page 80)					
JavaHelp™	Unchanged since Release 3. No upgrade necessary					Replace Packages
JavaMail ™	Unchanged since Release 3. No upgrade necessary					
JAX-Related Components JAXB JAXP JAXR JAX-RPC SAAJ WSCL	Apply platform-specific patch cluster See “Patch Cluster Procedures” on page 62.					119190-03
JCAPI	Unchanged since Release 3. No upgrade necessary					
JDMK	119044-01					119046-01
JSS	119209-05	119211-05	119213-06	119212-05	119214-06	Replace packages
KTSE	Unchanged since Release 3. No upgrade necessary					
LDAP C SDK	116837-02			116838-02		118353-02
LDAP J SDK	119725-02					Replace packages
MA Core	119527-02			119528-02		119529-02
MFWK	119803-02			119804-02		Replace packages

Table 2-1 Upgrade Technologies to Upgrade Shared Components from Java ES Release 3 *(Continued)*

Shared Component	Solaris 8 SPARC	Solaris 9 SPARC	Solaris 10 SPARC	Solaris 9 x86	Solaris 10 x86	Linux
NSPR	119209-05	119211-05	119213-05	119212-05	119214-05	Replace packages
NSS	119209-05	119211-05	119213-05	119212-05	119214-05	Replace packages
SASL	115328-02	115342-02	119345-01	115343-02	119346-01	Replace packages
SEDC	Refer to "Upgrading Sun Explorer Data Collector" on page 82					
SJWC	Replace packages					

Upgrading Release 2 Shared Components

Table 2-2 below shows the upgrade technologies to use when upgrading shared components from Java ES Release 2. Please note the following:

- **Solaris platform.** In most cases, you apply a platform-specific patch cluster to upgrade the shared components. In other cases, you either install or replace packages using Release 4 packages provided with the Java ES distribution.
- **Linux platform.** Shared components must be upgraded by installing or replacing RPM packages. Where a patch ID is listed in Table 2-2, a patch has been provided that bundles the RPM packages needed to upgrade the component, making it easy to download and install the packages. Special instructions are provided with the patch. Additionally, you can use the patch ID to track modifications made to the shared component. Patch clusters are not available for the Linux platform.

NOTE In Table 2-2, the trailing two digits in the patch ID specify the revision number for the patch. A higher revision number indicates a newer version.

Table 2-2 lists the minimum revision required for upgrade. If newer revisions of the patch become available, you should apply those revisions instead of the ones listed in the table.

The full names of shared components listed in Table 2-2 are provided in "Release 4 Shared Components" on page 27.

Table 2-2 Upgrade Technologies to Upgrade Shared Components from Java ES Release 2

Shared Component	Solaris 8 SPARC	Solaris 9 SPARC	Solaris 10 SPARC	Solaris 9 x86	Solaris 10 x86	Linux
ANT	Replace packages					
ACL	Replace packages					
BDB	Replace packages					
CAC	Install packages					
ICU	Apply platform-specific patch cluster					Replace Packages
IM-SDK	Apply platform-specific patch cluster					118791-09
J2SE™	Install J2SE 5.0 as described in "Upgrading J2SE for Java ES Release 4" on page 82.					
JAF	Apply platform-specific patch cluster					Install packages
JATO	Apply platform-specific patch cluster					Install packages
JavaHelp™	Replace packages					
JavaMail ™	Apply platform-specific patch cluster					Install package
JAX-Related Components JAXB JAXP JAXR JAX-RPC SAAJ WSCL	Apply platform-specific patch cluster					119190-03
JCAPI	Apply platform-specific patch cluster					118613-01
JDMK	Apply platform-specific patch cluster					119046-01
JSS	Apply platform-specific patch cluster					Install packages
KTSE	Apply platform-specific patch cluster					Replace Packages
LDAP C SDK	Apply platform-specific patch cluster					118353-02
LDAP J SDK	Apply platform-specific patch cluster					Install packages
MA Core	Apply platform-specific patch cluster					119529-02

Table 2-2 Upgrade Technologies to Upgrade Shared Components from Java ES Release 2 *(Continued)*

Shared Component	Solaris 8 SPARC	Solaris 9 SPARC	Solaris 10 SPARC	Solaris 9 x86	Solaris 10 x86	Linux
NSPR	Apply platform-specific patch cluster					Replace packages
NSS	Apply platform-specific patch cluster					Replace packages
SASL	Apply platform-specific patch cluster					Replace packages
SEDC	Refer to "Upgrading Sun Explorer Data Collector" on page 82					
SJWC	Install packages					

Upgrading Shared Components by Applying Individual Patches

On Solaris platforms, many packages for shared components can be upgraded from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4) by application of patches that modify the installed packages.

NOTE When upgrading from Java ES 2004Q2 (Release 2), you should not apply individual patches, but instead apply a patch cluster, as described in "Upgrading Shared Components with Patch Clusters" on page 61.

The advantage of patch technology over replacement of packages is that revisions applied by a patch can later be backed out, if needed. Typically, the size of a patch is smaller than the size of an updated package, so it is easier to download and install. Patches are generally more current than the latest available package. The most current revision of a patch is readily available for download from SunSolve, as described in this section.

Table 2-1 on page 56 shows the patch IDs for all shared component patches used to upgrade to Java ES Release 4 from Java ES Release 3.

If your upgrade plan calls for upgrading several shared components on a Solaris platform, using a patch cluster might be the most efficient way to perform the upgrade. A patch cluster contains all the patches available to upgrade shared components that use patch technology. Refer to "Upgrading Shared Components with Patch Clusters" on page 61 for more information on patch clusters.

If your upgrade plan calls for upgrading only a few shared components, you probably want to apply individual patches as described in the following procedure. You can later back out patches you apply, as described following the procedure.

NOTE Before making any changes to your system, it is advisable to first back up your system.

Patch Upgrades to Java ES 2005Q4

You can upgrade shared components using the individual patches shown in Table 2-1 on page 56. Use the following procedure.

1. Obtain the desired patch as indicated in Table 2-1.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Before applying the patch, read any special instructions in the README supplied with the patch.

4. Apply the patch using the `patchadd` command, as indicated in the following example, which applies the patch to the Mobile Access core shared component:

   ```
   patchadd 119527-02
   ```

   For information on the `patchadd` command, refer to the `patchadd(1M)` man page.

Rollback of Patch Upgrades

You can roll back an individual patch upgrade using the following procedure:

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Back out the patch using the `patchrm` command, as indicated in the following example, which backs out the patch to the Mobile Access core shared component:

   ```
   patchrm 119527-02
   ```

   For information on the `patchrm` command, refer to the `patchrm(1M)` man page.

Upgrading Shared Components with Patch Clusters

A patch cluster provides a convenient way to upgrade shared components to Java ES 2005Q4 (Release 4). For each Solaris platform there is a patch cluster that contains all the patches available to upgrade shared components that use patch technology. Applying a shared component patch cluster applies all the patches contained in the patch cluster.

The patch cluster you use does not depend on your upgrade path. You use the same platform-specific patch cluster when upgrading from Java ES Release 2 as you do when upgrading from Java ES Release 3.

The following patch clusters are available:

```
Java ES Component Patch Solaris 10 SPARC
Java ES Component Patch Solaris 10 x86
Java ES Component Patch Solaris 9 SPARC
Java ES Component Patch Solaris 9 x86
Java ES Component Patch Solaris 8 SPARC
```

You download a patch cluster from SunSolve at the following location:

http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

NOTE A patch cluster for the Linux platform is not available because only a few shared components provide patches to upgrade to Java ES Release 4 on the Linux platform.

Patch Cluster Contents

The patches bundled in a patch cluster vary according to each Solaris platform. As new patches become available, the contents of the patch cluster are updated. The `Cluster_readme` file provided with the patch cluster lists the patches it contains.

When you download a patch cluster, you get a platform-specific file in ZIP format. Extract the contents of the file to a directory from which you apply the patch cluster. The top level directory of the extracted contents includes the following files:

- `Cluster_readme`

  Provides information about applying the patch cluster

- `install_cluster` script

  Run this script to apply the patch cluster

- `copyright`

  Copyright notice for the patch cluster and documentation

The extracted contents also include directories for each patch contained in the patch cluster. These patch directories include README files applicable to each patch.

Patch Cluster Procedures

If your upgrade plan calls for upgrading only a few shared components, a patch cluster might not be the most efficient way to upgrade to Java ES Release 4. You might want to consider applying individual patches, as described in "Upgrading Shared Components by Applying Individual Patches" on page 59.

If your upgrade plan calls for upgrading several shared components on a Solaris platform, you probably want to upgrade the components using a patch cluster.

Keep in mind that the `install_cluster` script attempts to apply all patches in the patch cluster. The script upgrades shared components that are installed on the computer on which you run the script. It is normal for the patch cluster script to fail when it attempts to apply patches to shared components that are not on your computer or attempts to apply patches that are not needed.

During execution of the `install_cluster` script the script displays its progress. Detailed information is also available in a log file.

The following procedures show how to apply a shared component patch cluster.

NOTE You cannot roll back a patch cluster. Instead, you must keep track of all patches applied by the patch cluster script and roll back each patch individually (see "Rollback of Patch Upgrades" on page 61). It is advisable to back up your system before applying a patch cluster.

Upgrades from Solaris 8 and Solaris 9

1. Log in as root or become superuser.

   ```
   su -
   ```

2. If the following packages are present, remove them:

   ```
   SUNWjato
   SUNWjaxb
   SUNWjaxp
   SUNWjaf
   SUNWjmail
   SUNWxrgrt
   SUNwxrpcrt
   SUNWxsrt
   ```

   These selected packages might be present from JATO, Java Activation Framework (JAF), or the JAX family of shared components that were not part of a Java ES installation, or might be versions that cannot be upgraded using a patch cluster. These packages must be removed to ensure that the patch cluster script successfully upgrades to the newer versions of these packages.

   You can remove the packages by running the following command:

   ```
   pkgrm SUNWjato SUNWjaxb SUNWjaxp SUNWjaf SUNWjmail SUNWxrgrt    \
   SUNWxrpcrt  SUNWxsrt
   ```

3. Obtain the appropriate patch cluster for your Solaris platform from SunSolve at the following location:

   http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

4. Extract the contents of the platform-specific ZIP file into a location from which you can run the installation script.

5. Read the README, which contains important instructions and other information about the patch.

   The README contains a section “Save and Backout Options” that provides information on how to apply the patch cluster if you might later want to back out the changes.

6. Run the `install_cluster` script which installs the patches bundled in the patch cluster.

Upgrades on Solaris 10 (from Java ES Release 3 Only)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Determine the versions of the following packages, which might be present on your system:

   ```
   SUNWjaxp
   SUNWxrgrt
   SUNWxrpcrt
   SUNWxsrt
   ```

   a. Use the following command to determine the versions of the packages:

      ```
      pkgparam -v <package> | grep VERSION
      ```

   b. Compare the versions with the following versions:

      <SPARC> VERSION=7.0,REV=2003.05.07.00.23
      <x86> VERSION=7.0,REV=2003.10.10.14.34

      If the package versions *do not* match the version listed for your platform, or the packages are not installed on your computer, skip ahead to Step 3 on page 63.

      If the package versions *do* match the version listed for your platform, proceed to Step c below.

c. Determine if your Java ES distribution contains these packages:

A full Java ES distribution contains these packages. If you have a subset distribution, navigate to the directory appropriate for your platform to see if the packages are present:

```
Solaris_sparc/Product/shared_components/Packages/
Solaris_x86/Product/shared_components/Packages/
```

If your subset distribution *does not* contain these packages, then upgrading these packages is not necessary. Leave these packages installed on your computer and proceed to Step 3 on page 63.

If your distribution *does* contains these packages, proceed to Step d below.

d. Remove the installed packages from your system by running the following command:

```
pkgrm SUNWjaxp SUNWxrgrt SUNWxrpcrt SUNWxsrt
```

e. Install the corresponding packages from your distribution by issuing the following command:

```
pkgadd -d . SUNWjaxp SUNWxrgrt SUNWxrpcrt SUNWxsrt
```

3. Obtain the appropriate patch cluster for your Solaris platform from SunSolve at the following location:

`http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

4. Extract the contents of the platform-specific ZIP file into a location from which you can run the installation script.

5. Read the README, which contains important instructions and other information about the patch.

The README contains a section “Save and Backout Options” that provides information on how to apply the patch cluster if you might later want to back out the changes.

6. Run the `install_cluster` script which installs the patches bundled in the patch cluster.

Upgrading Components by Replacing Packages

Many shared components do not use patch technology to upgrade to Java ES 2005Q4 (Release 4). Instead, you upgrade these components by installing new packages or replacing existing packages with newer versions of the packages. In a few cases, you run special procedures to properly upgrade a shared component.

- For upgrades from Java ES 2004Q2 (Release 2), Table 2-2 on page 58 shows which shared components use package replacement to upgrade to Java ES Release 4.
- For upgrades from Java ES 2005Q1 (Release 3), Table 2-1 on page 56 shows which shared components use package replacement to upgrade to Java ES Release 4.
- Table 2-3 on page 69 provides links to any special procedures that might be required for Solaris platforms.
- Table 2-4 on page 73 provides links to any special procedures that might be required for the Linux platform.

Upgrade Strategy for Replacement of Packages

The number of shared components you upgrade depends on your upgrade plan and the Java ES components installed on your computer. For upgrades from Java ES Release 2 you must upgrade all shared components.

For upgrades from Java ES Release 3, you might be either upgrading all components or doing a selective upgrade of product components to Release 4. While you can choose to upgrade only those Java ES shared components needed to support the product components you select to upgrade, it is advisable to upgrade all shared components on your computer. For more information, refer to “Shared Component Upgrade Guidelines” on page 42.

NOTE The sequence of upgrading shared components is determined by component interdependencies, which should be reflected in your upgrade plan.

However, if you plan to upgrade J2SE to J2SE Release 5.0, you should upgrade this shared component first. J2SE is the base component for all Java ES components. Refer to “Upgrading J2SE for Java ES Release 4” on page 82 for more information.

Upgrade Path

The procedures for upgrading shared components by replacement of packages generally do not depend on your upgrade path. You follow the same procedures when upgrading from Java ES Release 2 as you do when upgrading from Java ES Release 3.

However, the sections "Packages for Solaris Platforms" on page 68 and "Packages for the Linux Platform" on page 72 contain tables that specify the few instances where the upgrade path determines the procedures you use.

Package Versions

When replacing packages, you should only replace packages with newer versions of those packages. The sections "Packages for Solaris Platforms" and "Packages for the Linux Platform" provide information on how to compare package versions before upgrading.

Package Locations

The packages for upgrading most shared components are provided with your Java ES distribution under one of the following directories, depending on your platform:

```
Solaris_sparc/Product/shared_components/Packages/
Solaris_x86/Product/shared_components/Packages/
Linux_x86/Product/shared_component/Packages/
```

Some packages on Solaris platforms have versions specific to the operating system. These packages are found under the following directories:

```
<Solaris_ARCH>/Product/shared_components/Solaris_10/Packages/
<Solaris_ARCH>/Product/shared_components/Solaris_8/Packages/
<Solaris_ARCH>/Product/shared_components/Solaris_9/Packages/
```

Localized versions of shared component packages can be found in the following directory:

```
<PLATFORM_ARCH>/Product/shared_components/Packages/locale/
```

Procedures for Replacement of Packages

With few exceptions, packages can be replaced following general procedures. These procedures are detailed in the following sections:

- "Upgrading Packages on Solaris Platforms" on page 70
- "Upgrading Packages on Linux Platforms" on page 74

Special Instructions

Some packages have special instructions for preserving configuration information or other data. Links to special instructions are contained in the package tables listed in the sections “Packages for Solaris Platforms” on page 68 and “Packages for the Linux Platform” on page 72.

All special instructions are detailed in the section “Components Requiring Special Upgrade Procedures” on page 75.

Packages for Solaris Platforms

Table 2-3 below lists the Solaris packages for shared components that are upgraded by replacement of packages. The table also lists the versions of the packages available in Java ES Release 4 and a link to any special instructions for upgrading the component.

For each shared component, the packages are listed in the sequence you would install them.

You should only replace packages with newer versions. Before you replace a package, compare the version of the package on your system with the version of the package you intend to use to replace it.

To determine the version of an installed package use the `pkgparam` command with the verbose (`-v`) option. The output of this command provides the package version, its revision, and the `SUNW_PRODVERS` version. For example:

```
pkgparam -v SUNWjato | grep VERSION
VERSION='2.1.4,REV=2004.11.10.16.05'

pkgparam -v SUNWjato | grep SUNW_PRODVERS
SUNW_PRODVERS='2.1.4'
```

NOTE The versioning system for different packages varies, but generally a higher number indicates a newer version of the package.

“Release 4 Shared Components” on page 27 specifies the full name of shared components listed in Table 2-3.

Table 2-3 Package Versions for Upgrading Shared Components on Solaris Platforms

Shared Component	Packages	Version Rev	SUNW_PRODVERS	Special Instructions
ANT	SUNWant	11.11.0 2005.04.06.16.31.04	1.6.2	No
ACL	SUNWaclg	8.1 2005.05.31.17.01.28	1.0.3	No
BDB	SUNWbdb SUNWbdbj	4.2.52 1.0.3	4.2.52, REV=1.0.3	No
CAC	SUNWcacaocfg SUNWcacao	1.1 15	1.1	Install or replace these packages according to the special instructions in "Upgrading Common Agent Container" on page 76.
IM-SDK	SUNWiimdv	6.1 2004.04.16.16.01.40	7.0	No
J2SE™ SPARC 32-bit	SUNWj5rt SUNWj5dev SUNWj5cfg SUNWj5man SUNWj5dmo SUNWj5jmp	1.5.0 2004.12.07.00.07	1.5.0_04/ 1.5.0_04-b05	Install J2SE™ platform 5.0 (Java 2 Platform, Standard Edition) as described in "Upgrading J2SE for Java ES Release 4" on page 82.
J2SE™ SPARC 64-bit	SUNWj5rtx SUNWj5dvx SUNWj5dmx	1.5.0 2004.12.06.22.09	1.5.0_04/ 1.5.0_04-b05	
JATO	SUNWjato SUNWjatodoc SUNWjatodmo	2.1.5 2005.04.06.08.07	2.1.5	Replace JATO packages according to instructions described in "Upgrading JATO" on page 80.
JavaHelp™	SUNWjhrt SUNWjhdev	2.0 2004.11.23	2.0/FCS	Replace these packages when upgrading from Java ES Release 2. These packages are unchanged from Java ES Release 3.

Table 2-3 Package Versions for Upgrading Shared Components on Solaris Platforms *(Continued)*

Shared Component	Packages	Version Rev	SUNW_PRODVERS	Special Instructions
SEDC	SUNWexplj SUNWexplu SUNWexplo	4.3.1 2004.06.25.07.21	4.3.1 GA	Upgrade Sun Explorer according to instructions in "Upgrading Sun Explorer Data Collector" on page 82.
SJWC	SUNWmctag SUNWmconr SUNWmcon SUNWmcos SUNWmcosx	2.2.4 2005.05.09.14.06	SNAG Development	Refer to "Upgrading Sun Java Web Console" on page 81 for information on upgrading SJWC. For the Solaris 10 platform, an upgrade of Sun Java Web Console for Java Enterprise System is not necessary.

Upgrading Packages on Solaris Platforms

The following procedure shows the general instructions for upgrading packages on Solaris platforms.

A few shared components require special instructions in addition to these general instructions. Table 2-3 on page 69 provides a link to special instructions for components that require them.

NOTE Before making any changes to your system, it is advisable to first back up your system.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Check Table 2-3 on page 69 for special instructions that might apply to the component you are upgrading.

   Follow any special instructions before upgrading the package. If no special instructions are indicated, proceed to the next step.

3. Navigate to the location of the packages in your Java ES Release 4 distribution.

   The packages are found under one of the following directories, depending on your platform:

   ```
   Solaris_sparc/Product/shared_components/Packages/
   Solaris_x86/Product/shared_components/Packages/
   ```

   NOTE If you have a subset distribution of Java Enterprise System, this subset distribution contains all the necessary shared components at the location indicated above.

4. Remove the current versions of the packages that you are upgrading by using the `pkgrm` command.

   For example, to remove packages for JATO:

   ```
   pkgrm SUNWjatodmo SUNWjatodoc SUNWjato
   ```

   For detailed information on removing packages, refer to the `pkgrm(1m)` man page.

5. Install the packages from your distribution using the `pkgadd` command.

   For example, to install packages for JATO:

   ```
   pkgadd -d . SUNWjato SUNWjatodoc SUNWjatodmo
   ```

   For detailed information on installing packages, refer to the `pkgadd(1M)` man page.

6. Verify that the package is correctly installed using the `pkgparam` and `pkginfo` commands.

   Use `pkgparam` with the -v option to verify the version. `pkginfo` provides provides additional information about the package.

   For detailed information on these commands refer to the `pkgparam(1)` and `pkginfo(1)` man pages.

Packages for the Linux Platform

Table 2-4 below lists the Linux RPM packages used to upgrade shared components. The table also provides a link to any special instructions for upgrading the component. The version of an RPM package is embedded in the package name.

You should only replace packages with newer versions. Before you replace a package, compare the version of the package on your system with the version of the package you intend to use to replace it. A higher version number indicates a newer version of the package.

The RPM naming conventions provides information about the version of the packages. Different shared components embed the versioning information differently. Generally, the number embedded in the file name provides the package version number and revision number.

To determine the version of an installed package use the `rpm` query command with the info (`-i`) option. This command displays package information, including name, version, and description. For example:

```
rpm -qi SUNWjato-2.1.5.i386.rpm
```

NOTE The versioning system for different packages varies, but generally a higher number indicates a newer version of the package.

"Release 4 Shared Components" on page 27 specifies the full name of shared components listed in Table 2-4.

Table 2-4 Packages for Upgrading Shared Components on the Linux Platform

Shared Component	Packages	Special Instructions
ANT	sun-ant-1.6.2-1.rpm	No
ACL	sun-aclg-1.0.3-1.i386.rpm	No
BDB	sun-berkeleydatabase-core-4.2.52-4.4.i386.rpm sun-berkeleydatabase-java-4.2.52-4.4.i386.rpm	No
CAC	sun-cacao-1.1-15.i386.rpm sun-cacaocfg-1.1-15.i386.rpm sun-cacao-man-1.1-15.i386.rpm	Install or replace these packages according to the special instructions in "Upgrading Common Agent Container" on page 76.
ICU	sun-icu-3.2-1.i386.rpm	No
IM-SDK	sun-im-dev-6.2.9.13.i386.rpm	No
J2SE™	jdk-1_5_0_04-linux-i586.rpm	Install J2SE™ platform 5.0 (Java 2 Platform, Standard Edition) as described in "Upgrading J2SE on the Linux Platform" on page 88.
JATO	SUNWjato-2.1.5.i386.rpm SUNWjatodmo-2.1.5.i386.rpm SUNWjatodoc-2.1.5.i386.rpm	Install or replace JATO packages according to instructions described in "Upgrading JATO" on page 80.
JavaHelp™	sun-javahelp-2.0-fcs.i586.rpm	Replace these packages according to instructions described in "Upgrading JavaHelp on the Linux Platform" on page 81.
JavaMail	sun-javamail-1.3.2-34.i386.rpm	Install this package only if upgrading from Java ES Release 2. This packages is unchanged from Java ES Release 3.
JSS	sun-jss-4.1-4.i386.rpm	If you are upgrading security shared components in preparation for upgrading Sun Cluster software, refer to "Upgrading Security Components (NSS, NSPR, JSS)" on page 76.
KTSE	sun-ktsearch-1.3-2.noarch.rpm	Install this package if upgrading from Java ES Release 2. This package is unchanged from Java ES Release 3.
LDAP J SDK	sun-ljdk-4.18-4.i386.rpm	Install this package if upgrading from Java ES Release 2. Replace this package if upgrading from Java ES Release 3.
MFWK	sun-mfwk-cfg-1.0.1-1.i386.rpm sun-mfwk-dev-1.0.1-1.i386.rpm sun-mfwk-man-1.0.1-1.i386.rpm	Install these packages if upgrading from Java ES Release 2. Replace these packages if upgrading from Java ES Release 3.

Table 2-4 Packages for Upgrading Shared Components on the Linux Platform *(Continued)*

Shared Component	Packages	Special Instructions
NSPR	sun-nspr-4.5.2-4.i386.rpm sun-nspr-devel-4.5.2-4.i386.rpm	If you are upgrading security shared components in preparation for upgrading Sun Cluster software, refer to "Upgrading Security Components (NSS, NSPR, JSS)" on page 76.
NSS	sun-nss-3.10.1-1.i386.rpm	If you are upgrading security shared components in preparation for upgrading Sun Cluster software, refer to "Upgrading Security Components (NSS, NSPR, JSS)" on page 76.
SASL	sun-sasl-2.18-1.i386.rpm	No
SJWC	SUNWmcon-2.2.4-1.i386.rpm SUNWmconr-2.2.4-1.i386.rpm SUNWmcos-2.2.4-1.i386.rpm SUNWmcosx-2.2.4-1.i386.rpm SUNWmctag-2.2.4-1.i386.rpm	Refer to "Upgrading Sun Java Web Console" on page 81 for information on upgrading SJWC. For the Solaris 10 platform, an upgrade of Sun Java Web Console for Java Enterprise System is not necessary.

Upgrading Packages on Linux Platforms

The following procedure shows the general instructions for updating packages on the Linux platform.

A few shared components require special instructions in addition to these general instructions. Table 2-4 on page 73 provides a link to special instructions for components that require them.

NOTE Before making any changes to your system, it is advisable to first backup your system.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. For each component, check Table 2-4 on page 73 for special instructions on updating these packages.

   Follow any special instructions before updating the package. If there are no special instructions proceed to the next step.

3. Locate the necessary RPM packages in the Java ES Release 4 distribution. The RPMs are found in the following directory:

   `Linux_x86/Product/shared_components/Packages/`

4. Update the appropriate RPMs for your situation using the `rpm -U` command.

   For example:

   ```
   rpm -Uvh SUNWjato-2.1.5.i386.rpm SUNWjatodmo-2.1.5.i386.rpm  \
   SUNWjatodoc-2.1.5.i386.rpm
   ```

   The Linux `rpm` utility correctly sequences the packages for installation.

   For detailed information on updating packages, refer to the `rpm` man page.

Components Requiring Special Upgrade Procedures

This section provides instructions for upgrading shared components that require special procedures. It contains special procedures for the following shared components:

Upgrading Security Components (NSS, NSPR, JSS)

If you are upgrading the security shared components (NSS, NSPR, and JSS) in preparation for upgrading Sun Cluster software, follow the procedures for upgrading dependency software in "Upgrading Sun Cluster Software" in *Sun Cluster Software Installation Guide for Solaris OS*, which is available at:

http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view

However, install the packages for the security components from the Java ES Release 4 distribution rather than from the Sun Cluster 1 of 2 CD-ROM or the Sun Cluster 2 of 2 CD-ROM.

Upgrading Common Agent Container

If you are upgrading the common agent container shared components in preparation for upgrading Sun Cluster software, follow the procedures for upgrading dependency software in "Upgrading Sun Cluster Software" in *Sun Cluster Software Installation Guide for Solaris OS*, which is available at:

http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view

However, install the packages for the common agent container components from the Java ES Release 4)distribution rather than from the Sun Cluster 1 of 2 CD-ROM or the Sun Cluster 2 of 2 CD-ROM.

The following sections describes the procedure for upgrading common agent container packages for standalone systems.

Upgrading from Java ES Release 2 on Solaris Platforms

Install common agent packages according to the procedures in "Upgrading Packages on Solaris Platforms" on page 70.

Upgrading from Java ES Release 2 on the Linux Platform

follow the procedures in "Upgrading Packages on Linux Platforms" on page 74. However, you must also apply Patch 120677-01, which provides an updated Linux RPM package. This patch is available from SunSolve at the following location:

http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

Upgrading from Java ES Release 3

Follow the special procedure below.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Make sure you update the shared components upon which common agent container depends before updating the common agent container shared component.

   Shared components upon which common agent container depends are Java 2 Platform Standard Edition (J2SE), Java Dynamic Management Kit Runtime (JDMK), Network Security Services (NSS), and Netscape Portable Runtime (NSPR). Refer to Table 2-1 on page 56 for information on upgrading these shared components.

   If you update J2SE to Version 5, then you need to update dependencies, as indicated in Step 8 on page 79.

3. If the current installation uses custom configuration settings (for example, which ports are used) capture the configuration settings using the following commands:

   On Solaris platforms:

   ```
   /opt/SUNWcacao/bin/cacaoadm list-params
   ```

   On the Linux platform:

   ```
   /opt/sun/cacao/bin/cacaoadm list-params
   ```

   The output will be similar to the following:

   ```
   java-flags=-Xms4M -Xmx64M
   jmxmp-connector-port=10162
   snmp-adaptor-port=10161
   snmp-adaptor-trap-port=10162
   commandstream-adaptor-port=10163
   retries=4
   ```

   The example above lists the default values. Note any nondefault settings for use in Step 7 on page 79.

4. Stop common agent container processes using the following commands:

   On Solaris platforms:

   ```
   /opt/SUNWcacao/bin/cacaoadm stop
   echo $?
   ```

   If the exit code is not 0, force the stop:

   ```
   /opt/SUNWcacao/bin/cacaoadm stop -f
   ```

   On the Linux platform:

   ```
   /opt/sun/cacao/bin/cacaoadm stop
   echo $?
   ```

   If the exit code is not 0, force the stop:

   ```
   /opt/sun/cacao/bin/cacaoadm stop -f
   ```

5. You can now upgrade the following common agent container packages, as indicated below:

   For Solaris platforms, follow the procedure in "Upgrading Packages on Solaris Platforms" on page 70 to upgrade the following packages:

   ```
   SUNWcacaocfg
   SUNWcacao
   ```

   For the Linux platforms, follow the basic procedure in "Upgrading Packages on Linux Platforms" on page 74 to upgrade the following packages. However, note the significant changes to the procedure below:

   ```
   sun-cacaocfg-1.1-15.i386.rpm
   sun-cacao-man-1.1-15.i386.rpm
   ```

   `rpm -U` is not supported by common agent container 1.1. To upgrade on Linux platforms, use the following commands:

   ```
   rpm -e sun-cacao-man-1.0
   rpm -e sun-cacao-1.0
   rpm -e sun-cacao-config-1.0
   rpm -i sun-cacao-config-1.1
   rpm -i sun-cacao-1.1
   rpm -i sun-cacao-man-1.1
   ```

6. On the Linux platform only, apply Patch 120677-01 before proceeding.

   This patch provides an updated Linux RPM package which must be installed before proceeding. Follow the instructions provided with the patch. Patch 120677-01 is available from SunSolve Patch Access at the following location:

   http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

7. Apply any custom configuration settings previously captured in Step 3 on page 77.

   On Solaris platforms, use the following commands:

```
/opt/SUNWcacao/bin/cacaoadm set-param java-flags=<Value>
/opt/SUNWcacao/bin/cacaoadm set-param jmxmp-connector-port=<Value>
/opt/SUNWcacao/bin/cacaoadm set-param snmp-adaptor-port=<Value>
/opt/SUNWcacao/bin/cacaoadm set-param snmp-adaptor-trap-port=<Value>
/opt/SUNWcacao/bin/cacaoadm set-param commandstream-adaptor-port=<Value>
/opt/SUNWcacao/bin/cacaoadm set-param retries=<Value>
```

   On the Linux platform, use the following commands:

```
/opt/sun/cacao/bin/cacaoadm set-param java-flags=<Value>
/opt/sun/cacao/bin/cacaoadm set-param jmxmp-connector-port=<Value>
/opt/sun/cacao/bin/cacaoadm set-param snmp-adaptor-port=<Value>
/opt/sun/cacao/bin/cacaoadm set-param snmp-adaptor-trap-port=<Value>
/opt/sun/cacao/bin/cacaoadm set-param commandstream-adaptor-port=<Value>
/opt/sun/cacao/bin/cacaoadm set-param retries=<Value>
```

8. If you upgraded J2SE to J2SE Version 5, run the rebuild-dependencies utility:

   On Solaris platforms:

```
/opt/SUNWcacao/bin/cacaoadm rebuild-dependencies
```

   On the Linux platform:

```
/opt/sun/cacao/bin/cacaoadm rebuild-dependencies
```

   The output of this command will be:

```
Property updated: [java-home].
Property updated: [jdmk-home].
Property updated: [nss-lib-home].
Property updated: [nss-tools-home].
```

9. Restart common agent container services:

   ```
   cacaoadm start
   ```

10. Verify the upgrade of common agent container:

    ```
    cacaoadm status
    cacaoadm verify-configuration
    ```

Upgrading JATO

The version of JATO packages provided with Java ES Release 4 contains an update required by Sun Java Studio Enterprise. You might want the updated JATO packages if you are using Sun Java Studio Enterprise. Otherwise, the earlier version of JATO provided with Java ES Release 3 does not need to be upgraded.

Upgrading JATO from Java ES Release 3 on Solaris Platforms

Replace the base version of JATO installed with Java ES Release 3 with the JATO packages provided with your Java ES Release 4 distribution.

You only need this later version of JATO if you are using Sun Java Studio Enterprise and want the updates provided with this later version.

Upgrading JATO from Java ES Release 2 on Solaris Platforms

1. Apply the patch cluster for your platform, as described in "Patch Cluster Procedures" on page 62.

2. Replace the base version of JATO installed by the patch cluster script with the JATO packages provided with your Java ES Release 4 distribution.

   You only need this later version of JATO if you are using Sun Java Studio Enterprise and want the updates provided with this later version.

Upgrading JATO from Java ES Release 3 on the Linux Platform

Replace the base version of JATO installed with Java ES Release 3 with the JATO RPM packages provided with your Java ES Release 4 distribution.

You only need this later version of JATO if you are using Sun Java Studio Enterprise and want the updates provided with this later version.

Upgrading JATO from Java ES Release 2 on the Linux Platform

Install the JATO RPM packages provided with your Java ES Release 4 distribution.

Upgrading JavaHelp on the Linux Platform

This special procedure is for upgrading JavaHelp on the Linux platform only. Perform this special procedure when upgrading JavaHelp from both Java ES Release 2 and Java ES Release 3. In both scenarios, the JavaHelp must be replaced with the RPM provided with your Java ES Release 4 distribution.

When replacing the JavaHelp RPM package, do not use the `-U` (upgrade) option to the `rpm` utility. Instead use the `-e` (erase) option followed by the `-i` (install) option, as illustrated below:

```
rpm -e sun-javahelp-version.rpm
rpm -i sun-javahelp-2.0-fcs.i586.rpm
```

Upgrading Sun Java Web Console

For Solaris 8 and Solaris 9 platforms, if you are upgrading Sun Java Web Console in preparation for upgrading Sun Cluster software, follow the procedures for upgrading dependency software in "Upgrading Sun Cluster Software" in *Sun Cluster Software Installation Guide for Solaris OS*, which is available at:

`http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view`

However, install the packages for Sun Java Web Console from the Java ES Release 4 distribution rather than from the Sun Cluster 1 of 2 CD-ROM or the Sun Cluster 2 of 2 CD-ROM.

Packages for upgrading Sun Java Web Console are not in the standard location for shared component packages. Instead, look for the packages in the following directory of your Java ES distribution:

`<Architecture>/Product/shared_components/Packages/<OperatingSystem>/`

Where *Architecture* can be `Solaris_sparc` or `Solaris_x86` and *OperatingSystem* can be `Solaris_8` or `Solaris_9`.

There are no upgrade procedures for Sun Java Web Console on the Solaris 10 platform. Solaris 10 provides the Sun Java Web Console as part of the operating system.

Upgrading Sun Explorer Data Collector

Sun Cluster software is the only component for which Sun Explorer is needed.

To upgrade Sun Explorer software, follow the procedures for upgrading dependency software in "Upgrading Sun Cluster Software" in *Sun Cluster Software Installation Guide for Solaris OS*, which is available at:

http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view

However, install the packages for the security components from the Java ES Release 4 distribution rather than from the Sun Cluster 1 of 2 CD-ROM or the Sun Cluster 2 of 2 CD-ROM.

Upgrading J2SE for Java ES Release 4

Java ES Release 4 is certified for Java 2 Platform, Standard Edition (J2SE) Version 5.0 Update 4, identified here as J2SE 5.0 Update 4. (J2SE 5.0 is sometimes referred to as developer version 1.5.0). Except as noted below, Java ES Release 4 still supports J2SE 1.4.2 and J2SE 5.0 Update 1.

NOTE High Availability Session Store (HADB) distributed with Java ES Release 4 requires J2SE Release 5. If you plan to upgrade HADB, then you must also upgrade to J2SE 5.0.

For Java ES Release 4, it is recommended that you upgrade J2SE to Version 5.0 Update 4, but keep J2SE 1.4.2 installed.

Java Enterprise System does not use the default J2SE installed on your computer, but instead maintains a symbolic link to the supported version of J2SE. After upgrading J2SE you need to set the Java ES symbolic link so it points to the upgraded J2SE.

Nevertheless, you should maintain pointers to J2SE 1.4.2 for those services that require the earlier version. Consult the appropriate product component documentation for information on how to maintain symbolic links to the earlier versions of J2SE.

The following sections provide instructions for upgrading J2SE on Solaris and Linux platforms:

- "Upgrading J2SE on Solaris Platforms" on page 83
- "Upgrading J2SE on the Linux Platform" on page 88

Upgrading J2SE on Solaris Platforms

The procedures you use to upgrade J2SE depends on whether you are upgrading from J2SE 1.4 or an earlier than Update 4 version of J2SE 5.0.

You should therefore determine the version of J2SE currently used by your Java ES installation. The default versions are as follows:

- Java ES Release 2. The default is J2SE 1.4.2
- Java ES Release 3. The default is J2SE Version 5.0 Update 1

For various reasons, you might have upgraded J2SE from the default versions. To determine which version of J2SE your Java ES installation is using, run the following command:

```
/usr/jdk/entsys-j2se/bin/java -version
```

- Here is an example of the version string displayed for J2SE 1.4.2 Update 5:

  ```
  java version "1.4.2_05"
  ```

- Here is an example of the version string displayed for J2SE 5.0 Update 1:

  ```
  java version "1.5.0_01"
  ```

General Procedure for Solaris Platforms

The general procedure for upgrading J2SE on Solaris platforms, depend on which version of J2SE you are starting from:

- Follow this procedure if you are upgrading from J2SE 1.4

  a. Install J2SE 5.0 Update 4 or later (see "Installing J2SE 5.0 on Solaris Platforms" on page 84).

  b. Set the Java ES symbolic link to point to the newly installed J2SE (see "Setting the J2SE Symlink for Java ES on Solaris Platforms" on page 87).

NOTE You do not need to remove J2SE 1.4. Both versions can be installed on the same computer.

You may elect to continue running some Java ES services on the previous version of J2SE. Consult the appropriate component product administration guides to do so. For example, you can change the J2SE symlink used by Application Server to point to the previous J2SE version.

- Follow this procedure if you are upgrading from an earlier than Update 4 version of J2SE 5.0

  a. Remove the earlier version of J2SE 5.0.

  b. Install J2SE 5.0 Update 4 or later (see "Installing J2SE 5.0 on Solaris Platforms" on page 84).

  c. Set the Java ES symbolic link to point to the newly installed J2SE (see "Setting the J2SE Symlink for Java ES on Solaris Platforms" on page 87).

  Or

  a. Upgrade the current version of J2SE by applying patches (see "Upgrading J2SE 5.0 on Solaris Platforms by Applying Patches" on page 86).

  b. Set the Java ES symbolic link to point to the newly updated J2SE (see "Setting the J2SE Symlink for Java ES on Solaris Platforms" on page 87).

When upgrading J2SE you might want to shut down any services that depend on the currently installed J2SE before proceeding. This is to avoid any problems that might arise with services that are using the current J2SE. If you do not shut down the services that depend on J2SE, after installing J2SE and setting the Java ES symbolic link to it, you should reboot your system.

Installing J2SE 5.0 on Solaris Platforms

You can install J2SE 5.0 on Solaris platforms using the software available from either of the following sources:

- The Sun Developer Network
- The Java ES distribution

NOTE The procedures in this section install J2SE 5.0 in the default location. If you want to install J2SE in a non-default location, follow the instructions from the Sun Developer Network at:

`http://java.sun.com/j2se/1.5.0/install.html`

To install J2SE 5.0 from the Sun Developer Network:

1. Navigate to the following location in the Sun Developer Network to retrieve the current version of JDK 5.0 and the installation instructions:

   `http://java.sun.com/j2se/1.5.0/download.jsp`

2. Follow the instructions for installation available with the download.

   Before installing J2SE, you might want to stop services that depend on J2SE, as described in "General Procedure for Solaris Platforms" on page 83.

3. After installation is complete, proceed to the section "Setting the J2SE Symlink for Java ES on Solaris Platforms" on page 87

To install J2SE 5.0 from the Java ES distribution:

1. Log in as root or become superuser.

   ```
   su -
   ```

2. [Optional] Shut down Java ES services as described in "General Procedure for Solaris Platforms" on page 83.

3. If you have an earlier version of J2SE 5.0 installed, remove it as indicated below:

   Remove these packages:

   ```
   pkgrm SUNWj5rt SUNWj5dev SUNWj5cfg SUNWj5man SUNWj5dmo
   ```

   For computers with 64-bit processors, remove these additional packages:

   ```
   pkgrm SUNWj5rtx SUNWj5dvx SUNWj5dmx
   ```

4. Navigate to the location of the J2SE packages in the Java ES Release 4 distribution.

   The packages are found under one of the following directories, depending on your platform:

   ```
   Solaris_sparc/Product/shared_components/Packages/
   Solaris_x86/Product/shared_components/Packages/
   ```

NOTE If you have a subset distribution of Java Enterprise System, this subset distribution contains the necessary packages to install J2SE.

5. Install the J2SE packages using the pkgadd command:

   Install these packages:

   ```
   pkgadd -d . SUNWj5rt SUNWj5dev SUNWj5cfg SUNWj5man SUNWj5dmo
   ```

   For computers with 64-bit processors, install these additional packages:

   ```
   pkgadd -d . SUNWj5rtx SUNWj5dvx SUNWj5dmx
   ```

   This installs J2SE 5.0 Update 4 into /usr/jdk/jdk1.5.0_04. Version 5.0 does not automatically become the default Java platform on Solaris 8 or Solaris 9 (unless there was no default), but does become the default on Solaris 10.

NOTE On Solaris 8 and 9, it is possible to make J2SE 5.0 the default Java platform by modifying the /usr/java symbolic link to point to /usr/jdk/jdk1.5.0_04.

However, changing the symbolic link in this manner may cause problems for some earlier Java applications that have not been tested with J2SE 5.0. For more information, refer to J2SE 5.0 installation notes at:

http://java.sun.com/j2se/1.5.0/compatibility.html

6. [Optional] Install Japanese man pages.

   Use the pkgadd command to install the new Japanese man page package:

   ```
   pkgadd -d . SUNWj5jmp
   ```

7. Proceed to the following section, "Setting the J2SE Symlink for Java ES on Solaris Platforms" on page 87.

Upgrading J2SE 5.0 on Solaris Platforms by Applying Patches

The following procedure shows how to upgrade an installed version of J2SE 5.0 platform to the supported version.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Obtain the patch required for your Solaris platform, as indicated by the table below.

Platform	Patch
SPARC	118666-03 or higher
SPARC 64 bit	118667-03 or higher
X86	118668-03 or higher
x86 64 bit	118669-03 or higher

   The trailing two digits of the patch ID specify the revision number for the patch. A higher revision number indicates a newer version of the patch. Refer to the README file for each patch listed for special instructions.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

3. [Optional] Shut down Java ES services as described in "General Procedure for Solaris Platforms" on page 83.

4. Apply the patch using the `patchadd` command.

   For example, for Solaris SPARC platforms:

   ```
   patchadd 118666-03
   ```

5. Proceed to the following section, "Setting the J2SE Symlink for Java ES on Solaris Platforms."

Setting the J2SE Symlink for Java ES on Solaris Platforms

Java Enterprise System maintains a symbolic link that points to the supported version of J2SE platform. Java Enterprise System maintains this link to ensure that Java ES services can find the correct J2SE runtime to use.

If you upgraded to J2SE 5.0 Update 4 from J2SE 1.4.2, then you need to set the symbolic link so it points to the newly installed J2SE 5.0. If you upgraded to J2SE 5.0 Update 4 from an earlier version of J2SE 5.0, then you just need to verify that your Java ES installation is using the updated version.

The following procedure shows how to set the Java ES symbolic link to your upgraded J2SE installation.

1. Reset the /usr/jdk/entsys-j2se symbolic link to point to the newly installed or updated J2SE installation as indicated below:

   If you installed J2SE 5.0 Update 4 in the default location, reset the symbolic link as follows:

   ```
   rm /usr/jdk/entsys-j2se
   ln -s /usr/jdk/instances/jdk1.5.0 /usr/jdk/entsys-j2se
   ```

   If you installed J2SE 5.0 in a non-default location, replace the default path (`/usr/jdk/instances/jdk1.5.0`) with the path to your non-default location.

2. If you previously stopped services prior to upgrading or installing J2SE 5.0 Update 4, restart the services.

   If you did not stop services prior to upgrading or installing J2SE 5.0 you might want to reboot your system so services that depend on J2SE 5.0 use the new symbolic link.

Verifying the Upgrade of J2SE

The following command verifies the version of J2SE referenced by the J2SE symbolic link:

```
/usr/jdk/entsys-j2se/bin/java -version
```

The command returns a string containing the developer version number. For example, if you installed J2SE 5.0 Update 4, then this command returns the following string:

```
java version "1.5.0_04"
```

If the above command does not return the correct version, check that Java ES symbolic link to J2SE is set correctly, as described in "Setting the J2SE Symlink for Java ES on Solaris Platforms."

Upgrading J2SE on the Linux Platform

The procedure you use to upgrade J2SE on Linux does not depend on your upgrade path. Use the same procedure whether you are upgrading from J2SE 1.4 or an earlier version of J2SE 5.0. On the Linux platform, you can have multiple versions of J2SE 5.0.

The general procedure for upgrading J2SE on the Linux platform is as follows.

1. Install J2SE 5.0 Update 4 or later (see "Installing J2SE 5.0 on Linux Platforms" on page 89).

2. Set the Java ES symbolic link to point to the newly installed J2SE (see "Setting the J2SE Symlink for Java ES on the Linux Platform" on page 90).

NOTE Removal of earlier versions of J2SE is optional. If other services depend on earlier versions, you probably want to keep the earlier versions installed.

Installing J2SE 5.0 on Linux Platforms

You can install J2SE 5.0 on Solaris platforms using the software available from either of the following sources:

- The Sun Developer Network
- The Java ES distribution

NOTE The procedures in this section install J2SE 5.0 in the default location. If you want to install J2SE in a non-default location, follow the instructions from the Sun Developer Network at:

`http://java.sun.com/j2se/1.5.0/install.html`

To install J2SE 5.0 from the Sun Developer Network:

1. Navigate to the following location in the Sun Developer Network to retrieve the current version of JDK 5.0 and the installation instructions:

   `http://java.sun.com/j2se/1.5.0/download.jsp`

2. Follow the instructions for installation available with the download.

   Before installing J2SE, you might want to stop services that depend on J2SE, as described in "General Procedure for Solaris Platforms" on page 83.

3. After installation is complete, proceed to the section "Setting the J2SE Symlink for Java ES on the Linux Platform" on page 90

To install J2SE 5.0 from the Java ES distribution:

1. Log in as root or become superuser.

   ```
   su -
   ```

2. [Optional] Shut down Java ES services as described in "General Procedure for Solaris Platforms" on page 83.

3. Navigate to the following directory in your Java ES distribution, which contains the `jdk-1_5_0_04-linux-i586.rpm` file:

   `Linux_x86/Product/shared_components/Packages/`

   NOTE If you have a subset distribution of Java Enterprise System, this subset distribution contains the necessary packages to install J2SE.

4. Install the RPM package using the following command:

   ```
   rpm -Uvh jdk-1_5_0_04-linux-i586.rpm
   ```

   Removal of earlier versions of J2SE is optional. If other services depend on the earlier versions, you probably want to leave those versions installed.

5. Continue to the procedure for setting the Java ES symlink, which appears in the next section.

Setting the J2SE Symlink for Java ES on the Linux Platform

Java Enterprise System maintains a symbolic link that points to the supported version of J2SE platform. Java Enterprise System maintains this link to ensure that Java ES service can find the correct J2SE runtime to use.

The following procedure shows how to set the Java ES symbolic link.

1. Reset the `/usr/jdk/entsys-j2se` symbolic link to point to the newly installed or updated J2SE installation as indicated below:

   If you installed J2SE 5 Update 4 in the default location, reset the symbolic link as follows:

   ```
   rm /usr/jdk/entsys-j2se
   ln -s /usr/java/jdk1.5.0_04 /usr/jdk/entsys-j2se
   ```

   These commands modify the path for J2SE 5.0 Update 4. Modify the path to the J2SE platform according to the version on your system.

   If you installed J2SE 5.0 in a non-default location, replace the default path (`/usr/java/jdk1.5.0_04`) with the path to your non-default location.

2. If you previously stopped services prior to upgrading or installing J2SE 5.0 Update 4, restart the services.

   If you did not stop services prior to upgrading or installing J2SE 5.0 you might want to reboot your system so services that depend on J2SE 5.0 use the new symbolic link.

Verifying the Upgrade of J2SE

The following command verifies the version of J2SE referenced by the J2SE symbolic link:

```
/usr/jdk/entsys-j2se/bin/java -version
```

The command returns a string containing the developer version number. For example, if you installed J2SE 5.0 Update 4, then this command returns the following string:

```
java version "1.5.0_04"
```

If the above command does not return the correct version, check that Java ES symbolic link to J2SE is set correctly, as described in "Setting the J2SE Symlink for Java ES on the Linux Platform."

Chapter 3

Sun Cluster Software

This chapter describes how to upgrade Sun Cluster software to Java ES 2005Q4 (Release 4): Sun Cluster 3.1 8/05.

The chapter provides a general overview of upgrade issues and procedures for upgrading Sun Cluster software to Java ES Release 4.

Sun Cluster software is supported only on Solaris platforms.

The upgrade of Sun Cluster software described in this chapter includes both Sun Cluster framework software and Sun Cluster data-service software, or agents.

Overview of Sun Cluster Software Upgrades

This section describes the following general aspects of Sun Cluster software that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Sun Cluster Software

Java ES Release 4 Sun Cluster software includes a number of new features, including improved cluster installation and upgrade functionality, enhanced support for Network Appliance NAS devices, a simplified SunPlex Manager interface, and other features detailed in the *Sun Cluster Release Notes*, http://docs.sun.com/doc/819-1405/6n3p13hac?a=view

Sun Cluster Software Upgrade Roadmap

Table 3-1 shows the supported Sun Cluster upgrade paths to Java ES Release 4. The table applies to the Solaris operating system only.

Sun Cluster versions do not map one-to-one to Java ES releases. This is because Sun Cluster software's interim feature releases (IFRs) were incorporated into Java ES between formal Java ES releases. For this reason, the upgrade of Java ES Release 3 Sun Cluster and Java ES Release 2 Sun Cluster to Java ES Release 4 Sun Cluster, as shown in Table 3-1, includes the upgrade of both Sun Cluster 3.1 4/04 and Sun Cluster 3.1 9/04 software to Java ES Release 4.

Table 3-1 Upgrade Paths to Java ES Release 4 Sun Cluster 3.1 8/05 (2005Q4) Software

Java ES Release	Sun Cluster Software Version	General Approach	Re-configuration Required
Release 3	Sun Cluster 3.1 9/04 or Sun Cluster 3.1 8/05	Direct upgrade: Performed using the Sun Cluster `scinstall` utility.	Cluster configuration migrated to upgraded version automatically
Release 2	Sun Cluster 3.1 4/04 or Sun Cluster 3.1 9/04	Direct upgrade: Performed using the Sun Cluster `scinstall` utility.	Cluster configuration migrated to upgraded version automatically
Release 1	Sun Cluster 3.1	Direct upgrade not certified: But it can be performed using the `scinstall` utility.	Cluster configuration migrated to upgraded version automatically
Pre-dates Java ES releases	Sun Cluster 3.0	Direct upgrade not certified: But it can be performed using the `scinstall` utility.	Cluster configuration migrated to upgraded version automatically

Sun Cluster Data

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

Table 3-2 Sun Cluster Data Usage

Type of Data	Location	Usage
Cluster configuration data	Cluster Configuration Repository, which is replicated and synchronized across all cluster nodes (CAUTION: Never edit CCR files manually; this can cause a node or the entire cluster to stop functioning)	Stores configuration information for all aspects of Sun Cluster operations: cluster node configuration, failover mechanisms, resource management, and so forth

Compatibility Issues

Java ES Release 4 Sun Cluster software includes new graphical administration interfaces, but is backwardly compatible with earlier releases of Sun Cluster agents.

Sun Cluster Dependencies

Sun Cluster dependencies on other Java ES components can impact the procedure for upgrading and re-configuring Sun Cluster software. Changes in Sun Cluster interfaces or functions, for example, could require upgraded versions of components upon which Sun Cluster software depends. The need to upgrade such components depends upon the specific upgrade path.

Sun Cluster has dependencies on the following Java ES components:

- **Shared components.** Sun Cluster software has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Data services.** Sun Cluster software requires specific data services (or agents) to make Java ES product components highly available. For each product component running in a Sun Cluster environment there must be a corresponding data service to manage the corresponding cluster resources. Agent packages are typically upgraded as part of the Sun Cluster upgrade process.

Upgrading Sun Cluster Software to Java ES Release 4

This section includes information about upgrading Sun Cluster software from both Java ES 2005Q1 (Release 3) and Java ES 2004Q2 (Release 2) to Java ES Release 4. The upgrade procedure is the same for the two Sun Cluster versions found in these Java ES releases: Sun Cluster 3.1 4/04 and Sun Cluster 3.1 9/04 software.

The section covers the following topics:

Introduction

When upgrading Sun Cluster software to Java ES Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by running the `scinstall` script which upgrades Sun Cluster software and applies the previous Sun Cluster configuration after the software upgrade is complete. However all nodes in a cluster environment must be upgraded to the same version, either by shutting down the cluster and upgrading all nodes, or through a rolling upgrade in which the nodes are successively upgraded one at a time without shutting down the cluster.
- **Upgrade Dependencies.** While Sun Cluster software has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Java ES Release 4 Sun Cluster software is compatible with the Release 3 versions of these components. Upgrade of these shared components is therefore optional with respect to upgrade of Sun Cluster software to Release 4.
- **Backward Compatibility.** Release 4 Sun Cluster software is backwardly compatible with earlier cluster agents, however all nodes in a cluster must run the same version of framework and agent software.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Sun Cluster software to earlier versions is not supported.
- **Platform Issues.** The approach for upgrading Sun Cluster software is the same on all Solaris platforms, however Sun Cluster software is not supported on Linux platforms.

Sun Cluster Upgrade

This section provides an overview of how to perform an upgrade of Sun Cluster software from Java ES Release 3 to Java ES Release 4:

- Pre-Upgrade Tasks
- Upgrading Sun Cluster Software
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

The section covers the case of a nonrolling Sun Cluster upgrade. The case of a rolling upgrade is a bit different, in that the cluster is not shut down. However both cases involve the same general procedures, as described below, for a given cluster node. The specific procedures can be found in the upgrade chapter of the *Sun Cluster Installation Guide*, `http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view`.

Pre-Upgrade Tasks

Before you upgrade Sun Cluster software you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Sun Cluster software by entering the following command:

```
% scinstall -pv
```

The command returns the Sun Cluster version and the version of each software package installed. If this command returns the 3.1 8/05 version, `3.1u4`, then no upgrade to Java ES Release 4 is needed.

Table 3-3 Sun Cluster Version Verification Outputs

Java ES Release	Sun Cluster Version Number
Release 1 (Sun Cluster 3.1)	`3.1`
Release 2 (Sun Cluster 3.1 4/04)	`3.1u2`
Release 2 or 3 (Sun Cluster 3.1 9/04)	`3.1u3`
Release 3 or 4 (Sun Cluster 3.1 8/05)	`3.1u4`

Prepare the Cluster Node for Upgrade

The cluster node must be removed from the cluster environment before Sun Cluster software can be upgraded:

- **Nonrolling upgrades.** Removing the node from the cluster environment means shutting down the environment: switching resource groups offline, disabling them, shutting down applications running in the environment, backing up shared data, shutting down the cluster, backing up the system disk, and rebooting the node into non-cluster mode.
- **Rolling upgrades.** Removing the node from the cluster environment means moving all resource groups and device groups from the node, backing up shared data and the system disk, and rebooting the node into non-cluster mode.

The details of these operations and others that might need to be performed in specific situations are provided in the upgrade chapter of the *Sun Cluster Installation Guide*, http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view.

Upgrade the Operating System

You might wish to make use of any upgrade downtime to upgrade your operating system to its most current version, and also upgrade the version of volume manager that you are using.

The details of these operations are provided in the upgrade chapter of the *Sun Cluster Installation Guide*, http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view.

Upgrade Sun Cluster Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. Upgrade of Release 3 shared components upon which Sun Cluster software depends is optional, but upgrade of Release 2 shared components to Release 4 is mandatory.

To upgrade all shared components upon which Sun Cluster software depends (see Table 1-6 on page 40), you can follow the instructions provided in the upgrade chapter of the *Sun Cluster Installation Guide*, http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view, except upgrade all the shared components even if minimum version requirements are met.

Obtain Required Configuration Information and Passwords

No special information about your currently installed version is needed. However you will have to log in as superuser to perform the upgrade.

Upgrading Sun Cluster Software

This section discusses considerations that impact the upgrade procedure for Sun Cluster software followed by a description of the procedure itself.

Upgrade Considerations

The upgrade of Sun Cluster software to Java ES Release 4 takes into account the following considerations:

- When upgrading Sun Cluster framework software it is a good idea to upgrade the data services needed to manage highly available Java ES components or other applications that run in your cluster environment.
- Upgrading Sun Cluster software also provides an opportunity to upgrade Java ES components or other applications that run in your cluster environment.

Upgrade Procedure

The procedure below applies to upgrading Sun Cluster software on each cluster node. The steps that follow are very general; details on how to perform these steps are provided in the upgrade chapter of the *Sun Cluster Installation Guide*, `http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view`.

1. Log in as root or become superuser.

   `su -`

2. Change to the following directory on the Java ES distribution:

   `cd /Solaris_`*arch*`/Product/sun_cluster/Solaris_`*ver*`/Tools`

   where *arch* is `sparc` or `x86` and *ver* is `8`, `9`, or `10` for Solaris 8, 9, or 10, respectively.

3. Run the `scinstall` utility.

   `./scinstall`

   A main menu is displayed for performing cluster installation, configuration, and upgrade tasks.

4. Upgrade Sun Cluster framework software and any desired data services.

   Upgraded data services need to be configured by migrating the corresponding resources to the upgraded resource types (see "Post-Upgrade Tasks" on page 101).

5. Apply any necessary patches to Sun Cluster framework software and to data services.

   Information on accessing and applying the relevant patches is provided in the *Sun Cluster Release Notes*, http://docs.sun.com/doc/819-1405.

6. Reboot the node into the cluster.

Verifying the Upgrade

You can verify successful upgrade of Sun Cluster software as follows:

1. Check the version number of Sun Cluster framework software.

   ```
   scinstall -pv
   ```

   See Table 3-3 on page 98 for output values.

2. Check the data service upgrade log file.

   The log file is referenced at the end of upgrade output messages.

Post-Upgrade Tasks

After you perform the upgrade of Sun Cluster software, you might need to perform a number of additional tasks, depending on whether you performed a nonrolling or a rolling upgrade. Among the tasks required to fully restore your cluster environment are:

- Verifying the status of the cluster configuration
- Migrating resources to new resource type versions
- Upgrading additional Java ES components or applications that are installed on the cluster

Details for these post-installation steps are provided in the upgrade chapter of the *Sun Cluster Installation Guide*, http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view.

Rolling Back the Upgrade

Rollback of Sun Cluster software is not supported. Changes made during the upgrade procedure cannot easily be backed out.

Chapter 4

Directory Server and Administration Server

This chapter describes how to upgrade Directory Server and Administration Server components to Java ES 2005Q4 (Release 4): Sun Java System Directory Server 5.2 2005Q4 and Sun Java System Administration Server 5.2 2005Q4.

These upgrades are documented together because they work closely together.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

- "Overview of Directory Server and Administration Server Upgrades" on page 104
- "Upgrading Directory Server and Administration Server from Java ES Release 3" on page 107
- "Upgrading Directory Server and Administration Server from Java ES Release 2" on page 122

NOTE File locations in this chapter are specified with respect to a directory path referred to as *serverRoot*. At least part of this path might have been specified as an installation directory when Directory Proxy Server was initially installed and configured. If not, a default value was assigned.

The default value of *serverRoot* depends on operating system platform:

- Solaris: `/var/opt/mps/serverroot`
- Linux: `/var/opt/sun/directory-server`

Overview of Directory Server and Administration Server Upgrades

This section describes the following general aspects of Directory Server and Administration Server components that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4

Java ES Release 4 versions of Directory Server and Administration Server represent only minor bug fixes and improvements. There are no new functional capabilities.

Java ES Release 4 Upgrade Roadmap

Table 4-1 shows the supported Directory Server and Administration Server upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 4-1 Upgrade Paths to Java ES Release 4: Sun Java System Directory Server 5.2 2005Q4 and Sun Java System Administration Server 5.2 2005Q4

Java ES Release	Directory Server, Administration Server, and Directory Proxy Server Version	General Approach	Re-configuration Required
Release 3	Sun Java System Directory Server 5 2005Q1 Sun Java System Administration Server 5 2005Q1	Direct upgrade: Apply patches and re-configure configuration directory.	Automatic re-configuration of data in configuration directory

Table 4-1 Upgrade Paths to Java ES Release 4: Sun Java System Directory Server 5.2 2005Q4 and Sun Java System Administration Server 5.2 2005Q4 *(Continued)*

Java ES Release	Directory Server, Administration Server, and Directory Proxy Server Version	General Approach	Re-configuration Required
Release 2	Sun Java System Directory Server 5.2 2004Q2 Sun Java System Administration Server 5.2 2004Q2	Direct upgrade: Apply patches and re-configure configuration directory.	Automatic re-configuration of data in configuration directory
Release 1	Sun One Directory Server 5.2 Sun One Administration Server 5.2	Direct upgrade not certified: But you can use the same approach as upgrading from Release 2.	Automatic re-configuration of data in configuration directory
Pre-dates Java ES releases	Sun One Directory Server 5.2 Sun One Administration Server 5.2	Direct upgrade not certified: But you can use the same approach as upgrading from Release 2.	Automatic re-configuration of data in configuration directory
	Sun One Directory Server 5.1, 5.0, or 4.x Sun One Administration Server 5.1, 5.0, or 4.x	No direct upgrade: Upgrade first to Release 3. Refer to the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062). Then upgrade from Release 3 to Release 4.	Refer to the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062).

Directory Server and Administration Server Data

Directory Server and Administration Server make use of Directory Server itself for storing configuration data. The data is stored in a specific tree structure within the directory. The Directory Server instance hosting the configuration is referred to as the configuration directory.

The configuration directory can be a dedicated Directory Server instance, which is a recommended security practice, or it can also host user identity data or service configuration data. The configuration directory can reside on the same computer as other Directory Server instances or the Administration Server; however in most deployment architectures, the configuration directory is remote from the other components that use it to store configuration information.

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

Table 4-2 Directory Server, Administration Server, and Directory Proxy Server Data Usage

Type of Data	Location	Usage
Directory Server configuration data	Configuration directory	Configuration of Directory Server
Administration Server configuration data	Configuration directory	Configuration of Administration Server

Compatibility Issues

Java ES Release 4 Directory Server and Administration Server do not introduce any interface changes. These components are, as a group, backwardly compatible with earlier versions. However, both of these components are not backwardly compatible with earlier versions of the others; both need to be upgraded as a unit.

Dependencies

Dependencies on other Java ES components can impact the procedure for upgrading and re-configuring Directory Server and Administration Server software. Each of these components has dependencies on Java ES components as follows:

- **Directory Server.** Directory Server has dependencies on specific Java ES shared components (see Table 1-6 on page 40). Directory Server has a dependency on Administration Server, which is used to configure Directory Server replication and other aspects of Directory Server functions.
- **Administration Server.** Administration Server (and the Administration Console user interface) has dependencies on specific Java ES shared components (see Table 1-6 on page 40). Administration Server has a dependency on Directory Server (specifically a configuration directory) where it stores configuration data.

Upgrading Directory Server and Administration Server from Java ES Release 3

This section includes information about upgrading Directory Server and Administration Server from Java ES 2005 Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Directory Server and Administration Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Java ES Release 3 version. Re-configuration of Directory Server and Administration Server are achieved by synchronizing the configuration directory with the upgraded software.
- **Upgrade Dependencies.** While Directory Server and Administration Server have dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Java ES Release 4 Directory Server and Administration Server are compatible with the Release 3 versions of these shared components. Upgrade of these shared components is therefore optional with respect to upgrade of Directory Server and Administration Server to Release 4.

  Directory Server has a hard upgrade dependency on Administration Server. These components should therefore be upgraded together to Release 4.
- **Backward Compatibility.** Release 4 Directory Server and Administration Server are backwardly compatible with their Release 3 versions.
- **Upgrade Rollback.** A rollback of the Release 4 upgrade is achieved on Solaris by removing the Release 4 upgrade patches and re-synchronizing the configuration directory with the previous software state. On Linux, however, there is no procedure for rolling back the Release 4 upgrade.
- **Platform Issues.** The general approach for upgrading Directory Server and Administration Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Directory Server and Administration Server Upgrade

This section describes how to perform an upgrade of Directory Server and Administration Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Directory Server and Administration Server by restarting the Directory Server daemon using the `-v` option:

```
cd serverRoot/bin/slapd/server
./ns-slapd -v
```

and then checking the startup messages in the Directory Server error log:

```
serverRoot/slapd-hostName/logs/errors
```

Table 4-3 Directory Server Version Verification Outputs

Java ES Release	Directory Server Version Number
Release 2	`Sun Java(TM) System Directory Server/5.2_Patch_2`
Release 3	`Sun Java(TM) System Directory Server/5.2_Patch_3`
Release 4	`Sun Java(TM) System Directory Server/5.2_Patch_4`

Note: If the `ns-slapd` command fails on the Solaris 10 platform, set the library path to null when running the command:

```
LD_LIBRARY_PATH= ./ns-slapd -v
```

Upgrade Directory Server and Administration Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, because Directory Server and Administration Server do not require upgrading Release 3 shared components, this task is optional.

Back Up Directory Server Data

The Directory Server and Administration Server upgrade process modifies configuration directory data. Therefore, before you upgrade, it is recommended that you back up your configuration directory 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* (`http://docs.sun.com/doc/817-7613`).

Obtain Required Configuration Information and Passwords

You should know the Directory Server administrator user ID and password for your currently installed version.

In addition, Directory Server and Administration Server must run as the same user and group. That is, they must run with the same UID and GID.

Upgrading Release 3 Directory Server and Administration Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Directory Server and Administration Server, followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Directory Server and Administration Server software to Java ES Release 4 takes into account the following considerations:

- Any Java ES components using a Directory Server instance (such as Access Manager, Communications Express, Messaging Server, Portal Server, and so forth) should be shut down before you upgrade that instance. However, most deployment architectures use multiple instances of Directory Server to provide high availability or scalability. In such cases, you can perform a rolling upgrade of Directory Server and the Directory Server clients need not be shut down.

- Administration Server must be upgraded before Directory Server because the re-configuration of data must take place in a particular order.
- The component being upgraded must be shut down when patches are being applied, however the associated configuration directory must subsequently be running to re-configure the component being upgraded.
- In a deployment architecture in which there are multiple instances of Directory Server running on a single computer (all corresponding 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 deployment architectures the configuration directory is a separate Directory Server instance. It might be local or on a different computer system from where the upgrade is being performed. Similarly, the Administration Server might be local or on a different computer system from where the upgrade of Directory Server is being performed.
- In some deployment architectures Directory Server has been installed standalone by deselecting Administration Server at installation time. In that case, however, the Administration Server upgrade procedure must still be performed (some Administration Server code is installed even in standalone mode), in addition to the Directory Server upgrade procedure, as described in the instructions that follow.
- The Release 4 Directory Server and Administration Server upgrade patches for Solaris OS are shown in the following table:

Table 4-4 Patches[1] to Upgrade Directory Server and Administration Server on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Directory Server	115614-26	115615-26
Directory Server localization	117015-21	117015-21
Administration Server	115610-23	115611-23
Administration Server localization	117047-24	117047-24

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Directory Server and Administration Server instances residing locally on the computer where the upgrade is taking place.

The steps below make use of two commands: `directoryserver(1m)` and `mpsadmserver(1m)`. For more information about these commands, see the *Directory Server Man Page Reference* and the *Administration Server Man Page Reference.*

1. Obtain the required patches, based on Table 4-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop the Administration Console if it is running locally.

4. Shut down all Java ES components dependent on the Directory Server instances that are to be upgraded. This step might depend on how these components are replicated within your deployment architecture.

   Components should be shut down in the following order:

   a. Directory Server clients: Access Manager, Communications Express, Messaging Server, Portal Server, and others

   b. Directory Proxy Server, if being used to access Directory Server

   c. Administration Server, if running locally

   d. Directory Server

   e. Configuration directory, if running locally as a separate Directory Server instance.

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

5. Make sure you have upgraded any Java ES components upon which Directory Server and Administration Server have hard upgrade dependencies (see "Upgrade Directory Server and Administration Server Dependencies" on page 109).

6. Upgrade Administration Server.

   You need to perform this step even if Directory Server had originally been installed in standalone mode on the computer where the upgrade is taking place (some Administration Server code is installed even in standalone mode).

   a. Restart the Administration Server to be upgraded.

   b. Apply the Administration Server patches in Table 4-4.

      Be sure to apply the Administration Server localization patch (117047) before applying the Administration Server base patch.

      `patchadd` *patch_ID*

   c. Confirm that the patch upgrade was successful:

      `showrev -p | grep` *patch_ID*

      The output should return the versions of patch IDs applied in Step b.

   d. Ensure that the configuration directory is running.

      If it is local you might have to start it up. If it is remote, check to make sure it is running.

   e. Synchronize the upgraded settings with the configuration directory.

      `/usr/sbin/mpsadmserver sync-cds`

      You will be prompted for the admin username and password.

7. Upgrade Directory Server.

   a. If you are running Directory Server in standalone mode, without Administration Server, perform the following procedure, otherwise proceed directly to Step 7b.

      I. Ensure that you have upgraded Administration Server, Step 6.

      II. Change directory to the *serverroot* directory.

      `cd /var/opt/mps/`*serverroot*

      III. Create a configuration directory:

      `mkdir -p admin-serv/config`

      IV. Create an `adm.config` file:

      `vi admin-serv/config/adm.conf`

v. Add the following text

```
isie: cn=Administration Server, cn=Server Group, cn=hostname,
ou=administration_domain, o=NetscapeRoot
```

All on one line where *hostname* is the fully qualified Directory Server host name and *administration_domain* is typically the host's domain name.

b. Ensure that the Directory Server instance being upgraded is shut down.

c. Apply the Directory Server patches in Table 4-5.

Be sure to apply the Directory Server localization patch (117015) before applying the Directory Server base patch.

```
patchadd patch_ID
```

d. Confirm that the patch upgrade was successful:

```
showrev -p | grep patch_ID
```

The output should return the versions of patch IDs applied in Step c.

e. Reset the default Directory Server version number:

```
/usr/sbin/directoryserver -d 5.2
```

f. Ensure that the configuration directory is running.

If it is local you might have to start it up. If it is remote, check to make sure it is running.

g. Synchronize the upgraded settings with the configuration directory.

```
/usr/sbin/directoryserver -u 5.2 sync-cds
```

You will be prompted for the admin username and password.

8. Restart all Java ES components in the reverse order they were shut down in Step 4.

a. Configuration directory, if local and running as a separate Directory Server instance

b. Directory Server

c. Administration Server, if running locally

d. Directory Proxy Server, if being used to access Directory Server

e. Directory Server clients: Access Manager, Communications Express, Messaging Server, Portal Server, and others

Upgrading Release 3 Directory Server and Administration Server (Linux)

This section discusses considerations that impact the upgrade procedure for Directory Server and Administration Server, followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Directory Server and its associated components to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 109), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Directory Server and Administration Server upgrade patches for Linux OS are shown in the following table:

Table 4-5 Patches[1] to Upgrade Directory Server and Administration Server on Linux

Description	Patch ID and RPM names

Directory Server	118080-11:
	`sun-directory-server-5.2-25.i386.rpm`
	`sun-directory-server-man-5.2-9.i386.rpm`

Directory Server localization	118290-12: `sun-directory-server-`*`Locale`*`-5.2-17.i386.rpm`

Administration Server	118079-10:
	`sun-admin-server-5.2-18.i386.rpm`
	`sun-server-console-5.2-18.i386.rpm`
	`sun-admin-server-man-5.2-8.i386.rpm`

Administration Server localization	118289-13:
	`sun-admin-server-`*`Locale`*`-5.2-19.i386.rpm`
	`sun-server-console-`*`Locale`*`-5.2-19.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies Directory Server and Administration Server instances residing locally on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

The steps below make use of two commands: `directoryserver(1m)` and `mpsadmserver(1m)`. For more information about these commands, see the *Directory Server Man Page Reference* and the *Administration Server Man Page Reference*.

1. Obtain the required patches using the patch numbers and RPM names from Table 4-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

   In the following procedure *oldVersion* signifies the RPM for the Release 3 version of Directory Server and Administration Server.

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop the Administration Console if it is running locally.

4. Shut down all Java ES components dependent on the Directory Server instances that are to be upgraded. This step might depend on how these components are replicated within your deployment architecture.

   Components should be shut down in the following order:

   a. Directory Server clients: Access Manager, Communications Express, Messaging Server, Portal Server, and others

   b. Directory Proxy Server, if being used to access Directory Server

   c. Administration Server, if running locally

   d. Directory Server

   e. Configuration directory, if running locally as a separate Directory Server instance.

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

5. Make sure you have upgraded any Java ES components upon which Directory Server and Administration Server have hard upgrade dependencies (see "Upgrade Directory Server and Administration Server Dependencies" on page 109).

6. Apply each of the RPMs for Administration Server.

   a. Apply the RPM for Administration Server: Product.

      You need to perform this step even if Directory Server had originally been installed in standalone mode on the computer where the upgrade is taking place.

      I. Apply the RPM as follows:

         Be sure to apply the Administration Server localization RPMs (118289) before applying the Administration Server base RPMs.

         ```
         rpm -Fvh sun-admin-server-Locale-5.2-19.i386.rpm
         rpm -Fvh sun-server-console-Locale-5.2-19.i386.rpm
         rpm -Fvh sun-admin-server-5.2-18.i386.rpm
         ...
         ```

         If your Administration Server was configured previously, the following error will be returned:

         ```
         error: execution of %preun scriptlet from
         sun-admin-server-5.2-oldVersion failed, exit status 1
         ```

         If this is the case, remove the old version of the RPM using the `--noscripts` option, as follows:

         ```
         rpm -e --noscripts sun-admin-server-5.2-oldVersion
         ```

      II. If your Administration Server was configured previously, ensure that the configuration directory is running.

         If it is local you might have to start it up. If it is remote, check to make sure it is running.

      III. Synchronize the upgraded settings with the configuration directory.

         ```
         /opt/sun/sbin/mpsadmserver sync-cds
         ```

         You will be prompted for the admin username and password.

      IV. Confirm that the upgrade was successful:

         ```
         rpm -q sun-admin-server
         ```

         The new version number of the RPM should be returned.

b. Apply the RPM for the Administration Server: Console.

```
rpm -Fvh sun-server-console-5.2-18.i386.rpm
```

c. Apply the RPM for the Administration Server: man pages.

```
rpm -Uvh sun-admin-server-man-5.2-8.i386.rpm
```

7. Apply each of the RPMs for Directory Server.

a. If you are running Directory Server in standalone mode, without Administration Server, apply the Administration Server RPM.

```
rpm -Fvh sun-admin-server-5.2-18.i386.rpm
```

Otherwise proceed directly to Step 7b.

b. Apply the RPM for the Directory Server: Product.

I. Ensure that the Directory Server instance being upgraded is shut down.

II. Apply the RPM as follows:

Be sure to apply the Directory Server localization RPMs (`118290`) before applying the Directory Server RPMs.

```
rpm -Fvh sun-directory-server-Locale-5.2-17.i386.rpm
rpm -Fvh sun-directory-server-5.2-25.i386.rpm
...
```

If your Directory Server was configured previously, the following error will be returned:

```
error: execution of %preun scriptlet from
sun-directory-server-5.2-oldVersion failed, exit status 1
```

If this is the case, remove the old version of the RPM using the `--noscripts` option, as follows:

```
rpm -e --noscripts sun-directory-server-5.2-oldVersion
```

III. If your Directory Server was configured previously, ensure that the configuration directory is running

If it is local you might have to start it up. If it is remote, check to make sure it is running.

IV. Synchronize the upgraded settings with the configuration directory.

```
/opt/sun/sbin/directoryserver sync-cds
```

You will be prompted for the admin username and password.

v. Confirm that the upgrade was successful:

```
rpm -q sun-directory-server
```

The new version number of the RPM should be returned.

c. Apply the RPM for the Directory Server: man pages.

```
rpm -Uvh sun-directory-server-man-5.2-9.i386.rpm
```

8. Restart all Java ES components in the reverse order they were shut down in Step 4.

a. Configuration directory, if local and running as a separate Directory Server instance

b. Directory Server

c. Administration Server, if running locally

d. Directory Proxy Server, if being used to access Directory Server

e. Directory Server clients: Access Manager, Communications Express, Messaging Server, Portal Server, and others

Verifying the Upgrade

You can verify successful upgrade of Directory Server and Administration Server by restarting the Directory Server daemon using the `-v` option:

```
cd serverroot/bin/slapd/server
./ns-slapd -v
```

and then checking the startup messages in the Directory Server error log:

`/var/opt/mps/`*`serverroot`*`/logs/errors`

See Table 4-3 on page 108 for output values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure (Solaris)” on page 111 and “Upgrade Procedure (Linux)” on page 115.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Directory Server and Administration Server, followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Directory Server and Administration Server is pretty much the reverse of the procedure for upgrading to Release 4. The patches are removed and the configuration directory is re-synchronized.

One special consideration is that when you apply patches, you upgrade the SSL certificate database to a cert8 format. The patch backs up the cert7 data, and then converts it to cert8 format. If you subsequently decide to roll back the upgrade and have added new certificates to the certificate database, you should manually extract these certificates, back out the patches, and then add the certificates back to the previous cert7 format certificate database.

When you roll back an upgrade after having changed the SSL certificate database, you cannot start in SSL mode. To work around this problem, turn off SSL mode, restart Directory Server and Administration Server, reinstall the certificate, and then enable SSL mode.

Rollback Procedure (Solaris)

1. Stop the Administration Console if it is running locally.
2. Shut down all Java ES components dependent on the Directory Server instances that are to be rolled back. This step depends on how these components are replicated within your deployment architecture.

   Components should be shut down in the following order:

   a. Directory Server clients: Access Manager, Communications Express, Messaging Server, Portal Server, and others

   b. Directory Proxy Server, if being used to access Directory Server

   c. Administration Server, if running locally

   d. Directory Server

   e. Configuration directory, if running locally as a separate Directory Server instance.

   For information about how to shut down a Java ES component, see its respective administration guide.
3. Roll back the Directory Server upgrade.

   a. Ensure that the Directory Server instance being rolled back is shut down.

   b. Remove the Directory Server patches in Table 4-5.

      `patchrm` *patch_ID*

c. Ensure that the configuration directory is running.

If it is local you might have to start it up. If it is remote, check to make sure it is running.

d. Synchronize the rolled back settings with the configuration directory.

```
/usr/sbin/directoryserver -u 5.2 sync-cds
```

You will be prompted for the admin username and password.

e. If you are running Directory Server standalone, without Administration Server, you must roll back the partial Administration Server upgrade, follow the instructions in Step 4.

4. Roll back the Administration Server upgrade.

a. Remove the Administration Server patches in Table 4-5.

`patchrm` *patch_ID*

b. Ensure that the configuration directory is running.

If it is local you might have to start it up. If it is remote, check to make sure it is running.

c. Synchronize the upgraded settings with the configuration directory.

```
/usr/sbin/mpsadmserver sync-cds
```

You will be prompted for the admin username and password.

5. Roll back upgrades to any Java ES components upon which Directory Server and Administration Server have hard upgrade dependencies.

6. Restart all Java ES components in the reverse order they were shut down in Step 2.

a. Configuration directory, if local and running as a separate Directory Server instance

b. Directory Server

c. Administration Server, if running locally

d. Directory Proxy Server, if being used to access Directory Server

e. Directory Server clients: Access Manager, Communications Express, Messaging Server, Portal Server, and others

Multiple Instance Upgrades

The procedures in "Release 3 Directory Server and Administration Server Upgrade" on page 108 do not explicitly deal with deployment architectures in which Directory Server is replicated for availability or scalability. These architectures might include Directory Server multi-master replication or the deployment of Directory Server as a data service in a Sun Cluster environment.

This section discusses Directory Server upgrades in these situation.

Rolling Upgrades of Multimaster Replicates

Multiple instances of Directory Server on different computer systems, as used in multimaster replication deployment architectures, can be sequentially upgraded one instance at a time. The upgrade of each instance on its respective host computer is performed while the other instances are left running. This rolling upgrade allows the directory service to remain online while the individual Directory Server instances that provide the service are being upgraded.

Upgrading Directory Server as a Data Service

This section describes how to upgrade and roll back Directory Server as a data service in a Sun Cluster environment. Consider the following points before you upgrade or back out Directory Server as a Sun Cluster data service:

- Back up data before performing an upgrade or rollback operation.
- Patch Directory Server and its associated Administration Server on all cluster nodes sequentially rather than in parallel.
- All cluster nodes should run the same version and release of Directory Server and its associated Administration Server.
- If you are running the cluster in failover mode, consider upgrading from HAStorage to HAStoragePlus.

Upgrading Directory Server as a Sun Cluster Data Service

1. Stop each Directory Server instance and its associated Administration Server.

   ```
   serverroot/stop-admin
   serverroot/slapd-instanceName/stop-slapd
   ```

2. Make the current cluster node the active node:

   ```
   scswitch -z -g ldap-group -h this-node-name
   ```

3. Upgrade Directory Server on the current node as described in "Release 3 Directory Server and Administration Server Upgrade" on page 108.

4. Make another cluster node the active node:

   `scswitch -z -g` *ldap-group* `-h` *another-node-name*

5. Repeat Step 3 and Step 4 until all nodes in the cluster are upgraded.

Rolling Back Directory Server as a Sun Cluster Data Service

1. Stop each Directory Server instance and its associated Administration Server.

   `serverroot/stop-admin`
   `serverroot/slapd-`*instanceName*`/stop-slapd`

2. Make the current cluster node the active node:

   `scswitch -z -g` *ldap-group* `-h` *this-node-name*

3. Roll back Directory Server on the current node as described in "Rolling Back the Upgrade (Solaris)" on page 118.

4. Make another cluster node the active node:

   `scswitch -z -g` *ldap-group* `-h` *another-node-name*

5. Repeat Step 3 and Step 4 until Directory Server is rolled back on all nodes in the cluster.

Upgrading Directory Server and Administration Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Directory Server and Administration Server to Release 4 is the same as that for upgrading Release 3 Directory Server and Administration Server to Release 4, with the exception that the pre-upgrade tasks should include the upgrading to Release 4 of all shared components (see Table 1-6 on page 40) and all locally-resident product components upon which Directory Server and Administration Server depend:

Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.

To upgrade Release 2 Directory Server and Administration Server to Release 4, use the instructions in "Upgrading Directory Server and Administration Server from Java ES Release 3" on page 107, except substitute Release 2 wherever Release 3 is referenced.

Chapter 5

Directory Proxy Server

This chapter describes how to upgrade Directory Proxy Server to Java ES 2005Q4 (Release 4): Sun Java System Directory Proxy Server 5.2 2005Q4.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

NOTE File locations in this chapter are specified with respect to a directory path referred to as *serverRoot*. At least part of this path might have been specified as an installation directory when Directory Proxy Server was initially installed and configured. If not, a default value was assigned.

The default value of *serverRoot* depends on operating system platform:

- Solaris: `/var/opt/mps/serverroot`
- Linux: `/var/opt/sun/directory-server`

Overview of Directory Proxy Server Upgrades

This section describes the following general aspects of Directory Proxy Server that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4

Java ES Release 4 Directory Proxy Server represents only minor bug fixes and improvements. There are no new functional capabilities.

Java ES Release 4 Upgrade Roadmap

Table 5-1 shows the supported Directory Proxy Server upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 5-1 Upgrade Paths to Java ES Release 4: Sun Java System Directory Proxy Server 5.2 2005Q4

Java ES Release	Directory Proxy Server Version	General Approach	Re-configuration Required
Release 3	Sun Java System Directory Proxy Server 5.2 2005Q1	Direct upgrade: Apply patches and re-configure configuration directory.	Automatic re-configuration of data in configuration directory
Release 2	Sun Java System Directory Proxy Server 5.2 2004Q2	Direct upgrade: Apply patches and re-configure configuration directory.	Automatic re-configuration of data in configuration directory
Release 1	Sun One Directory Proxy Server 5.2	Direct upgrade not certified: But you can use the same approach as upgrading from Release 2.	Automatic re-configuration of data in configuration directory

Table 5-1 Upgrade Paths to Java ES Release 4: Sun Java System Directory Proxy Server 5.2 2005Q4 *(Continued)*

Java ES Release	Directory Proxy Server Version	General Approach	Re-configuration Required
Pre-dates Java ES releases	Sun One Directory Proxy Server 5.2	Direct upgrade not certified: But you can use the same approach as upgrading from Release 2.	Automatic re-configuration of data in configuration directory
	Sun One Directory Access Router 5.0 or 5.0 SP1	No direct upgrade: Upgrade first to Release 3. Refer to the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062). Then upgrade from Release 3 to Release 4.	Refer to the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062).

Directory Proxy Server Data

Directory Proxy Server makes use of Directory Server for storing configuration data. The data is stored in a specific tree structure within the directory. The Directory Server instance hosting the configuration is referred to as the configuration directory.

In most deployment architectures, the configuration directory is remote from the other components that use it to store configuration information.

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

Table 5-2 Directory Proxy Server Data Usage

Type of Data	Location	Usage
Directory Proxy Server configuration data	Configuration directory	Configuration of Directory Proxy Server

Compatibility Issues

Java ES Release 4 Directory Proxy Server does not introduce any interface changes and is backwardly compatible with earlier versions.

Dependencies

Dependencies on other Java ES components can impact the procedure for upgrading and re-configuring Directory Proxy Server software. Directory Proxy Server has dependencies on specific Java ES shared components (see Table 1-6 on page 40). Directory Proxy Server provides front-end access to Directory Server and uses Administration Server for configuration purposes. Directory Proxy Server therefore has dependencies on both Directory Server and Administration Server.

Upgrading Directory Proxy Server from Java ES Release 3

This section includes information about upgrading Directory Proxy Server from Java ES 2005 Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Directory Proxy Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Java ES Release 3 version. Re-configuration of Directory Proxy Server is achieved by automatically synchronizing the configuration directory with the upgraded software.

- **Upgrade Dependencies.** While Directory Proxy Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Java ES Release 4 Directory Proxy Server is compatible with the Release 3 versions of these shared components. Upgrade of these shared components is therefore optional with respect to upgrade of Directory Proxy Server to Release 4.

  Directory Proxy Server has a hard upgrade dependency on both Directory Server and Administration Server. All three components should therefore be upgraded together to Release 4.

- **Backward Compatibility.** Release 4 Directory Proxy Server is backwardly compatible with its Release 3 version.

- **Upgrade Rollback.** A rollback of the Release 4 upgrade is achieved on Solaris platforms by removing the Release 4 upgrade patches. On the Linux platform, however, there is no procedure for rolling back the Release 4 upgrade.

- **Platform Issues.** The general approach for upgrading Directory Proxy Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Directory Proxy Server Upgrade

This section describes how to perform an upgrade of Directory Proxy Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Directory Proxy Server using the following commands:

```
cd serverRoot/bin/dps/server/bin
./ldapfwd -v
```

The output is shown in the following table:

Table 5-3 Directory Proxy Server Version Verification Outputs

Java ES Release	Directory Proxy Server Version Number
Release 2	`Sun ONE Directory Proxy Server Version 5.2_Patch_2`
Release 3	`Sun ONE Directory Proxy Server Version 5.2_Patch_3`
Release 4	`Sun ONE Directory Proxy Server Version 5.2_Patch_4`

Upgrade Directory Proxy Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4.

Directory Proxy Server has hard upgrade dependencies on Directory Server and Administration Server, even when they run on remote computers, so these components should be upgraded before upgrading Directory Proxy Server.

Upgrading of Java ES Release 3 shared components upon which Directory Proxy Server depends is optional, but recommended.

You can upgrade Directory Proxy Server dependencies in the following order, all before you upgrade Directory Proxy Server. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, "Directory Server and Administration Server" on page 103.

Back Up Directory Server Data

The Directory Proxy Server upgrade process modifies configuration directory data. Therefore, before you upgrade, it is recommended that you back up your configuration directory 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* (`http://docs.sun.com/doc/817-7613`).

Obtain Required Configuration Information and Passwords

Directory Proxy Server must run as the same user and group as Directory Server and Administration Server. That is, they must all run with the same UID and GID.

Upgrading Release 3 Directory Proxy Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Directory Proxy Server followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Directory Proxy Server software to Java ES Release 4 takes into account the following considerations:

- Any Java ES components using a Directory Proxy Server instance (such as Access Manager, Communications Express, Messaging Server, Portal Server, and so forth) should be shut down before you upgrade that instance. However, many deployment architectures use multiple instances of Directory Proxy Server to provide high availability or scalability. In such cases, you can perform a rolling upgrade of Directory Proxy Server and the Directory Proxy Server clients need not be shut down.
- The upgrade of Directory Proxy Server should only be performed after the upgrade of Administration Server and Directory Server because re-configuration must take place in a particular order.
- Directory Proxy Server must be shut down when patches are being applied, however the associated configuration directory must be running to perform re-configuration.
- In a deployment architecture in which there are multiple instances of Directory Proxy Server running on a single computer (all corresponding to the same installed Directory Proxy Server image), upgrading the Directory Proxy Server image will upgrade all the instances. In such architectures, there is only one Administration Server instance per installed Directory Proxy Server image.

- The Release 4 Directory Proxy Server upgrade patches for Solaris OS are shown in the following table:

Table 5-4 Patches[1] to Upgrade Directory Proxy Server on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Directory Proxy Server	116373-18	116374-18
Directory Proxy Server localization	117017-20	117017-20

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Directory Proxy Server instances residing locally on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 5-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop the Administration Console if it is running locally.

4. 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.

5. Make sure you have upgraded any Java ES components upon which Directory Proxy Server has hard upgrade dependencies (see “Upgrade Directory Proxy Server Dependencies” on page 128).

6. Upgrade Directory Proxy Server.

   a. Ensure that the configuration directory is running.

      If it is local you might have to start it up. If it is remote, check to make sure it is running.

   b. Apply the Directory Proxy Server patches in Table 5-4.

      Be sure to apply the Directory Proxy Server localization patch (117017) before applying the Directory Proxy Server base patch.

      `patchadd` *patch_ID*

   c. Confirm that the patch upgrade was successful:

      `showrev -p | grep` *patch_ID*

      The output should return the versions of patch IDs applied in Step b.

7. Restart Directory Proxy Server and all Java ES components dependent on Directory Proxy Server.

   To restart Directory Proxy Server:

   serverRoot`/dps-`*hostName*`/restart-dps`

Upgrading Release 3 Directory Proxy Server (Linux)

This section discusses considerations that impact the upgrade procedure for Directory Proxy Server followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Directory Proxy Server to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 129), except that the Linux Release 4 upgrade patches differ from the Solaris OS patches.

The Release 4 Directory Proxy Server upgrade patch for Linux OS is shown in the following table:

Table 5-5 Patches[1] to Upgrade Directory Proxy Server on Linux

Description	Patch ID and RPM names
Directory Proxy Server	118096-08: `sun-directory-proxy-server-5.2-13.i386.rpm`

Table 5-5 Patches[1] to Upgrade Directory Proxy Server on Linux *(Continued)*

Description	Patch ID and RPM names
Directory Proxy Server localization	118288-11: `sun-directory-proxy-server-`*Locale*`-5.2-16.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to Directory Proxy Server instances residing locally on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patch using the patch number and RPM names from Table 5-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop the Administration Console if it is running locally.

4. 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.

5. Make sure you have upgraded any Java ES components upon which Directory Proxy Server has hard upgrade dependencies (see "Upgrade Directory Proxy Server Dependencies" on page 128).

6. Apply the RPMs for Directory Proxy Server.

   a. Ensure that the configuration directory is running.

      If it is local you might have to start it up. If it is remote, check to make sure it is running.

b. Apply the RPMs.

Be sure to apply the Directory Proxy Server localization RPM before applying the Directory Proxy Server base RPM.

```
rpm -Fvh sun-directory-proxy-server-Locale-5.2-16.i386.rpm
rpm -Fvh sun-directory-proxy-server-5.2-13.i386.rpm
```

The upgraded settings are automatically synchronized with the configuration directory.

7. Restart Directory Proxy Server and all Java ES components dependent on Directory Proxy Server.

To restart Directory Proxy Server:

serverRoot/dps-*hostName*/restart-dps

Verifying the Upgrade

You can verify successful upgrade of Directory Proxy Server using the following commands:

```
cd serverRoot/bin/dps/server/bin
./ldapfwd -v
```

See Table 5-3 on page 128 for output values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure (Solaris)” on page 130 and “Upgrade Procedure (Linux)” on page 132.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Directory Proxy Server, followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Directory Proxy Server is pretty much the reverse of the procedure for upgrading to Release 4. The patches are removed and the configuration directory is re-synchronized.

One special consideration is that when you apply patches, you upgrade the SSL certificate database to a cert8 format. The patch backs up the cert7 data, and then converts it to cert8 format. If you subsequently decide to roll back the upgrade and have added new certificates to the certificate database, you should manually extract these certificates, back out the patches, and then add the certificates back to the previous cert7 format certificate database.

When you roll back an upgrade after having changed the SSL certificate database, you cannot start in SSL mode. To work around this problem, turn off SSL mode, restart Administration Server and Directory Proxy Server, reinstall the certificate, and then enable SSL mode.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop the Administration Console if it is running locally.

3. 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.

4. Roll back the Directory Proxy Server upgrade.

   a. Ensure that the configuration directory is running.

      If it is local you might have to start it up. If it is remote, check to make sure it is running.

   b. Remove the Directory Proxy Server patches in Table 5-5.

      `patchrm` *patch_ID*

5. Roll back upgrades to any Java ES components upon which Directory Proxy Server has hard upgrade dependencies, in particular Directory Server and Administration Server.

6. Restart Directory Proxy Server and all Java ES components dependent on Directory Proxy Server.

Multiple Instance Upgrades

In some deployment architectures Directory Proxy Server is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Directory Proxy Server components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Directory Proxy Server, you can perform a rolling upgrade in which you upgrade the Directory Proxy Server instances sequentially without interrupting service. You upgrade each instance of Directory Proxy Server while the others remain running. You perform the upgrade of each instance as described in "Release 3 Directory Proxy Server Upgrade" on page 127.

Upgrading Directory Proxy Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Directory Proxy Server to Release 4 is the same as that for upgrading Release 3 Directory Proxy Server to Release 4, with the exception that the pre-upgrade tasks should include the upgrading to Release 4 of all shared components (see Table 1-6 on page 40) and all locally-resident product components upon which Directory Proxy Server depends.

Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.

To upgrade Release 2 Directory Proxy Server to Release 4, use the instructions in "Upgrading Directory Proxy Server from Java ES Release 3" on page 126, except substitute Release 2 wherever Release 3 is referenced.

Chapter 6

Web Server

This chapter describes how to upgrade Web Server to Java ES 2005Q4 (Release 4): Sun Java System Web Server 6.1 SP5 2005Q4.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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 value of *WebServer-base* depends on operating system platform:

- Solaris: `/opt/SUNWwbsvr`
- Linux: `/opt/sun/webserver`

Overview of Web Server Upgrades

This section describes the following general aspects of Web Server that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Web Server

Java ES Release 4 versions of Web Server represents a number of bug fixes, including security fixes that depend upon the NSS shared component.

For details, see the appropriate release notes.

Web Server Upgrade Roadmap

Table 6-1 shows the supported Web Server upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 6-1 Upgrade Paths to Java ES Release 4: Sun java System Web Server 6.1 SP5 2005Q4

Java ES Release	Web Server Version	General Approach	Re-configuration Required
Release 3	Sun Java System Web Server 6 2005Q1 Update 1 SP 4	Direct upgrade: Performed by applying patches.	None
Release 2	Sun Java System Web Server 6 2004Q2 Update 1 SP 2 Platform and Enterprise Editions	Direct upgrade: Performed by applying patches.	None
Release 1	Sun ONE Web Server 6.1 (2003Q4)	Direct upgrade not certified: But can be performed by applying patches.	None
Pre-dates Java ES releases		No direct upgrade.	

Web Server Data

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

Table 6-2 Web Server Data Usage

Type of Data	Location	Usage
Configuration data	*WebServer-base*/https-*instanceName*/config/obj.conf and other files in the same directory	Configuration of Web Server instance

Compatibility Issues

Java ES Release 4 Web Server does not introduce any interface changes and is backwardly compatible with earlier versions.

Web Server Dependencies

Web Server has no dependencies on other Java ES components other than on Java ES shared components (see Table 1-6 on page 40).

Upgrading Web Server from Java ES Release 3

This section includes information about upgrading Web Server from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Web Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. No re-configuration of Web Server is required in upgrading from Java ES Release 3 Web Server to Release 4.
- **Upgrade Dependencies.** While Web Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Web Server requires only that NSS and NSPR be upgraded to Release 4. Upgrade of other shared components is optional with respect to upgrade of Web Server to Release 4.
- **Backward Compatibility.** Release 4 Web Server is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade to Release 3 is achieved by removing the patches applied during the upgrade.
- **Platform Issues.** The general approach for upgrading Web Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Web Server Upgrade

This section describes how to perform an upgrade of Web Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Web Server by starting the Web Server instance server with the `-version` option:

WebServer-base`/https-`*hostName*`.`*domainName*`/start -version`

Table 6-3 Web Server Version Verification Outputs

Java ES Release	Web Server Version Number
Release 2	6.1SP2
Release 3	6.1SP4
Release 4	6.1SP5

Upgrade Web Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, the upgrade of Web Server to Release 4 requires only that the NSS and NSPR shared component be upgraded from Release 3 to its Release 4 version before upgrading Web Server. Instructions for upgrading NSS and NSPR to Release 4, or other Java ES shared components you might wish to upgrade, are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.

Back Up Web Server Data

The Web Server upgrade from Release 3 to Release 4 does not modify configuration data. There is no need to back up current data.

Obtain Required Configuration Information and Passwords

No special information about your currently installed version is needed. However you will have to log in as superuser to perform the upgrade.

Upgrading Release 3 Web Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Web Server followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Web Server software to Java ES Release 4 takes into account the following considerations:

- 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 when patches are being applied to the installed image.

- The Release 4 Web Server upgrade patches for Solaris OS are shown in the following table:

Table 6-4 Patches[1] to Upgrade Web Server on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Web Server core (SUNWwbsvr)	116648-17	116649-17
Web Server localization	117514-10	117515-10

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to all Web Server instances corresponding to the same installed Web Server image on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 6-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

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

   WebServer-base / https-*instanceName* / stop
   WebServer-base / https-admserv / stop

4. If you have not already done so, upgrade the NSS and NSPR shared component to Release 4 and any other shared components you wish to upgrade.

   See “Upgrade Web Server Dependencies” on page 142.

5. Apply the appropriate Web Server patches in Table 6-4.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Restart the Web Server instances that were stopped in Step 3.

Upgrading Release 3 Web Server (Linux)

This section discusses considerations that impact the upgrade procedure for Web Server followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Web Server software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 142), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Web Server upgrade patches for Linux OS are shown in the following table:

Table 6-5 Patches[1] to Upgrade Web Server on Linux

Description	Patch ID and RPM names
Web Server core	118202-09 • `sun-webserver-6.1.5-6.i386.rpm`
Web Server localization	118203-06 • `sun-webserver-`*Locale*`-6.1.5-1.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table

Upgrade Procedure (Linux)

The procedure documented below applies to all Web Server instances corresponding to the same installed Web Server image on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 6-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

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

   WebServer-base / https-*instanceName* / stop
   WebServer-base / https-admserv / stop

4. If you have not already done so, upgrade the NSS and NSPR shared component to Release 4 and any other shared components you wish to upgrade.

   See "Upgrade Web Server Dependencies" on page 142.

5. Apply the RPMs for Web Server in Table 6-5.

   ```
   rpm -Fvh sun-webserver-6.1.5-6.i386.rpm
   ```

6. Confirm that the upgrade was successful:

   ```
   rpm -q sun-webserver
   ```

   The new revision number of the RPM should be returned.

7. Restart the Web Server instances that were stopped in Step 3.

   WebServer-base / https-admserv / start
   WebServer-base / https-*instanceName* / start

Verifying the Upgrade

You can verify the upgrade of Web Server to Release 4 by starting the Web Server instance server with the `-version` option:

WebServer-base`/https-`*hostName*`.`*domainName*`/start -version`

See Table 6-3 on page 141 for output values.

Also, you can check the entries in the following log file:

WebServer-base`/setup/upgrade.log`

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 143 and "Upgrade Procedure (Linux)" on page 145.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Web Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Web Server is pretty much the reverse of the procedure for upgrading to Release 4. The patches are removed.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   `su -`

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

   WebServer-base / https-*instancename* / stop
   WebServer-base / https-admserv / stop

3. Remove the patches in Table 6-4 on page 143.

   `patchrm` *patch_ID*

4. Restart the Web Server instances that were stopped in Step 2.

Upgrading Web Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Web Server to Release 4 is the same as that for upgrading Release 3 Web Server to Release 4, with the exception that the pre-upgrade tasks should include the upgrading of all shared components upon which Web Server depends (see Table 1-6 on page 40) from their Release 2 versions to Release 4.

Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51.

To upgrade Release 2 Web Server to Release 4, use the instructions in “Upgrading Web Server from Java ES Release 3” on page 140, except substitute Release 2 wherever Release 3 is referenced. The upgrade from Release 2 to Release 4, however, also requires modification of the `obj.conf` configuration file, but this is performed automatically.

Chapter 7

Message Queue

This chapter describes how to upgrade Message Queue software from previous Java ES versions to Java ES 2005 (Release 4): Sun Java System Message Queue 3 Enterprise Edition 2005Q4.

The chapter provides a general overview of Message Queue upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

NOTE Message Queue commands used in this chapter are run with respect to the directory location of executable files, which depends on operating system platform:

- Solaris: `/usr/bin`
- Linux: `/opt/sun/mq/bin`

Overview of Message Queue Upgrades

This section describes the following general aspects of Message Queue that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Message Queue

Java ES Release 4 Message Queue represents minor code fixes with no new features or enhancements. As such, Release 4 does not introduce any new compatibility issues (see "Compatibility Issues" on page 152).

Message Queue software includes two editions, a Platform Edition and an Enterprise Edition, each corresponding to a different feature set and licensed capacity. Enterprise edition is for deploying and running messaging applications in an enterprise production environment. Platform Edition is mainly for developing, debugging, and load testing messaging applications and components. The Platform Edition can be downloaded free from the Sun website and is also bundled with the Solaris OS and with the Java ES Application Server platform. An upgrade from an earlier Java ES release version to Release 4 converts any installed Platform Edition to Enterprise Edition.

Message Queue Upgrade Roadmap

Table 7-1 shows the supported Message Queue upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 7-1 Upgrade Paths to Java ES Release 4 Message Queue 3.6 SP3 2005Q4

Java ES Release	Message Queue Version	General Approach	Re-configuration Required
Release 3	Sun Java System Message Queue 2005Q2 (3.6) Enterprise Edition only	Direct upgrade: Performed using the `mqupgrade` script.	None
Release 2	Sun Java System Message Queue 2004Q2 (3.5) Platform and Enterprise Editions	Direct upgrade: Performed using the `mqupgrade` script.	Performed automatically on Solaris platforms, and an `mqmigrate` script is available on Linux platforms.
Release 1	Sun Java System Message Queue 3.01 SP2 Platform and Enterprise Editions	Direct upgrade not certified: But can be performed using the `mqupgrade` script.	Performed automatically on Solaris platforms, and an `mqmigrate` script is available on Linux platforms.
Pre-dates Java ES releases	Sun Java System Message Queue 3.01 SP1 and earlier versions Platform and Enterprise Editions	No direct upgrade: But you can upgrade first to Release 3 using procedures in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`). Then upgrade from Release 3 to Release 4.	

In addition to the Java ES releases of Message Queue shown in Table 7-1, Message Queue Platform Edition is also bundled with Solaris operating system software. Upgrade of the bundled versions of Message Queue to Release 4 Enterprise Edition can be performed by the Java ES installer. You simply select Message Queue for installation by the installer, as in a new install, and the installer software will automatically upgrade the bundled version, performing any re-configuration of Message Queue that might be necessary.

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

Table 7-2 shows the location of data on Solaris systems. The location on Linux systems is similar, and can be found in the *Message Queue Administration Guide* (`http://docs.sun.com/doc/819-2571`). In Table 7-2, *instanceName* identifies the name of the Message Queue broker instance with which the data is associated.

Table 7-2 Message Queue Data Usage

Data Category	Location (on Solaris)	Usage
Broker instance configuration properties	`/var/imq/instances/`*instanceName*`/props/config.properties`	Broker and related services configurations
Persistent store for dynamic application data	`/var/imq/instances/`*instanceName*`/fs350/` or a JDBC-accessible data store	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	`/var/imq/instances/`*instanceName*`/etc/passwd` or an LDAP directory server	Stores user data used for authentication and authorization
Security: access control file (default location)	`/var/imq/instances/`*instanceName*`/etc/accesscontrol.properties`	Sets the rules that authorize user access to destinations and related capabilities
Security: passfile directory (default location)	`/var/imq/instances/`*instanceName*`/etc/`	Stores encrypted password information.
Security: broker's keystore file location	`/etc/imq/`	Stores encrypted certificate information for secure messaging.

Compatibility Issues

Release 4 Message Queue introduces no new incompatibilities over Release 3. The following general Message Queue compatibility issues relate to versions earlier than Release 3.

Protocol Compatibility

Message Queue has a dependency on a web container to provide HTTP protocol support between Message Queue clients and broker. Due to a protocol change, when using Sun Java System Web Server to provide a web container for the Message Queue `imqhttp.war` application, you cannot upgrade the Web Server component without also upgrading Message Queue (see "Post-Upgrade Tasks" on page 160 and page 165.

Broker Compatibility

A Release 4 Message Queue broker will inter-operate with a Release 3 or Release 2 broker, however changes in broker properties and the persistent store schema with respect to Release 2 can impact compatibility.

Release 4 Message Queue can use Release 3 and Release 2 data, except that on Linux systems, Release 2 data must be first migrated to Release 4.

When updating to Release 4 Message Queue, consider the following:

- You can use earlier Message Queue `config.properties` files. You can also copy them to another location and consult the property settings they contain when you configure Release 4 Message Queue brokers.
- Any persistent Message Queue data—messages, destinations, durable subscriptions—is automatically converted, if necessary, to Release 4 Message Queue data when starting up a broker for the first time. For example, any existing destinations will be converted, if necessary, to Release 4 Message Queue destinations, preserving existing attributes and using default values of new attributes.
- If you mix Message Queue Release 2 brokers and Message Queue Release 4 brokers in a cluster, the master broker must be a Message Queue Release 2 broker (whichever is older), and the cluster will run as a Message Queue Release 2 cluster.

Administered Object Compatibility

Release 4 Message Queue administered objects are identical to Release 3 administered objects. However, some Release 3 administered objects were renamed or enhanced with new attributes with respect to earlier versions. Therefore, when upgrading from Release 2 Message Queue to Release 4, you should consider the following:

- You can use the same object store and administered objects that you created in Release 2; however, it is best to migrate your administered objects to Release 4. The Administration Console (`imqadmin`) and the ObjectManager command line utility (`imqobjmgr`), when performing an update operation, will convert Release 2 administered objects into Release 4 administered objects.
- The Release 4 client runtime will look up and instantiate Release 2 administered objects and convert them for use by Release 4 clients. However, this will *not* convert Release 2 administered objects residing in the object store from which the lookup was made.
- Existing Release 2 clients (applications and / or components)—that is, clients that directly instantiate administered objects rather than look them up—are compatible with Release 4. However, if they are to use the *new* administered object attributes (see Chapter 16 of the *Message Queue Administration Guide* (`http://docs.sun.com/doc/819-2571`) for information on administered object attributes), they will need to be rewritten. (Re-compiling Release 2 clients with Release 4 will show which Message Queue Release 2 attributes have been renamed in Release 4. The old names will still work.)
- Scripts that start Java clients and which set administered object attribute values using command line options are compatible with Release 4. However, if they are to use the *new* administered object attributes (see Chapter 16 of the *Message Queue Administration Guide (*`http://docs.sun.com/doc/819-2571`) for information on administered object attributes), they will need to be rewritten.

Administration Tool Compatibility

Because of the addition of new commands and new administrative capabilities in Release 3, the Release 4 administration tools (the Administration Console and command line utilities) only work with Release 3 and Release 4 brokers. However, all Release 2 commands and command options remain supported.

Client Compatibility

Release 3 clients are completely compatible with Release 4 Message Queue. When upgrading from Release 2 to Release 4, however, you should consider the following compatibility issues, regarding Java clients:

- A Release 4 broker will support a Release 2 client (but without additional Release 4 capabilities).
- A Release 4 Java client can connect to a Release 2 broker (but without additional Release 4 capabilities).
- C client programs are supported only by Release 2, Release 3, or Release 4 brokers running with a trial license (Platform Edition) or Enterprise Edition license.

Message Queue Dependencies

Message Queue dependencies on other Java ES components can impact the procedure for upgrading and re-configuring 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 (see Table 1-6 on page 40).
- **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, then 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, Java ES Application Server, or third-party web containers.

Upgrading Message Queue from Java ES Release 3

This section includes information about upgrading Message Queue from Java ES 2005Q1 (Release 3) to Java ES Release 4. The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Message Queue to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed using an `mqupgrade` script that replaces previous software packages with new ones and migrates configuration data from Release 3 automatically.
- **Upgrade Dependencies.** While Message Queue has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Message Queue is compatible with the Release 3 versions of all these components. Upgrade of these shared components is therefore optional with respect to upgrade of Message Queue to Release 4.

  In addition, Release 4 Message Queue is optionally dependent on Directory Server and Web Server (or Application Server), as described in "Message Queue Dependencies" on page 155. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Message Queue to Release 4.
- **Backward Compatibility.** Release 4 Message Queue is fully compatible with Release 3 (see "Compatibility Issues" on page 152).
- **Upgrade Rollback.** There is no utility for rolling back the Message Queue upgrade to Release 3. You have to remove the upgraded components and manually restore the previous version and configuration data.
- **Platform Issues.** The general approach for upgrading Message Queue is the same on both Solaris and Linux operating systems. The procedures that follow indicate platform-specific commands or file locations where appropriate.

Release 3 Message Queue Upgrade

This section describes how to perform a Message Queue upgrade from Java ES Release 3 to Java ES Release 4:

Pre-Upgrade Tasks

Before you upgrade Message Queue, perform the procedures described in the following sections. Where the procedure depends on platform-specific commands, the task will indicate the operating system to which it applies.

Verify Current Version Information (Solaris Systems)

You can determine the version and edition of Message Queue installed on your system by starting the Message Queue broker with the `-version` option:

```
imqbrokerd -version
```

Table 7-3 Message Queue Version Verification Outputs

Java ES Release	Message Queue Version Number
Release 2	`Sun Java(tm) System Message Queue 3 2004Q2` `Version:  3.5`
Release 3	`Sun Java(tm) System Message Queue 3 2005Q1` `Version:  3.6`
Release 4	`Sun Java(tm) System Message Queue 3 2005Q4` `Version:  3.6 SP3`

Upgrade Message Queue Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, because Message Queue does not require upgrading the Java ES Release 3 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 that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51).
2. **Directory Server (optional).** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, “Directory Server and Administration Server” on page 103.
3. **Web Container Software (optional).** Instructions for upgrading Web Server or Application Server are provided in Chapter 6, “Web Server” on page 137 and Chapter 9, “Application Server” on page 175, respectively.

Back Up Message Queue

There is no script for rolling back Message Queue to its pervious state. Because Release 4 data is compatible with Release 3 data, there is no reason to back up configuration data. In addition, there is no reason to back up the installed image because you can use the Release 3 installer should you need to roll back Release 4 Message Queue to Release 3.

Upgrading Release 3 Message Queue

The upgrade of Message Queue software to Java ES Release 4 makes use of the `mqupgrade` script, which installs freshbitted packages that contain the patches shown in Table 7-4.

Table 7-4 Patches[1] to Upgrade Message Queue

Component	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10	Linux
Message Queue Core	119132-06	119133-06	119136-06
Message Queue C-runtime	119134-04	119135-04	
Message Queue jmsclient & xmlclient			119137-04
Message Queue localization	119691-03	119692-03	119693-03

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

The upgrade procedure consists of the following steps:

1. Stop any running Message Queue client applications.

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

2. Stop any running brokers. You will be prompted for the admin user name and password.

   `imqcmd shutdown bkr [-b` *hostName*`:`*port*`]`

3. If you do not want to preserve dynamic data, the Message Queue flat-file user repository, and the Message Queue access control file associated with each broker instance, remove this data using the following command.

   `imqbrokerd -name` *instanceName* `-remove instance`

   Otherwise, dynamic data and configuration information will be retained and used for Release 4 Message Queue.

4. Log in as Root.

   `su -`

5. Change directories to the location of the `Tools` directory of the Java ES distribution.

   On Solaris SPARC:
   `cd Solaris_sparc/Product/message_queue/Tools`

   On Solaris x86:
   `cd Solaris_x86/Product/message_queue/Tools`

   On Linux x86:
   `cd Linux_x86/Product/message_queue/Tools`

6. Run the `mqupgrade` script.

   a. Start the script:

      `./mqupgrade`

      The `mqupgrade` script lists installed Message Queue components.

   b. Enter `y` (yes) to upgrade Message Queue components.

      The `mqupgrade` script detects and lists installed localization files.

      If you do not want to upgrade Message Queue components, enter `n` (no). The `mqupgrade` script will exit without upgrading Message Queue components.

c. If prompted, enter `y` (yes) to upgrade localization files.

The `mqupgrade` script sends output to a log file in the following location:

`/var/sadm/install/logs/Message_Queue_upgrade_'`*date*`'.log`

Verifying the Message Queue Upgrade

After you finish the upgrade procedure, verify that it was successful by starting the Message Queue broker with the `-version` option.

```
imqbrokerd -version
```

The command returns the Java ES version number as well as the Message Queue-specific version number.

Post-Upgrade Tasks

If you have upgraded the web container and are using the Message Queue HTTP tunneling servelet, you may need to re-deploy it in the new web container. Otherwise, there has been no change to the HTTP tunneling servlet between Release 3 and Release 4, and you do not need to re-deploy it after upgrading Message Queue to Release 4. See the *Message Queue Administration Guide*, (`http://docs.sun.com/doc/819-2571`) for more information on HTTP support.

Rolling Back the Upgrade

No scripts are provided for rolling back Message Queue to its pre-upgrade state. The process must be performed manually using the following steps:

1. Stop any running Message Queue client applications.
2. Stop any running brokers. You will be prompted for the admin user name and password.

   `imqcmd shutdown bkr [-b` *hostName*`:`*port*`]`
3. If you want to delete dynamic data, the Message Queue flat-file user repository, and the Message Queue access control file associated with each broker instance, remove this data using the following command.

   `imqbrokerd -name` *instanceName* `-remove instance`

4. Log in as root or become superuser.

   ```
   su -
   ```

5. Retrieve the list of installed Message Queue packages with the following command:

   Solaris:
   ```
   pkginfo | grep -i "message queue"
   ```

   Linux:
   ```
   rpm -qa | grep mq
   ```

6. Remove the Message Queue packages, using the following command:

   Solaris:
   `pkgrm` *packageName*
   where *packageName* is any of the Message Queue packages. To remove multiple packages, separate the package names by a space.

   Linux:
   `rpm -e --nodeps` *RPMName*
   where *RPMName* is any of the Message Queue rpm components. To remove multiple components, separate the RPM names by a space.

   Because other products might be using Message Queue packages, be careful about removing them. The `pkgrm` command will warn you of any dependencies on a package before removing it. When prompted, confirm your removal request by typing **y** (yes).

7. Type “q” to quit.

8. Exit the root shell.

9. Re-install Release 3 Message Queue.

   Use the Java ES Release 3 installer. Release 4 Message Queue data will work fine.

Multiple Instance Upgrades

To upgrade a Message Queue cluster, in which multiple brokers interact to provide a scalable message service, you can do a rolling upgrade in which the cluster remains online as each Message Queue instance is upgraded from Release 3 to Release 4. The two conditions to keep in mind when performing a cluster upgrade are:

- While a broker is shut down for upgrade, the persistent messages it is storing are not available until the broker is restarted.
- The Master broker should be upgraded last.

Otherwise the procedure is straightforward: you shut down, upgrade, and restart the brokers one at a time until all have been upgraded.

Upgrading Message Queue from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Message Queue to Release 4 is nearly the same as that for upgrading Release 3 Message Queue to Release 4 (see "Upgrading Message Queue from Java ES Release 3" on page 156). For upgrading from Release 2, however, there is a small difference between operating system platforms.

In addition, the pre-upgrade tasks should include the upgrading of all shared components upon which Message Queue depends (see Table 1-6 on page 40) from their Release 2 versions to Release 4.

Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.

Upgrading Release 2 Message Queue (Solaris)

Use the instructions in "Upgrading Message Queue from Java ES Release 3" on page 156, except substitute Release 2 wherever Release 3 is referenced.

Upgrading Release 2 Message Queue (Linux)

On Linux systems, an upgrade from Release 2 to Release 4 includes a data migration step that is not needed in updating from Release 3 to Release 4, namely the migration of broker instance data to the appropriate Release 4 location. If you want to preserve your Release 2 data in upgrading to Release 4, Message Queue provides a migration tool, `mqmigrate`, to perform this migration.

Upgrade Procedure

To upgrade from Release 2 to Release 3, you use the same instructions as used in "Upgrading Message Queue from Java ES Release 3" on page 156, except you run the `mqmigrate` script before you run the `mqupgrade` script, as detailed in the following procedure.

1. Stop any running Message Queue client applications.
2. Stop any running brokers. You will be prompted for the admin user name and password.

   `imqcmd shutdown bkr [-b` *hostName*`:`*port*`]`
3. If you do not want to preserve dynamic data, the Message Queue flat-file user repository, and the Message Queue access control file associated with each broker instance, remove this data using the following command.

   `imqbrokerd -name` *instanceName* `-remove instance`

   Otherwise, dynamic data and configuration information will be retained and used for Release 4 Message Queue.
4. Log in as root or become superuser.

   `su -`
5. Change directories to the location `Tools` directory of the Java ES distribution.

   `cd Linux_x86/Product/message_queue/Tools`
6. Migrate broker instance data using the following command:

   `./mqmigrate`

   The `mqmigrate` script will move Release 2 broker instance configuration data to the appropriate R4 location.

7. Run the `mqupgrade` script.

   a. Start the script:

      ```
      ./mqupgrade
      ```

      The `mqupgrade` script lists installed Message Queue components.

   b. Enter `y` (yes) to upgrade Message Queue components.

      The `mqupgrade` script detects and lists installed localization files.

      If you do not want to upgrade Message Queue components, enter `n` (no). The `mqupgrade` script will exit without upgrading Message Queue components.

   c. If prompted, enter `y` (yes) to upgrade localization files.

      The `mqupgrade` script sends output to a log file in the following location:

      `/var/sadm/install/logs/Message_Queue_upgrade_'`*`date`*`'.log`

Installing the Compatibility Package

If you have scripts or your Release 2 client applications contain scripts that depend on the location of Release 4 installed files, you will need to install the `sun-mq-compat` package, which contains symlinks from Release 2 file locations to Release 4 file locations.

The `sun-mq-compat` package is in the following location where you unzipped the Java ES distribution.

```
Linux_x86/Product/message_queue/Packages
```

Perform the following steps to Install the `sun-mq-compat` Package:

1. Log in as root or become superuser.

   ```
   su -
   ```

2. From the Packages directory, enter the following command:

   `rpm -ivh --nodeps sun-mq-compat-3.6-`*`RelNo`*`.i386.rpm`

Post-Upgrade Tasks

If you are using the HTTP tunneling servlet to provide HTTP connection service support, the upgrade of Message Queue from Release 2 to Release 4 has upgraded the servlet. This requires you to re-deploy it after upgrading Message Queue to Release 4. See the *Message Queue Administration Guide*, *(*http://docs.sun.com/doc/819-2571) for more information on HTTP support.

Migrate Release 2 administered objects to their Release 4 versions using the Administration Console (`imqadmin`) and / or the ObjectManager command line utility (`imqobjmgr`) to perform an update operation.

Chapter 8

High Availability Session Store

This chapter describes how to upgrade High Availability Session Store to Java ES 2005Q4 (Release 4): High Availability Session Store (HADB) 4.4.2.

The chapter provides a general overview of upgrade issues before covering the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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

The value of *HADB-base* is in relation to the Application Server directory structure, as follows:

AppServer8-base/hadb/*version_number*

The default value of *HADB-base* depends on the default value of *AppServer8-base*, which depends on operating system platform:

- Solaris: /opt/SUNWappserver/appserver/hadb/*version_number*
- Linux: /opt/sun/appserver/hadb/*version_number*

Overview of HADB Upgrades

This section describes the following general aspects of HADB that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 HADB

Java ES Release 4 versions of HADB represents bug fixes to the Java ES 2005Q1 (Release 3) version.

HADB Upgrade Roadmap

Table 8-1 shows the supported HADB upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 8-1 Upgrade Paths to Java ES Release 4: HADB 4.4.2 (2005Q4)

Java ES Release	HADB Version	General Approach	Re-configuration Required
Release 3	HADB 4.4.1 (2005Q1)	Direct upgrade: Online and Offline upgrades are both available.	None
Release 2	HADB 4.4.0-14 (2004Q2)	Upgrade not supported.	None
Release 1	Not available	No upgrade	None
Pre-dates Java ES releases	Not available	No upgrade.	None

HADB Data

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

Table 8-2 HADB Data Usage

Type of Data	Location	Usage
Dynamic application data	`/var/opt/SUNWhadb` and `/etc/opt/SUNWhadb`	High availability session store and configuration information.

Compatibility Issues

HADB provided with Java ES Release 4 is backwardly compatible with HADB provided with Java ES Release 3.

HADB Dependencies

HADB provided with Java ES Release 4 requires Java™ 2 Platform, Standard Edition (J2SE™) Version 5.0 or later.

Upgrading HADB from Java ES Release 3

This section includes information about upgrading HADB from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 HADB to Java ES Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** Upgrades consist of removing the Java ES Release 3 HADB packages and adding the Java ES Release 4 packages. There are two upgrades approaches available:
  - **Online upgrade.** Use online upgrade to avoid interruption of HADB services.
  - **Offline upgrade.** Use offline upgrade if you can interrupt HADB services when replacing HADB packages with newer versions.
- **Upgrade Dependencies.** HADB requires J2SE Version 5.0 or later.
- **Backward Compatibility.** HADB provided with Java ES Release 4 is backwardly compatible with HADB provided with Java ES Release 3.
- **Upgrade Rollback.** Rollback from the Java ES Release 4 upgrade to Java ES Release 3 is achieved by restoring the Release 3 version, which is left undisturbed in a separate directory by the upgrade to Release 4.
- **Platform Issues.** The general approach for upgrading HADB is the same on both Solaris and Linux operating systems.

Release 3 HADB Upgrade

This section describes how to perform an upgrade of HADB from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade HADB you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of HADB using standard version checking utilities. For example:

Solaris:

```
pkgparam -v SUNWhadba
```

Linux:

```
rpm -qi sun-hadb-a-4.4.2-7.i386.rpm
```

Table 8-3 HADB Version Verification Outputs

Java ES Release	HADB Version Number
Release 2	VERSION=4.4.0,REV=14
	SUNW_PRODVERS=4.4.0
Release 3	VERSION=4.4.1,REV=7
	SUNW_PRODVERS=4.4.1
Release 4	VERSION=4.4.2,REV=7
	SUNW_PRODVERS=4.4.2

Upgrade HADB Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. The upgrade of HADB to Release 4 depends on J2SE 5.0 or later.

Back Up Directory Data

The HADB upgrade from Java ES Release 3 to Java ES Release 4 does not in itself modify HADB dynamic data. However, you can back up the Java ES Release 3 packages in case you need to roll back the upgrade.

Obtain Required Configuration Information and Passwords

HADB upgrade requires you to know the superuser password.

Upgrading Release 3 HADB

This section discusses considerations that impact the upgrade procedure for HADB followed by a description of the procedure itself.

Upgrade Considerations

The upgrade of HADB software to Java ES Release 4 takes into account the following considerations:

- Based on your production requirements, you need to determine whether an online or offline upgrade is more appropriate.
- The Java ES Release 4 upgrade packages for Solaris and Linux platforms are shown in the following table. Solaris packages are listed in their installation sequence.

Table 8-4 Package Versions for Upgrading HADB on Solaris Platforms

Solaris Packages	Linux Packages
SUNWhadba	sun-hadb-a-4.4.2-7.i386.rpm
SUNWhadbc	sun-hadb-c-4.4.2-7.i386.rpm
SUNWhadbe	sun-hadb-e-4.4.2-7.i386.rpm
SUNWhadbi	sun-hadb-i-4.4.2-7.i386.rpm
SUNWhadbj	sun-hadb-j-4.4.2-7.i386.rpm
SUNWhadbm	sun-hadb-m-4.4.2-7.i386.rpm
SUNWhadbo	sun-hadb-o-4.4.2-7.i386.rpm
SUNWhadbs	sun-hadb-s-4.4.2-7.i386.rpm
SUNWhadbv	sun-hadb-v-4.4.2-7.i386.rpm
SUNWhadbx	sun-hadb-x-4.4.2-7.i386.rpm

Online Upgrades of HADB

Online upgrades of HADB are only available when upgrading from Java ES Release 3.

When you perform an online upgrade of HADB, you first install HADB on each server in the cluster being upgraded. Each server first unregisters from an earlier installation of HADB and then registers with the newly installed version of HADB.

For details on performing an online upgrade, refer to the following section in *Sun Java System Application Server Enterprise Edition 8.1 2005Q2 High Availability Administration Guide (*http://docs.sun.com/doc/819-2555/6n4r9qo7n?a=view)

Offline Upgrades of HADB

An offline upgrade of HADB is available when upgrading from either Java ES Release 3.

To perform an offline upgrade, shut down your HADB services and replace the existing HADB packages with the newer versions available from your Java ES Release 4 distribution, shown in Table 8-4 on page 172.

Use the procedures documented in "Upgrading Packages on Solaris Platforms" on page 70 and "Upgrading Packages on Linux Platforms" on page 74.

Verifying the Upgrade

After completing the online upgrade, verify the upgrade by using the following procedure. After verifying that the upgrade is successful, the old installation packages can be deleted.

To verify that running processes are using the upgraded HADB services, you can perform the following steps.

1. For all HADB services running, issue either of the following commands:

HADB-base/bin/ma -V
HADB-base/bin/hadbm -V

For example, on the Solaris 8 platform:

HADB-base/bin/ma -V
Sun Java System High Availability Database 4.4 Database Management Agent
Version : 4.4.2.7 [V4-4-2-7 2005-05-26 13:49:01 *server@domain*] \
(SunOS_5.8_sparc)

2. Check whether the database is running by issuing the commands in the following example for a database named ExampleDB:

   HADB-base/bin/hadbm status -n *databaseName*

   ```
   HADB-base/bin/hadbm list
   Database
   ExampleDB

   HADB-base/bin/hadbm status ExampleDB
   Database    Status
   ExampleDB    FaultTolerant

   HADB-base/bin/hadbm status -n ExampleDB
   NodeNo    HostName     Port        NodeRole   NodeState  MirrorNode
   0         sungod012    15000       active     running      1
   1         sungod012    15020       active     running      0
   ```

   All HADB services for listed nodes should in the “running” state.

3. Verify that all products using HADB are using the new HADB path by issuing the command in following example for a database named ExampleDB:

   ```
   HADB-base/bin/hadbm get PackageName ExampleDB
   Attribute   Value
   PackageName V4.4.2.7
   ```

   The above command displays the current version of HADB. For a detailed listing, issue the following command:

   ```
   hadbm get --all ExampleDB
   ```

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrading HADB from Java ES Release 3” on page 170.

Rolling Back the Upgrade

To roll back the upgrade to HADB, replace the newer versions of the HADB packages you installed with the versions you previously had installed. Use the same procedure described in “Rolling Back the Upgrade” on page 174. There is no data or configuration files that need to be changed.

Chapter 9

Application Server

This chapter describes how to upgrade Application Server to Java ES 2005Q4 (Release 4): Sun Java System Application Server Enterprise Edition 8.1 2005Q4.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

NOTE File locations in this chapter are specified with respect to directory paths referred to as *AppServer8-base* and *AppServer8Config-base* (Application Server 8.1), *AppServer7-base* and *AppServer7Config-base* (Application Server 7). At least part of these paths might have been specified as installation directories when Application Server was installed. If not, the Java ES installer assigned a default value.

The default values of these directory paths are shown in the following table.

Path Name	Solaris OS	Linux OS
AppServer8-base	`/opt/SUNWappserver/appserver`	`/opt/sun/appserver`
AppServer8Config-base	`/var/opt/SUNWappserver`	`/var/opt/sun/appserver`
AppServer7-base	`/opt/SUNWappserver7`	`/opt/SUNWappserver7`
AppServer7Config-base	`/var/opt/SUNWappserver7`	`/var/opt/SUNWappserver7`

Overview of Application Server Upgrades

This section describes the following general aspects of Application Server that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Application Server

Java ES Release 4 Application Server represents selected bug fixes to the Release 3 version. Functionally Release 4 Application Server is the same as Release 3.

Application Server Upgrade Roadmap

Table 9-1 shows the supported Application Server upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 9-1 Upgrade Paths to Java ES Release 4: Sun Java System Application Server Enterprise Edition 8.1 2005Q4

Java ES Release	Application Server Version	General Approach	Re-configuration Required
Release 3	Sun Java System Application Server Enterprise Edition 8.1 2005Q1	Direct upgrade: Performed by applying patches.	None
Release 2	Sun Java System Application Server 7.0 Upgrade 3 (2004Q2) Platform and Enterprise Editions	Direct upgrade: Use the Java ES installer then the re-configuration utility.	Environment variables and other configuration data. J2EE components and applications need to be migrated to new Application Server environment and redeployed.

Table 9-1 Upgrade Paths to Java ES Release 4: Sun Java System Application Server Enterprise Edition 8.1 2005Q4 *(Continued)*

Java ES Release	Application Server Version	General Approach	Re-configuration Required
Release 1	Sun ONE Application Server 7.0 Upgrade 1 (2003Q4) Platform and Enterprise Editions	Direct upgrade not certified: But you can use the Java ES installer then the re-configuration utility.	Environment variables and other configuration data. J2EE components and applications need to be migrated to new Application Server environment and redeployed.
Pre-dates Java ES releases		No direct upgrade: But you can upgrade first to Release 3 using procedures in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062). Then upgrade from Release 3 to Release 4.	

In addition to the Java ES releases of Application Server shown in Table 9-1, Application Server Platform Edition is also bundled with Solaris operating system software. Upgrade of the bundled versions of Application Server to Release 4 Enterprise Edition can be performed by the Java ES installer. You simply select Application Server for installation by the installer, as in a new install, and the installer software will automatically upgrade the bundled version, performing any re-configuration of Application Server that might be necessary.

Application Server Data

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

Table 9-2 Application Server Data Usage

Type of Data	Location	Usage
Environment variables	*AppServer8-base*/config/asenv	Global variables
Configuration data	Release 3 & Release 4: domain.xml and server.policy files in *AppServer8Config-base*/domains/*domainName*/config Release 2: server.xml and server.policy files in *AppServer7Config-base*/domains/*domainName*/*instanceName*/config	Configuration of Application Server instances
Deployment data	Release 3 & Release 4: *AppServer8Config-base*/domains/*domainName*/applications Release 2: *AppServer7Config-base*/domains/*domainName*/*instanceName*/applications	Configuration of J2EE container for specific J2EE components and applications.

Compatibility Issues

Release 4 Application Server does not introduce any interface changes with respect to Release 3. However, there are major interface changes between Release 4 and Release 2, making Release 4 incompatible with Release 2.

Application Server Dependencies

Application Server dependencies on other Java ES components can impact the procedure for upgrading and re-configuring 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 (see Table 1-6 on page 40).
- **Message Queue.** Application Server depends on Message Queue to provide J2EE Java Message Service-compliant asynchronous messaging support.
- **Web Container (optional).** Application Server depends upon web container services for its optional load balancing plugin. This support can be provided either by Java ES Web Server or third-party web containers (such as Apache Web Server, and Microsoft IIS).
- **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 Release 3

This section includes information about upgrading Application Server from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Application Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. No re-configuration or migration of J2EE components is required to upgrade from Release 3 Application Server to Release 4.
- **Upgrade Dependencies.** While Application Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Application Server is compatible with the Release 3 versions of all these components. Upgrade of these shared components is therefore optional with respect to upgrade of Application Server to Release 4.

  In addition, Release 4 Application Server is dependent upon Release 4 Message Queue and optionally dependent on Java ES Web Server or third-party web containers, as described in “Application Server Dependencies” on page 179. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Application Server to Release 4.

  Release 4 Application Server is also optionally dependent upon on High Availability Session Store. If being used by Application Server, High Availability Session Store should be upgraded to Release 4. Note that upgrade of High Availability Session Store automatically upgrades the J2SE shared component to Release 4.
- **Backward Compatibility.** Release 4 Application Server is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade to Release 3 is achieved by removing the patches applied during the upgrade.

- **Platform Issues.** The general approach for upgrading Application Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Application Server Upgrade

This section describes how to perform an upgrade of Application Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Application Server by entering the following command:

AppServer8-base`/bin/asadmin version --verbose`

Table 9-3 Application Server Version Verification Outputs

Java ES Release	Application Server Version Number
Release 2	`Sun ONE Application Server 7.0.0_03c`
Release 3	`Sun Java Enterprise System Application Server Enterprise Edition 8.0.0_01`
Release 4	`Sun Java Enterprise System Application Server Enterprise Edition`

Upgrade Application Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, because the upgrade of Application Server to Release 4 does not require upgrading other Release 3 components, this task is optional. If you choose to upgrade components upon which Application Server depends, those components would generally be upgraded in the following order:

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in "Upgrading Java ES Shared Components" on page 51.
2. **Message Queue.** Instructions for upgrading Message Queue to Release 4 are provided in Chapter 7, "Message Queue" on page 149.
3. **Web Container Software (optional).** Instructions for upgrading Web Server or Application Server are provided in Chapter 6, "Web Server" on page 137 and Chapter 9, "Application Server" on page 175, respectively.
4. **High Availability Session Store (optional).** Instructions for upgrading High Availability Session Store are provided in Chapter 8, "High Availability Session Store" on page 167.

Back Up Application Server Data

The Application Server upgrade from Release 3 to Release 4 does not modify configuration data. There is no need to back up current data.

Obtain Required Configuration Information and Passwords

You should know the Application Server administrator user ID and password for your currently installed version.

Upgrading Release 3 Application Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Application Server followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Application Server software to Java ES Release 4 takes into account the following considerations:

- Any J2EE components running in an Application Server instance should be shut down before you upgrade that instance. However, if load balancing provides for high availability or scalability, this requirement can be relaxed.

- All instances of Application Server running on a single computer (all corresponding to the same installed Application Server image) must be shut down while the patch is being applied to the installed image.
- In multiple node deployments, perform the upgrade procedure on each node or computer that hosts Application Server instances.
- The Release 4 Application Server upgrade patch for Solaris OS is shown in the following table:

Table 9-4 Patches[1] to Upgrade Application Server on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Application Server	119166-10	119167-10
Application Server localization	119024-10	119025-10

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Application Server instances residing locally on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 9-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Shut down all J2EE components running in the Application Server instances that are to be upgraded.

3. Shut down all Application Server instances on the computer that is to be upgraded.

   AppServer8-base`/bin/asadmin stop-domain` *domainName*

4. Apply the appropriate Application Server patch in Table 9-4.

   `patchadd` *patch_ID*

5. Modify the `asant` script.

a. Rename the existing `asant` script as `asant.bak`.

The script is at the following location:

AppServer8-base`/bin/asant`

b. Copy the `asant.template` file from

AppServer8-base`/lib/install/templates/ee`

to

AppServer8-base`/bin/asant`

c. Edit the script.

Replace the `%CONFIG_HOME%` token with *AppServer8-base*`/config`.

d. If any manual changes had been made in the original script file (`asant.bak`), merge them into the new `asant` script.

6. Confirm that the patch upgrade was successful:

`showrev -p | grep` *patch_ID*

The output should return the versions of patch IDs applied in Step 4.

7. Restart the Application Server instances.

AppServer8-base`/bin/asadmin start-domain` *domainName*

Upgrading Release 3 Application Server (Linux)

This section discusses considerations that impact the upgrade procedure for Application Server followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Application Server software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 182), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Application Server upgrade patch for Linux OS is shown in the following table:

Table 9-5 Patches[1] to Upgrade Application Server on Linux

Description	Patch ID and RPM names
Application Server	119168-10 sun-asac-8.1.2-10.i386.rpm, sun-asacee-8.1.2-10.i386.rpm sun-ascml-8.1.2-10.i386.rpm, sun-ascmn-8.1.2-10.i386.rpm sun-ascmnse-8.1.2-10.i386.rpm, sun-asdb-8.1.2-10.i386.rpm sun-asdem-8.1.2-10.i386.rpm, sun-asdemdb-8.1.2-10.i386.rpm sun-ashdm-8.1.2-10.i386.rpm sun-asJdbcDrivers-8.1.2-10.i386.rpm sun-asjdoc-8.1.2-10.i386.rpm, sun-aslb-8.1.2-10.i386.rpm sun-asman-8.1.2-10.i386.rpm, sun-asmanee-8.1.2-10.i386.rpm sun-asu-8.1.2-10.i386.rpm, sun-asuee-8.1.2-10.i386.rpm sun-asut-8.1.2-10.i386.rpm, sun-aswbcr-8.1.2-10.i386.rpm
Application Server localization	119026-10 sun-asacee-*Locale*-8.1.1-51.i386.rpm sun-ascmnse-*Locale*-8.1.1-51.i386.rpm sun-asu-*Locale*-8.1.1-51.i386.rpm sun-asuee-*Locale*-8.1.1-51.i386.rpm

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies Application Server instances residing locally on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 9-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Shut down all J2EE components running in the Application Server instances that are to be upgraded.

3. Shut down all Application Server instances on the computer that is to be upgraded.

   AppServer8-base`/bin/asadmin stop-domain` *domainName*

4. Back up the following files:
   - all files under *AppServer8-base*/pointbase/tools/serveroption
   - *AppServer8-base*/samples/common.properties
5. Apply the RPMs for Application Server in Table 9-5.

   ```
   rpm -Fvh sun-as<module>-8.1.2-10.i386.rpm
   rpm -Fvh sun-as<module-Locale>-8.1.1-51.i386.rpm
   ```
6. Restore the files backed up in Step 4 to their original locations:
   - all files under *AppServer8-base*/pointbase/tools/serveroption
   - *AppServer8-base*/samples/common.properties
7. Modify the asant script.
   a. Rename the existing asant script as asant.bak.

      The script is at the following location:

      AppServer8-base/bin/asant

   b. Copy the asant.template file from

      AppServer8-base/lib/install/templates/ee

      to

      AppServer8-base/bin/asant

   c. Edit the script.

      Replace the %CONFIG_HOME% token with *AppServer8-base*/config.

   d. If any manual changes had been made in the original script file (asant.bak), merge them into the new asant script.
8. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-as
   ```

   The new version numbers of the RPMs should be returned.
9. Restart the Application Server instances.

   AppServer8-base/bin/asadmin start-domain *domainName*

Verifying the Upgrade

You can verify that the patch has been properly applied using the following command:

AppServer8-base`/bin/asadmin version --verbose`

See Table 9-3 on page 181 for output values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 183 and "Upgrade Procedure (Linux)" on page 185.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Application Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Application Server is pretty much the reverse of the procedure for upgrading to Release 4. The patches are removed.

Rollback Procedure (Solaris)

1. Shut down all J2EE components running in the Application Server instance that is to be upgraded.
2. Shut down the Application Server instance that is to be upgraded.
3. Remove the patches in Table 9-4.

   `patchrm` *patch_ID*
4. Restart the Application Server instance.

Upgrading Application Server from Java ES Release 2

This section includes information about upgrading Application Server from Java ES Release 2 to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 2 Application Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by installing Release 4 Application Server using the Java ES installer and choosing the configure later option. Re-configuration is subsequently achieved using the `asupgrade` utility. Following the Application Server upgrade you have to migrate Release 2 J2EE components and applications to Release 4.
- **Upgrade Dependencies.** Upgrade of any Java ES component on a computer from Release 2 requires the upgrade of all other Java ES components hosted by the computer; selective upgrade of Java ES components from Release 2 to Release 4 is not supported. In particular, all Java ES shared components used by Application Server need to be upgraded. Message Queue must also be upgraded, if residing on the same computer, and if Web Server is being used for load balancing, it also must be upgraded.

  If being used by Application Server, High Availability Session Store should be upgraded to Release 4. However High Availability Session Store was integrated into Release 2 Application Server and cannot be independently upgraded to Release 4.
- **Backward Compatibility.** Release 4 Application Server is not backwardly compatible with the Release 2 version. J2EE components and applications need to be migrated run in a Release 4 Application Server environment.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade to Release 2 is achieved by simply reverting back to the Release 2 installation (Release 2 configuration data is not removed by the upgrade process).

- **Platform Issues.** The general approach for upgrading Application Server is the same on both Solaris and Linux operating systems.

Release 2 Application Server Upgrade

This section describes how to perform an upgrade of Application Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Application Server by entering the following command:

AppServer7-base`/bin/asadmin version --verbose`

See Table 9-3 on page 181 for version outputs.

Upgrade Application Server Dependencies

The upgrade of Application Server dependencies should include the upgrading to Release 4 of all locally-resident product components upon which Application Server depends. Shared components are upgraded automatically by the Java ES Installer as part of the upgrade procedure (see Step 3 on page 191).

When upgrading Application Server dependencies, they should be upgraded in the following order, all before you upgrade Application Server. You can skip any that might already have been upgraded.

- **Message Queue**. See Chapter 7, “Message Queue” on page 149
- **Web Server (optional).** See Chapter 6, “Web Server” on page 137

Back Up Application Server Data

The Application Server upgrade from Release 2 to Release 4 does not overwrite Release 2 configuration data. However, for safe measure, the configuration directory of all Application Server instances should be backed up before performing the upgrade to Release 4.

The configuration directories are at the following location:

AppServer7-base/domains/*domainName*/*instanceName*/config

Obtain Required Configuration Information and Passwords

You should know the following information about your currently installed version:

- Application Server administrator user ID, password, and master password
- Release 2 Application Server base directory

Upgrading Release 2 Application Server

This section discusses considerations that impact the upgrade procedure for Application Server followed by a description of the procedure itself.

Upgrade Considerations

The upgrade of Application Server software to Java ES Release 4 takes into account the following considerations:

- Any J2EE components running in an Application Server instance should be shut down before you upgrade that instance. However, if you use load balancing to provide high availability or scalability, this requirement might be relaxed.
- All instances of Application Server running on a single computer (all corresponding to the same installed Application Server image) must be shut down while the installed image is being upgraded.

Upgrade Procedure

The procedure documented below applies to all Application Server instances residing locally on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

```
su -
```

2. Stop all Application Server and related processes.

3. Install Release 4 Application Server using the Java ES installer, choosing the Configure Later option.

   Choose to install at least the first three subcomponents, including the Node Agent component.

   Once Application Server software is installed, be sure to perform the post-install procedures provided in "Configuring Application Server After a Configure Later Installation" on page 193.

4. Identify both target and source installation directories, for example:

   - Default Release 2 source on Solaris: `/opt/SUNWappserver7`
   - Default Release 4 target on Solaris: `/opt/SUNWappserver/appserver`

5. Run the `asupgrade` utility.

   The `asupgrade` utility creates a Release 4 node agent under which it migrates Release 2 Application Server instances.

   The utility is located under the Application Server directory, for example:

   - Upgrade wizard mode: *AppServer8-base*`/bin/asupgrade`
   - Upgrade console mode: *AppServer8-base*`/bin/asupgrade -c`

   The upgrade wizard or upgrade console will guide you through the upgrade steps.

   For more information about the Application Server `asupgrade` utility, refer to Chapter 3 of the *Application Server Enterprise Edition 8.1 Upgrade and Migration Guide 2005Q1* (`http://docs.sun.com/doc/819-0222`).

6. Start the Domain Administration Server (DAS).

   AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
   `--password` *password domainName*

7. Restart upgraded Application Server instances.

   Do this by starting the node agent under which the upgraded Application Server instances have been migrated:

   AppServer8-base`/bin/asadmin start-node-agent --user` *admin_ID*
   `--password` *password nodeagentName*

   where *nodeagentName* has the form *hostName_domainName*.
   The default *domainName* is `domain1`.

Verifying the Upgrade

Start the Admin Console and verify that these servers are started. If any of the servers are not running, check the following log file for failures that might be caused by port conflicts:

AppServer8Config-base/nodeagents/*nodeagentName*/*instanceName*/logs/server.log

If there failures due to port conflicts, use the Admin Console to modify the port numbers to eliminate the conflicts, then stop and restart the node agent.

You can verify the upgrade of Application Serverto Release 4 by entering the following command:

AppServer8-base/bin/asadmin version --verbose

See Table 9-3 on page 181 for output values.

Post-Upgrade Tasks

There are a number of post-upgrade tasks beyond the steps described in "Upgrade Procedure" on page 190. These involve the migration of Release 2 J2EE components and applications to run in a Release 4 Application Server environment and redeploying them to the appropriate Application Server instances.

For more information about migrating J2EE components and applications, refer to Chapter 4 of the *Application Server Enterprise Edition 8.1 Upgrade and Migration Guide 2005Q1* (http://docs.sun.com/doc/819-0222).

Rolling Back the Upgrade

The procedure for rolling back the upgrade to Release 4 of Application Server is simply to revert to the Release 2 version of Application Server, which was not removed by the upgrade to Release 4.

Multiple Instance (Cluster) Upgrades:

The Application Server's asupgrade utility can be used to upgrade multiple instance clusters. For instructions, see Chapter 3 of the *Application Server Enterprise Edition 8.1 2005Q2 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-2559).

Configuring Application Server After a Configure Later Installation

After a Configure Later installation, you will need to run a script to set the Application Server environment. Use the following steps.

1. Locate the accessory distribution for Application Server:

   Sun Java Enterprise System 2005Q4 Accessory CD 1,
   Application Server Add Ons for Solaris SPARC and x86,
   CD Image 1 of 1

   Accessory contents can be downloaded from the Sun Download Center at
   `http://www.sun.com/software/javaenterprisesystem/get.xml`

2. Refer to the `ReadMe` file in the `Addon` folder in the accessory distribution and perform the procedures specified.

   a. Run the `postInstall` script.

      The main script in the `Addon` folder, `postInstall`, should be run from the accessory distribution, otherwise some files will not be found.

      The scripts configure and create the *AppServer8-base*`/bin/*` shell scripts and an `config/asenv` file from templates that are installed during installation. (Normally the installer creates the `bin/*` shell scripts, but if Configure Later is chosen, they have to be created as described.)

   b. Create a new domain.

      When using the `asadmin create-domain` command to create a new domain, you specify values for two parameters: `adminPort` and `instancePort`. The `adminPort` value can be the same as that used by the Release 2 server instance, however the `instancePort` value should not be the same as that used by any of the Release 2 server instances. By choosing an unused `instancePort` value, you will avoid conflict between the Release 4 DAS instance and the Release 2 server instances that are migrated to Release 4 (see Step 5 on page 191).

3. If necessary, modify the environment settings in the *AppServer8-base*`/config/asenv` file.

   You have to edit the file manually.

NOTE To configure Application Server for load balancing, refer to the "Configuring Web Servers for HTTP Load Balancing" section in the "Application Server High Availability Features" chapter of the *Sun Java System Application Server Enterprise Edition High Availability Administration Guide* (http://docs.sun.com/doc/819-0216).

Chapter 10

Web Proxy Server

This chapter describes how to upgrade Web Proxy Server to Java ES 2005Q4 (Release 4): Sun Java System Web Proxy Server 4.0 2005Q4.

The chapter provides a general overview of upgrade issues as well as the upgrade procedure. The upgrade of Web Proxy Server is supported only on Solaris platforms:

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 Server was initially installed. If not, the Java ES installer assigns a default value.

The default value of *WebProxyServer-base* depends on operating system platform:

- Solaris: `/opt/SUNWproxy`
- Linux: `/opt/sun/webproxyserver`

Overview of Web Proxy Server Upgrades

This section describes the following general aspects of Web Proxy Server that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Web Proxy Server

Java ES Release 4 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 Java Enterprise System.

Web Proxy Server Upgrade Roadmap

Table 10-1 shows the Web Proxy Server upgrade path to Java ES Release 4. Web Proxy Server was not included in previous Java ES releases. The table applies to the Solaris operating system only, because Web Proxy Server was not previously supported on the Linux operating system.

Table 10-1 Upgrade Paths to Java ES Release 4: Sun Java System 4: Web Proxy Server 4.0.1 2005Q4

Java ES Release	Web Proxy Server Version	General Approach	Re-configuration Required
Pre-dates Java ES releases	Sun ONE Web Proxy Server 3.6 (Hereafter referred to as Version 3.6)	Direct upgrade: Performed using the Java ES installer to install in new location then migrating configuration data using administration tools	Configuration information must be migrated to new location.

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 10-2 Web Proxy Server Data Usage

Type of Data	Location	Usage
Configuration data	*WebProxyServer-base*/proxy-server*id*/config directory Contains files such as: server,xml, magnus.conf, obj.conf, and so forth	Stores configuration information for the server, cache, filters, routing, and other functional aspects of Web Proxy Server

Compatibility Issues

Release 4 Web Proxy Server represents a major change in the Netscape Server API (NSAPI) interface supported by Version 3.6. Any NSAPI plug-ins developed for Version 3.6 will need to be recompiled against the current version of NSAPI to resolve incompatibilities.

Web Proxy Server Dependencies

Web Proxy Server dependencies on other Java ES components can impact the procedure for upgrading and re-configuring Web Proxy Server software.

Web Proxy Server has dependencies on specific Java ES shared components (see Table 1-6 on page 40), but has no dependencies on other Java ES product components. It can be used with Directory Server, Web Server, and Application Server, but has no dependencies on these components.

Upgrading Web Proxy Server to Release 4

This section includes information about upgrading Web Proxy Server from Version 3.6 to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

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

- **General Upgrade Approach.** The upgrade is performed by using the Java ES installer to install Release 4 Web Proxy Server in a directory different from version 3.6. The Web Proxy Server administration server is then used to migrate configuration settings (but not the cache content) from Version 3.6 to Release 4.
- **Upgrade Dependencies.** While Web Proxy Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Web Proxy Server is compatible with the Release 3 versions of these components. Upgrade of these shared components, however, is automatically performed by the Java installer when upgrading Web Proxy Server to Release 4.
- **Backward Compatibility.** Release 4 Web Proxy Server is backwardly compatible with Version 3.6, except that plug-ins developed using the NSAPI interface supported by Version 3.6 must be recompiled with the NSAPI interface supported by Release 4.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Web Proxy Server is achieved by reverting to Version 3.6, which was left unchanged by the upgrade.
- **Platform Issues.** The approach for upgrading Web Proxy Server is the same on all Solaris platforms, however Version 3.6 is not supported on Linux platforms.

Web Proxy Server Upgrade

This section provides an overview of how to perform an upgrade of Web Proxy Server to Java ES Release 4. Web Proxy Server was not previously supported on the Linux platform. Hence upgrade of Web Proxy Server to Java ES Release 4 is only performed on the Solaris platform. The section covers the following topics:

- Pre-Upgrade Tasks
- Upgrading Web Proxy Server
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Web Proxy Server by entering the following command:

WebProxyServer-base`/`*proxy-serverid*`/start -version`

Table 10-3 Web Proxy Server Version Verification Outputs

Java ES Release	Web Proxy Server Version Number
non-Java ES release Version 3.6	`3.6`
Release 4	`4.0.1`

Upgrade Web Proxy Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, the Java ES installer that is used to upgrade Web Proxy Server to Release 4 automatically upgrades all shared components upon which Web Proxy Server depends (see Table 1-6 on page 40).

Back Up Web Proxy Server Data

The Web Proxy Server upgrade to Release 4 does not modify Version 3.6 configuration data. However any unsaved changes to Version 3.6 configuration data made using the administration interface must be saved before performing the upgrade.

Obtain Required Configuration Information and Passwords

To upgrade from Version 3.6, you need to know the installation directory path for that installed version.

Upgrading Web Proxy Server

This section discusses considerations that impact the upgrade procedure for Web Proxy Server followed by a description of the procedure itself.

Upgrade Considerations

All Web Proxy Server instances corresponding to the same installed Web Proxy Server image are upgraded at the same time. However, the migration of configuration data has to be done separately for each instance. All such instances should be shut down when patches are being applied to the installed image.

Upgrade Procedure

The procedure documented below applies to Web Proxy Server software on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Install Web Proxy Server Release 4.

   a. Run the Java ES installer.

   b. Select Web Proxy Server from the selection panel.

   c. Select the Configure Now option.

   d. Quit the Java ES installer when installation is complete.

3. Migrate configuration settings to the newly installed version.

   This operation must be performed separately for each Web Proxy Server instance.

   a. Start the Web Proxy Server Administration Server.

      WebProxyServer-base/proxy-admserv/start

   b. Log in to the administration graphical interface.

   c. Click on the Server tab and then click Migrate Server.

   d. Specify the Version 3.6 installation directory path.

   e. Select the instance to migrate.

   f. Click the Migrate button.

      After successful migration, the migration screen provides a list of additional configurations that must be performed manually. It provides the data that needs to be added and the corresponding configuration file.

      For more information on migrating configuration settings refer to *Sun Java System Web Proxy Server 4 2005Q4 Installation and Migration Guide* (http://docs.sun.com/doc/819-3649)

4. Make any additional configuration changes specified in Step f.

   Refer to the *Sun Java System Web Proxy Server 4 2005Q4 Configuration File Reference* (http://docs.sun.com/doc/819-3651) for more information.

Verifying the Upgrade

You can verify the upgrade of Web Proxy Server to Release 4 by starting a Web Proxy Server instance with the -version option:

WebProxyServer-base/*proxy-serverid*/start -version

See Table 10-3 on page 199 for output values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure” on page 200.

Rolling Back the Upgrade

The upgrade of Web Proxy Server to Release 4, documented in “Upgrading Web Proxy Server” on page 200, cannot be rolled back. However, you can revert to Version 3.6, which was left in tact by the Release 4 upgrade procedure.

Chapter 11

Access Manager

This chapter describes how to upgrade Access Manager software from previous Java ES versions to Java ES 2005Q4 (Release 4): Sun Java System Access Manager 7 2005Q4.

The chapter provides a general overview of Access Manager upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

NOTE File locations in this chapter are specified with respect to two directory paths referred to as *AccessManager-base* and *AccessManagerConfig-base*. At least part of these paths 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* depends on operating system platform:

- Solaris: `/opt/SUNWam`
- Linux: `/opt/sun/identity`

The default value of *AccessManagerConfig-base* depends on operating system platform:

- Solaris: `/etc/opt/SUNWam`
- Linux: `/etc/opt/sun/identity`

Overview of Access Manager Upgrades

This section describes the following general aspects of Access Manager that impact upgrading to Java ES 2005Q4 (Release 4):

NOTE Versions of Access Manager that predated Java ES Release 3 were named Identity Server. Hence references to Identity Server in this chapter are to earlier versions of the Java ES Access Manager component.

About Java ES Release 4 Access Manager

Java ES Release 4 Access Manager has been enhanced in major ways. On the back end, the product has been re-architected to support multiple identity repositories, or user data stores. Thus Release 4 Access Manager supports not only an LDAP directory such as Directory Server, but other data storage protocols and formats as well. Release 4 Access Manager includes new interfaces and new services to support the integration of multiple identity repositories.

On the front end, a new Access Manager Console is used to configure the new Access Manager services and identity repositories.

The new functional capabilities and interfaces make Release 4 Access Manager a major new release. In order to provide backward compatibility, Release 4 can be run in legacy mode, which supports the Java ES components that depend on Release 3 Access Manager services (for more information, see “Compatibility Issues” on page 207).

Access Manager Upgrade Roadmap

Table 11-1 shows the supported Access Manager upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 11-1 Upgrade Paths to Java ES Release 4: Sun Java System Access Manager 7 2005Q4

Java ES Release	Access Manager Version	General Approach	Re-configuration Required
Release 3	Sun Java System Access Manager 6.3 2005Q1	Direct upgrade: Performed by removing the Release 3 version and then doing a full installation and re-configuration of Release 4.	Configuration data Customized JSPs for Access Manager console and authentication UI Directory schema
Release 2	Sun Java System Identity Server 6.2 2004Q2 and also 6.2 SP1	Direct upgrade: Performed by removing the Release 2 version and then doing a full installation and re-configuration of Release 4.	Configuration data Customized JSPs for Access Manager console and authentication UI Directory schema
Release 1	Sun ONE Identity Server 6.1	No direct upgrade: But you can upgrade first to Release 3 using procedures in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062). Then upgrade from Release 3 to Release 4.	Configuration data Customized JSPs for Access Manager console and authentication UI Directory schema
Pre-dates Java ES releases	Sun ONE Identity Server 6.0 or 6.0 SP 1 or iPlanet Directory Server Access Management Edition (DSAME) 5.1	No direct upgrade.	

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 11-2 Access Manager Data Usage

Type of Data	Location	Usage
Configuration data	*AccessManagerConfig-base*/`config/AMConfig.properties` *AccessManagerConfig-base*/`config/serverconfig.xml` 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 (Java ES Release 3 and 4):
	`server.policy` and `domain.xml` files in *AppServer8Config-base*/`domains/`*domainName*/`config`
	Application Server (Java ES Release 2):
	`server.policy` and `server.xml` files in *AppServer7Config-base*/`domains/`*domainName*/`config`
	WebSphere and WebLogic:
	Respective policy and configuration files are modified when Access Manager is configured for these web containers.
		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.

Compatibility Issues

The new functional capabilities of Release 4 Access Manager involve the following new interfaces:

- Plug-ins for multiple back-end identity repositories
- New directory information tree structure for storing service configuration information so that authentication properties and authorization policies can be grouped into access control *realms* that can be associated with a user or group of users.
- New API for Access Manager clients
- New Access Manager Console user interface

Access Manager support for these new interfaces is enabled by configuring Access Manager to run in enhanced (Realm) mode. However, Realm mode is not compatible with the earlier Java ES Release 3 or Release 2 Access Manager. For example, directory data has to be migrated to support Realm mode operation. The enhanced Access Manager Console is needed to support enhanced Access Manager services.

In addition, Realm mode does not support other Java ES components, such as Portal Server, Communications Express, Messaging Server, and others.

To support backward compatibility, Release 4 Access Manager can be configured to run in Legacy mode. With some minor exceptions (see *Sun Java System Access Manager 7 2005Q4 Release Notes (*http://docs.sun.com/doc/819-2134), Legacy mode is backwardly compatible with Release 3 Access Manager.

Legacy mode is necessary to support other Java ES components, as well as older versions of Access Manager policy agents, 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 4 Legacy mode.

Even when configured to run in Legacy mode, however, Release 4 Access Manager is incompatible with Release 3 Delegated Administrator. If Access Manager is upgraded to Release 4, then Delegated Administrator also must be upgraded to Release 4 to provision users for Messaging Server and Calendar Server. However, Messaging Server and Calendar Server do not, themselves, have to be upgraded to Release 4.

Access Manager Dependencies

Access Manager dependencies on other Java ES components can impact the procedure for upgrading and re-configuring 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 (see Table 1-6 on page 40). 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, Java ES Application Server, or third-party web containers (from Weblogic and WebSphere). Access Manager upgrades must therefore be re-configured 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.

Upgrading Access Manager from Java ES Release 3

This section includes information about upgrading Access Manager from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Access Manager to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by removing previous versions and newly installing Release 4. An `ampre70upgrade` script is provided for removing the Release 3 version and the Java ES installer is then used to install Release 4. Re-configuration of Access Manager is subsequently performed using the `amconfig` script, and directory schema is migrated using the `amupgrade` script.
- **Upgrade Dependencies.** While Access Manager has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Access Manager is compatible with the Release 3 versions of all these components. Upgrade of these components is therefore optional with respect to upgrade of Access Manager to Release 4.

  In addition, Release 4 Access Manager is dependent upon Directory Server and Web Server (or Application Server or third-party web containers), as described in “Access Manager Dependencies” on page 208. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Access Manager to Release 4.
- **Backward Compatibility.** Release 4 Access Manager is not compatible with Release 3, however it does support a compatible legacy mode (see “Compatibility Issues” on page 207).

- **Upgrade Rollback.** There is no utility for rolling back the Access Manager upgrade. In fact, the number of re-configurations required to roll back Access Manager to its original state make such a rollback impractical.
- **Platform Issues.** The general approach for upgrading Access Manager is the same on both Solaris and Linux operating systems. The procedures that follow indicate platform-specific commands or file locations where appropriate.

Full Release 3 Access Manager Upgrade

This section describes how to perform a full Access Manager upgrade from Java ES Release 3 to Java ES Release 4:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Access Manager using the following command:

AccessManager-base`/bin/amadmin --version`

Table 11-3 Access Manager Version Verification Outputs

Java ES Release	Access Manager Version Number
Release 2	`6.2`
Release 3	`6 2005Q1`
Release 4	`7 2005Q4`

Upgrade Access Manager Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, because Access Manager does not require upgrading the Java ES Release 3 components upon on which it depends, this task is optional.

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

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.

2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, "Directory Server and Administration Server" on page 103.

3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 6, "Web Server" on page 137 and Chapter 9, "Application Server" on page 175, respectively.

   If web container software is not upgraded before Access Manager, the upgrade procedure (using the `amconfig` script) will configure and re-deploy Access Manager to the existing web container.

Back 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* (`http://docs.sun.com/doc/817-7613`).

Back Up Release 3 Access Manager Configuration Information

Because the re-configuration of Release 4 Access Manager software requires the re-configuration of the Release 3 version, it is important to back up configuration files to a known location. The following files should be backed up:

- The `AMConfig.properties` file
  AccessManagerConfig-base`/config/AMConfig.properties`
- The `serverconfig.xml` file
  AccessManagerConfig-base`/config/serverconfig.xml`

- Web container configuration files:
  - 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 *AppServer7Config-base*`/domain/domain1/config`
  - For third-party web containers: the appropriate configuration files
- JAR files for authentication and customized modules.

  AccessManager-base`/lib`

Back Up Web Container Customized Files

If you have any web container customized files referenced by Access Manager, you should back them up. These customizations might include the following:

- Customized Access Manager console JSP pages.
  AccessManager-base`/web-src/applications`
- Customized authentication UI JSP pages.
  AccessManager-base`/web-src/services`
- Customized XML files.
  AccessManagerConfig-base`/config/xml`

TIP Make note of your customizations so you can re-apply them using the backed-up code after you upgrade Access Manager.

Back Up Release 3 Access Manager Log and Debug Files

For the purpose of analyzing system state information, it is a good idea to back up log and debug files so they are not lost. These files are at the following locations:

- Debug files
  `/var/`*AccessManager-base*`/debug`
- Log files
  `/var/`*AccessManager-base*`/logs`

Obtain Required Configuration Information and Passwords

To upgrade Access Manager, you must provide specific configuration information, including:

- Access Manager administrator user ID and password
- LDAP user ID and password
- Directory Manager name and password for the Directory Server instance that Access Manager is using

Upgrading Release 3 Access Manager

The upgrade of Access Manager software to Java ES Release 4 includes procedures for re-configuring Access Manager and for migrating Access Manager data.

Upgrade Summary

The procedure for upgrading Access Manager consists of the following steps:

1. Remove the Java ES Release 3 Version of Access Manager. Use the `ampre70upgrade` script.
2. Install the Java ES Release 4 Version of Access Manager. Use the Java ES Release 4 installer with the Configure Later option.
3. Upgrade mobile access software.
4. Re-customize JSPs for Access Manager.
5. Undeploy Access Manager, re-configure, and re-deploy into a Web Container. Use the `amconfig` script.
6. Update the directory structure and schema. Use the `amupgrade` script.

These steps are each documented in the following procedures.

Upgrade Procedures

1. Remove the Java ES Release 3 Version of Access Manager.
   a. Log in as root to the computer hosting Release 3 Access Manager or become superuser.

      ```
      su -
      ```

   b. Change directory to the *platform*`/Product/identity_svr/Tools` directory in the Java ES Release 4 distribution.

   c. Obtain the values of the following parameters to be requested by the `ampre70upgrade` script:

Table 11-4 Access Manager Configuration Parameters: `ampre70upgrade`

Parameter	Value
Directory Server Host	Set the fully-qualified name: *hostname.domian*
Directory Server Port	Specify a non-SSL port number[1] Default: 389
Top-Level Administrator DN	Default: `uid=amadmin,ou=People,dc=iplanet,dc=com`
Top-Level Administrator Password	

1. The pre-upgrade process will not complete successfully if you specify a Directory Server SSL port such as the default SSL value of 636.

d. Make sure that Directory Server is running or start it if it is not.

e. Run the `ampre70upgrade` script.

```
./ampre70upgrade
```

The script backs up Access Manager configuration files and removes Release 3 base packages (localized packages must be removed manually per Step f).

f. Manually remove localized Access Manager packages on your computer.

The `ampre70upgrade` script does not remove localized Access Manager packages. They must be removed by hand to perform a properly localized upgrade.

- Use pkgrm on Solaris platforms to remove: `SUNWaml`*Locale*, `SUNW`*Locale*`ammmap`
- Use `rpm -e` on Linux to remove: `sun-identity-sdk-`*Locale*

2. Install the Java ES Release 4 Version of Access Manager.

a. Run the Java ES installer on the computer hosting Release 3 Access Manager.

b. Select Access Manager from the selection panel.

If a “Conflict” message appears on the screen, it means the Installer has found Access Manager configuration information from the previous version, which is expected. Re-configuration will be performed in subsequent steps. You can ignore this “Conflict” message and proceed.

c. Specify the same installation directory in which Release 3 was installed.

d. Select the Configure Later option.

e. Quit the Java ES installer when installation is complete.

NOTE If you are using the Java ES Installer command line interface to install Access Manager, it will automatically install Directory Server software as well. If you are using a remote Directory Server you can uninstall the local Directory Server software using the procedures in the *Java Enterprise System Installation Guide for UNIX*.

3. Upgrade mobile access software.

Access Manager mobile access software needs to be upgraded by patching the Release 3 version. The patches needed are shown in the following table:

Table 11-5 Patches[1] to Upgrade Access Manager Mobile Access software

Description	Solaris Patch ID	Linux Patch ID
Mobile Access software	119530-01 (SPARC) 119531-01 (x86)	119532-01 • `sun-identity-mobileaccess-6.2-25.i386.rpm` • `sun-identity-mobileaccess-config-6.2-25.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

a. Obtain the required patches using the patch numbers from Table 11-5.

Patches can be downloaded to `/tmp` from:
`http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

b. Log in as root or become superuser.

```
su -
```

c. Apply the patches in Table 11-5.

On Solaris:

```
patchadd patch_ID
```

On Linux:

```
rpm -Fvh sun-identity-mobileaccess-6.2-25.i386.rpm
rpm -Fvh sun-identity-mobileaccess-config-6.2-25.i386.rpm
```

4. Re-customize JSPs for Access Manager.

Re-apply the Release 3 customizations to JSPs for the Access Manager Console and authentication user interface (UI) that you saved under "Back Up Web Container Customized Files" on page 212.

Then, copy the customized JSP files to the correct directories. For example, on Solaris systems:

- 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* (http://docs.sun.com/doc/819-2139).

5. Undeploy Access Manager, re-configure, and re-deploy into a Web Container.

Configure Access Manager for your specific web container by running the amconfig script. The amconfig script (and the associated amsamplesilent template input file) resides in the following directory:

AccessManager-base/bin

For information about the amconfig script and the amsamplesilent template file, see the *Sun Java System Access Manager Administration Guide* (http://docs.sun.com/doc/817-7647).

Perform the following steps to re-configure and re-deploy Access Manager to the web container:

a. If you choose to upgrade your web container software, as described in "Upgrade Access Manager Dependencies" on page 211, make sure the upgrade is complete.

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

c. Create an `amconfig` input file based on the `amsamplesilent` template input file:

`cp amsamplesilent` *config-file*

d. Set the configuration parameters in *config-file*.

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

Table 11-6 Access Manager Configuration Parameters

Parameter	Value
Upgrade Parameters	
DEPLOY_LEVEL	26 (for undeploy) or 1 (for re-configure and deploy)
DIRECTORY_MODE	5 (Existing Upgrade)
AM_REALM	set to `disabled` (Realm Mode is disabled, Legacy Mode is therefore enabled) (Default = `enabled`)
JAVA_HOME	set to JDK Release 4 directory: `/usr/java/jdk1.5.0_04/`
WEB_CONTAINER	set to the value appropriate to the web container type you are using and fill out only the corresponding section of *config-file*.
WS61_INSTANCE (If using Web Server as the web container)	=`https-`<*hostname*>`.`<*domain*> where the value above matches the instance name in /*WebServer-base*/`SUNWsbsvr/` The values is case-sensitive.
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`

Table 11-6 Access Manager Configuration Parameters *(Continued)*

Parameter	Value
SERVER_DEPLOY_URI	com.iplanet.am.services.deploymentDescriptor
PASSWORD_DEPLOY_URI	com.sun.identity.password.deploymentDescriptor
AM_ENC_PWD	am.encryption.pwd

For other parameters, provide the same values that were used in the Release 3 configuration that you are upgrading, unless you are changing web container or passwords.

e. Run `amconfig` to undeploy Access Manager

Set the value of `DEPLOY_LEVEL` in *config-file* to `26`.

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

f. Run `amconfig` to reconfigure Access Manager and deploy into web container.

Set the value of `DEPLOY_LEVEL` in *config-file* to `1`.

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

6. Update the directory structure and schema.

Release 4 Access Manager co-exists with the Release 3 directory structure, but the structure must be modified to support Release 4 capabilities. Update the Access Manager directory structure and schema to Release 4 by running the `amupgrade` script, which is installed in the following directory:

- Solaris:
  AccessManager-base`/upgrade/scripts`
- Linux:
  AccessManager_base`/identity/upgrade/scripts`

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

Table 11-7 Access Manager Configuration Parameters: amupgrade

Parameter	Value
Directory Server Host	Set the fully qualified name: *hostname.domain*
Directory Server Port	Specify a non-SSL port number[1] Default: 389
Directory Manager DN	Default: cn=Directory Manager
Directory Manager Password	
Top-Level Administrator DN	Default: uid=amadmin,ou=People,dc=iplanet,dc=com
Top-Level 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.

1. The upgrade process will not complete successfully if you specify a Directory Server SSL port such as the default SSL value of 636.

b. Run the amupgrade script.

```
cd AccessManager-base/upgrade/scripts
./amupgrade
```

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

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

Solaris:
```
/var/sadm/install/logs/
       Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

Linux:
```
/var/log/Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

7. Start Access Manager.

Re-start the web container in which Access Manager is deployed.

Verifying the Access Manager Upgrade

After you finish the upgrade procedure, verify that it was successful as follows:

1. Log in to the Access Manager console as `amadmin` using the following URL:

   `http://`*hostname.domain*`:`*port*`/amconsole`

   where *hostname.domain*:*port* is the fully qualified host name and port number of the web container hosting Access Manager.

   Verify that new Release 4 services referred to in "About Java ES Release 4 Access Manager" on page 204 are available under the "Service Configuration" tab.

2. Review the status of the upgrade by checking the following upgrade log files in the `/var/sadm/install/logs` directory:

   Sun Java Enterprise System installer:

   - `Java_Shared_Component_Install.`*timestamp*
   - `Java_Enterprise_System_install.`*Atimestamp*
   - `Java_Enterprise_System_install.`*Btimestamp*
   - `Java_Enterprise_System_Summary_Report_install.`*timestamp*

   `amupgrade` script:

   - `Sun_Java_System_Identity_Server_upgrade_dit_log.`*timestamp*

3. Review Access Manager trouble shooting files for errors.

   The files are located at `/var/opt/SUNWam/debug`

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, refer to the *Sun Java System Access Manager Administration Guide* (`http://docs.sun.com/doc/817-7647`).

Rolling Back the Upgrade

No scripts are provided for rolling back Access Manager to its pre-upgrade state. The process must be performed manually using Access Manager data that was backed up as part of the pre-upgrade tasks (see "Back Up Release 3 Access Manager Log and Debug Files" on page 212). Rollback is too difficult to be practical.

Multiple Instance Upgrades: Release 3 and Release 4 Co-existence

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 3 includes a step for migrating directory schema to support Release 4. Release 3 Access Manager does not support Release 4 directory schema, however Release 4 Access Manager does support Release 3 directory schema.

Java ES Release 4 Access Manager and Release 3 Access Manager instances can coexist and run concurrently against the same Directory Server only if the directory schema has not yet been migrated to Release 4. Therefore, in rolling upgrades, the directory schema should not be migrated to Release 4 until all Access Manager instances have been first upgraded to Release 4.

In performing rolling upgrades, upgrade each instance of Access Manager as described in "Upgrading Release 3 Access Manager" on page 213, except for "Update the directory structure and schema."step on page 218. When all instances have been upgraded, then that step can be performed.

Release 3 Access Manager SDK-only Upgrades

In some deployment architectures, the Access Manager SDK component is installed on one or more computer systems without installing other Access Manager components on those computers. Access Manager SDK serves as a remote interface to Access Manager and must be re-configured for the same operational mode as Access Manager: Legacy or Realm. As a remote interface to Access Manager, the SDK does not need to be configured to access Directory Server.

If Access Manager SDK is being used to support a web component, such as Portal Server or Communications Express, which depends upon web container services, Access Manager SDK must be configured for the corresponding web container. However, Access Manager SDK can also support non-web components, and no web container is needed.

The procedure for upgrading Access Manager SDK is a subset of the procedure for the full Access Manager upgrade, based on the above characteristics.

This section describes how to perform an Access Manager SDK-only upgrade from Java ES Release 3 to Java ES Release 4:

- Pre-Upgrade Tasks
- Upgrading Release 3 Access Manager SDK
- Verifying the Access Manager SDK Upgrade
- Rolling Back the Upgrade

Pre-Upgrade Tasks

The pre-upgrade tasks for Access Manager SDK are the same as for the full Access Manager upgrade, but exclude those related to Directory Server and administration tool customizations. The pre-upgrade tasks needed for Access Manager SDK are the following:

- “Upgrade Access Manager Dependencies” on page 211

  However, for Access Manager SDK, there is no dependency on Directory Server, and a dependency on web container software only in the case where Access Manager SDK runs in a web container.

- “Back Up Release 3 Access Manager Log and Debug Files” on page 212

  However, for Access Manager SDK, you only need to back up web container configuration files in the case where Access Manager SDK runs in a web container.

- "Back Up Release 3 Access Manager Log and Debug Files" on page 212

  You also need to obtain the admin username and password for accessing these files.

Upgrading Release 3 Access Manager SDK

The upgrade procedures for Access Manager SDK are the same as for the full Access Manager upgrade, but exclude those related to administration tool customizations and migrating directory schema.

1. Remove the Java ES Release 3 version of Access Manager SDK.

   Follow the instructions in "Remove the Java ES Release 3 Version of Access Manager." on page 213, except remove only Access Manager SDK.

2. Install Java ES Release 4 version of Access Manager SDK.

   Follow the instructions in "Install the Java ES Release 4 Version of Access Manager." on page 214, except install only Access Manager SDK.

3. Re-configure Access Manager SDK.

   Follow the instructions in "Undeploy Access Manager, re-configure, and re-deploy into a Web Container." on page 216, except set the `DIRECTORY_MODE=5` and the `DEPLOY_LEVEL` parameter as follows:

   - `If` Access Manager SDK is configured for a web container: `DEPLOY_LEVEL=4` (upgrade the SDK and configure the web container)
   - `If` Access Manager SDK is not configured for a web container: `DEPLOY_LEVEL=3` (upgrade the SDK only)

Verifying the Access Manager SDK Upgrade

There are three ways you can verify a successful Access Manager SDK upgrade:

- Run Portal Server, Communications Express, or other component that uses Access Manager SDK to interface with Access Manager, and check that the authentication works.
- Run the Access Manager SDK examples provided in the following location:

  /AccessManager-base`/samples/sdk`
- Check the value of the `com.iplanet.am.version` property, which is in the `AMConfig.properties` file:

  AccessManagerConfig-base`/config/AMConfig.properties`

Upgrade Rollback

No scripts are provided for rolling back Access Manager to its pre-upgrade state. The process must be performed manually using Access Manager data that was backed up as part of the pre-upgrade tasks (see "Back Up Release 3 Access Manager Log and Debug Files" on page 212). Rollback is too difficult to be practical.

Upgrading Access Manager from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Access Manager to Release 4 is the same as that for upgrading Release 3 Access Manager to Release 4, with a couple of exceptions, noted below.

Pre-Upgrade Tasks

Before you upgrade Access Manager, perform the procedures described in "Pre-Upgrade Tasks" on page 210, except replace "Upgrade Access Manager Dependencies" on page 211 with the following section and add the "Upgrade Directory Schema" section below.

Upgrade Access Manager Dependencies

As compared to the upgrade from Release 3, the Release 2 to Release 4 pre-upgrade tasks should include the upgrading to Release 4 of all shared components (see Table 1-6 on page 40) and all locally-resident product components upon which Access Manager depends.

When upgrading Access Manager dependencies, they should be upgraded in the following order, all before you upgrade Access Manager. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Directory Server rarely resides on the same computer as Access Manager, however, instructions for upgrading Directory Server to Release 4 are provided in "Upgrading Directory Server and Administration Server from Java ES Release 2" on page 122.

3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in "Upgrading Web Server from Java ES Release 2" on page 147 and "Upgrading Application Server from Java ES Release 2" on page 188, respectively.

Upgrade Directory Schema

If Directory Server was configured with Directory Preparation Tool (`comm_dssetup.pl`) to support Messaging Server, Calendar Server, or other communications components, you must first upgrade the directory schema using the Release 4 version of Directory Preparation Tool *before* upgrading Access Manager. Perform this pre-upgrade task after you have upgraded Access Manager dependencies. The procedure for upgrading Directory Preparation Tool is described in "Upgrading Directory Preparation Tool from Java ES Release 2" on page 240.

Release 2 Access Manager Upgrade

The procedure for upgrading Access Manager from Release 2 to Release 4 depends on the web container in which you are deploying Access Manager software.

Upgrading Release 2 Access Manager: Web Server Web Container

To upgrade Release 2 Access Manager to Release 4, when deploying into a Web Server web container, follow the instructions in "Upgrading Release 3 Access Manager" on page 213, except substitute Release 2 wherever Release 3 is referenced.

Upgrading Release 2 Access Manager: Application Server Web Container

To upgrade Release 2 Access Manager to Release 4, when deploying into a Application Server web container, there are two cases:

- Release 4 Application Server has been freshly installed. In this case, to upgrade Release 2 Access Manager to Release 4, follow the instructions in "Upgrading Release 3 Access Manager" on page 213, except substitute Release 2 wherever Release 3 is referenced.
- Release 2 Application Server has been upgraded to Release 4. In this case, to upgrade Release 2 Access Manager to Release 4, follow the instructions below.

To upgrade Access Manager when deployed in an upgraded Application Server web container, you follow Step 1 on page 213 through Step 4, except substitute Release 2 wherever Release 3 is referenced.

To summarize, Step 1 through Step 4 are as follows:

1. Remove the Release 2 Version of Access Manager.

   Use the `ampre70upgrade` script. Follow the instructions in "Remove the Java ES Release 3 Version of Access Manager." on page 213.

2. Install the Java ES Release 4 Version of Access Manager. Use the Java ES Release 4 installer with the Configure Later option.

3. Upgrade mobile access software.

4. Re-customize JSPs for Access Manager.

The Release 2 Application Server instance in which Access Manager was originally deployed (*instanceName*), when upgraded to Release 4, was migrated under a node agent created by the upgrade process. Upgrade of Access Manager in this upgraded Application Server instance requires the following additional steps:

5. Make sure that the following components that support Access Manager are running.

   a. Check that Directory Server is running.

   b. Start the Domain Administration Server (DAS) if it is not already started.

      AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
      `--password` *password domainName*

   c. Start the upgraded Application Server instance in which Access Manager is deployed (*instanceName*), if that server instance is not already running.

      Do this by starting the node agent under which the upgraded Application Server instance has been migrated:

      AppServer8-base`/bin/asadmin start-node-agent --user` *admin_ID*
      `--password` *password nodeagentName*

   In the above commands, and in subsequent steps, the following conventions are used:

   - *nodeAgentName* has the form *hostName_domainName.*
   - The default *domainName* is `domain1`
   - The default *instanceName* is `server1`

6. Undeploy Access Manager, reconfigure, and re-deploy into the Application Server instance. Use the `amconfig` script.

   a. Create an `amconfig` input file based on the `amsamplesilent` template input file:

      `cp amsamplesilent` *config-file*

   b. Set the configuration parameters in *config-file*.

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

Table 11-8 Access Manager Configuration Parameters

Parameter	Value
Upgrade Parameters	
DEPLOY_LEVEL	26 (for undeploy) or 1 (for re-configure and deploy)
DIRECTORY_MODE	5 (Existing Upgrade)
AM_REALM	set to `disabled` (Realm Mode is disabled, Legacy Mode is therefore enabled); Default = `enabled`
JAVA_HOME	set to JDK Release 4 directory: `/usr/java/jdk1.5.0_04/`
WEB_CONTAINER	set to the value for Application Server web container and fill out only the corresponding section of *config-file*.
AS81_INSTANCE	=*instanceName*
AS81_ADMIN_IS_SECURE	=`false`
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`
AM_ENC_PWD	`am.encryption.pwd`

For other parameters, provide the same values that were used in the Release 2 configuration that you are upgrading, unless you are changing web container or passwords.

c. Run `amconfig` to undeploy Access Manager.

Set the value of `DEPLOY_LEVEL` in *config-file* to `26`.

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

d. Run `amconfig` to reconfigure Access Manager and deploy into web container.

Set the value of `DEPLOY_LEVEL` in *config-file* to `1`.

```
cd /AccessManager-base/bin
./amconfig -s AccessManager-base/bin/config-file
```

7. Copy the `server.policy` file from the following directory:

AppServer8Config-base`/domains/`*domainName*`/config`

to the following target directory:

AppServer8Config-base`/nodeagents/`*nodeagentName*`/`*instanceName*`/config`

8. Modify the Release 4 Application Server `domain.xml` file.

a. Copy the Access Manager `classpath-suffix` and `server-classpath` information in the `server.xml` file of the Release 2 Application Server instance in which Access Manager was originally deployed:

AppServer7Config-base`/domains/`*domainName*`/`*instanceName*`/config/server.xml`

b. Append the copied classpath information to the `classpath-suffix` and `server-classpath` entries, respectively, of the `domain.xml` file of the upgraded Application Server instance in which Access Manager is deployed:

AppServer8Config-base`/nodeagents/`*nodeagentName*`/`*instanceName*`/config/domain.xml`

The classpath information should be added to the *instanceName*`-config` block of the Release 4 Application Server `domain.xml` file. This block begins with the following line:

```
<config dynamic-reconfiguration-enabled="true"
name="instanceName-config">
```

When making an addition to a classpath entry, be sure to include a colon (":"), or whatever path separator is being used in classpath entries, between the old and new information. You can also delete all entries with the *AppServer7-base* path (be careful not to introduce errors).

9. Restart the DAS.

AppServer8-base`/bin/asadmin stop-domain --user` *admin_ID*
`--password` *password domainName*

AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
`--password` *password domainName*

10. Restart the server instance in which Access Manager is deployed.

AppServer8-base`/bin/asadmin stop-node-agent --user` *admin_ID*
`--password` *password nodeagentName*

AppServer8-base`/bin/asadmin start-node-agent --user` *admin_ID*
`--password` *password nodeagentName*

11. Update the directory structure and schema as described in Step 6 on page 218.

Verifying the Access Manager Upgrade

After you finish the upgrade procedure, verify that it was successful, as described in "Verifying the Access Manager Upgrade" on page 220.

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, refer to the *Sun Java System Access Manager Administration Guide* (http://docs.sun.com/doc/817-7647).

Rolling Back the Upgrade

No scripts are provided for rolling back Access Manager to its pre-upgrade state. The process must be performed manually using Access Manager data that was backed up as part of the pre-upgrade tasks (see "Back Up Release 3 Access Manager Log and Debug Files" on page 212). Rollback is too difficult to be practical.

Chapter 12

Directory Preparation Tool

This chapter describes how to upgrade Directory Preparation Tool to Java ES 2005Q4 (Release 4): Sun Java System Directory Preparation Tool 6.3 2005Q4.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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

The default value of *DirPrepTool-base* depends on operating system platform:

- Solaris: `/opt/SUNWcomds`
- Linux: `/opt/sun/comms/dssetup`

Overview of Directory Preparation Tool Upgrades

This section describes the following general aspects of Directory Preparation Tool that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Directory Preparation Tool

Java ES Release 4 versions of Directory Preparation Tool represents a number of minor fixes needed to prepare Directory Server for use by Release 4 communications components (Messaging Server, Calendar Server, Communications Express, and Delegated Administrator).

For details, see the appropriate release notes.

Directory Preparation Tool Upgrade Roadmap

Table 12-1 shows the supported Directory Preparation Tool upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 12-1 Upgrade Paths to Java ES Release 4: Sun Java System Directory Preparation Tool 6.3 2005Q4

Java ES Release	Directory Preparation Tool Version	General Approach	Re-configuration Required
Release 3	Sun Java System Directory Preparation Tool 6.2 2005Q1	Direct upgrade: Perform by applying patches.	Prepare Directory Server for Release 4 communications components
Release 2	`comm_dssetup.pl` script Version 6.1 Revision 0.2 (bundled with Messaging Server and Calendar Server)	Direct upgrade: Perform by applying genesis patch followed by an upgrade patch.	Prepare Directory Server for Release 4 communications components

Table 12-1 Upgrade Paths to Java ES Release 4: Sun Java System Directory Preparation Tool 6.3 2005Q4 *(Continued)*

Java ES Release	Directory Preparation Tool Version	General Approach	Re-configuration Required
Release 1	`comm_dssetup.pl` script (bundled with Messaging Server and Calendar Server)	Direct upgrade not certified: But can perform by applying genesis patch followed by an upgrade patch.	Prepare Directory Server for Release 4 communications components
Pre-dates Java ES releases	`ims_dssetup.pl` script (bundled with Messaging Server)	No direct upgrade.:	

Directory Preparation Tool Data

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

Table 12-2 Directory Preparation Tool Data Usage

Type of Data	Location	Usage
Directory Server schema	Directory Server	Prepare Directory Server for Release 4 communications components: modifies schema, creates new entries and creates indexes

Compatibility Issues

Release 4 Directory Preparation Tool does not introduce any interface changes and is backwardly compatible with earlier versions.

Directory Preparation Tool Dependencies

Directory Preparation Tool has no dependencies on other Java ES components other than Directory Server. Directory Preparation Tool is used to configure Directory Server for use with Java ES communications components.

Upgrading Directory Preparation Tool from Java ES Release 3

This section includes information about upgrading Directory Preparation Tool from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Directory Preparation Tool to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. Directory Preparation Tool is then used to modify Directory Server as required to support Release 4 Messaging Server, Calendar Server, Communications Express, and Delegated Administrator components.
- **Upgrade Dependencies.** Directory Preparation Tool has no dependencies on Java ES shared components and is compatible with Release 3 Directory Server. Upgrade of Directory Server is therefore optional with respect to upgrade of Directory Preparation Tool to Release 4.
- **Backward Compatibility.** Release 4 Directory Preparation Tool is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade to Release 3 is achieved by removing the patches applied during the upgrade. The Release 3 Directory Preparation Tool can then be run against Directory Server to back out changes made by the Release 4 version.
- **Platform Issues.** The general approach for upgrading Directory Preparation Tool is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures, and you normally cannot roll back patches on the Linux platform.

Release 3 Directory Preparation Tool Upgrade

This section describes how to perform an upgrade of Directory Preparation Tool from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Directory Preparation Tool you should perform the tasks described below.

Verify Current Version Information

You can verify the version of Directory Preparation Tool last run against Directory Server by checking attribute values of the `cn=CommServers,o=comms-config` entry written by the tool:

```
./ldapsearch -D "cn=Directory Manager" -w password
             -b cn=CommServers,o=comms-config cn="CommServers"
             sunkeyvalue
```

The entry has two attributes that specify the current version:

- `dssetup_ver`=*version* (for example, `6.3`)
- `dssetup_rev`=*revision* (for example, `2.01`)

The tool will write a message to console only if the version of Directory Preparation Tool being run is the same or earlier than the version that was previously run. See the upgrade procedures, Step 5 on page 237 (Solaris) or Step 5 on page 239 (Linux), for how to run the tool.

Upgrade Directory Preparation Tool Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, the upgrade of Directory Preparation Tool to Release 4 does not depend upon any other Java ES component.

Back Up Directory Data

The Directory Preparation Tool upgrade from Release 3 to Release 4 does not in itself modify Directory Server data. However, as a safety measure, it is a good idea to back up Directory Server before upgrading the Directory Preparation Tool and running it against Directory Server.

Obtain Required Configuration Information and Passwords

Directory Preparation Tool upgrade requires you to know the superuser password. The tool remembers parameter values used in the previous run and supplies them as defaults when run the next time.

Upgrading Release 3 Directory Preparation Tool (Solaris)

This section discusses considerations that impact the upgrade procedure for Directory Preparation Tool followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Directory Preparation Tool software to Java ES Release 4 takes into account the following considerations:

- Release 3 Directory Preparation Tool was installed with Directory Server and resides on any computer hosting Directory Server.
- The upgrade of Directory Preparation Tool must be performed on the computer hosting every Directory Server instance being used by Messaging Server, Calendar Server, Communications Express, or Delegated Administrator components.
- The Release 4 Directory Preparation Tool upgrade patch for Solaris OS are shown in the following table:

Table 12-3 Patches[1] to Upgrade Directory Preparation Tool on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Directory Preparation Tool (DSSETUP)	118245-05	118246-05

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to the Directory Preparation Tool installed on the computer where Directory Server resides.

1. Obtain the required patches, based on Table 12-3.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   `su -`

3. Apply the appropriate Directory Preparation Tool patches in Table 12-3.

   `patchadd` *patch_ID*

4. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 3.

5. Run the Directory Preparation Tool against Directory Server.

   a. Confirm that Directory Server is running.

   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`

   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      `perl comm_dssetup.pl`

      Provide the parameters requested by the script.

Upgrading Release 3 Directory Preparation Tool (Linux)

This section discusses considerations that impact the upgrade procedure for Directory Preparation Tool followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Directory Preparation Tool software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 236), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Directory Preparation Tool upgrade patch for Linux OS is shown in the following table:

Table 12-4 Patches[1] to Upgrade Directory Preparation Tool on Linux

Description	Patch ID and RPM names
Directory Preparation Tool (DSSETUP)	118247-05 • `sun-comms-dssetup-6.3-2.5.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to the Directory Preparation Tool installed image on the computer where Directory Server resides.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patch using the patch number and RPM name from Table 12-4. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Apply the RPMs for Directory Preparation Tool in Table 12-4.

   ```
   rpm -Uvh sun-comms-dssetup-6.3-2.5.i386.rpm
   ```

4. Confirm that the upgrade was successful:

   `rpm -q sun-comms-dssetup`

   The new version number of the RPM should be returned.

5. Run the Directory Preparation Tool against Directory Server.

   a. Confirm that Directory Server is running.

   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`

   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      `perl comm_dssetup.pl`

      Provide the parameters requested by the script.

Verifying the Upgrade

You can verify successful upgrade of Directory Preparation Tool and extension of directory schema by checking the log file created when running the script. The log file is located at:

`/var/tmp/dssetup_`*yyyymmddhhmmss*

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 237 and "Upgrade Procedure (Linux)" on page 238.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Directory Preparation Tool followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Directory Preparation Tool reverses the procedure for upgrading to Release 4. However, among the changes made by Directory Preparation Tool are modifications to Directory Server schema. These changes are not backed out by the rollback procedure described below, however the schema changes are backwardly compatible.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Remove the patches in Table 12-3 on page 237.

   `patchrm` *patch_ID*

3. Run the rolled-back Directory Preparation Tool against Directory Server.

   Directory Server modifications, including indexes are restored to their previous states, however schema changes remain in place. There is no negative impact to the schema extensions; they are backwardly compatible.

Upgrading Directory Preparation Tool from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Directory Preparation Tool to Release 4 is similar to that for upgrading Release 3 Directory Preparation Tool to Release 4, with the following exception.

In Java ES Release 2, Directory Preparation Tool (then called `comm_dssetup`) was bundled with Messaging Server and Calendar Server and not installed as a separate package. Hence no Directory Preparation Tool installed packages or RPMs reside on the computer hosting Directory Server. For this reason, to upgrade from Release 2 to Release 4, you have to install Directory Preparation Tool packages:

- For Solaris platforms, the DPT packages are installed as genesis patches, which contain the full Directory Preparation Tool software. You then apply patches to upgrade to Release 4.
- For Linux platforms, the Release 4 packages are directly installed.

Upgrades from Release 2 Directory Preparation Tool to Release 4 is similar to that described in “Upgrading Directory Preparation Tool from Java ES Release 3” on page 234. The pre-upgrade and post-upgrade considerations are the same, except you substitute Release 2 wherever Release 3 is referenced. The specific upgrade procedures, however, are described in the following sections.

Release 2 Upgrade Procedure (Solaris)

The procedure documented below applies to the Directory Preparation Tool installed on the computer where Directory Server resides.

1. Obtain the required genesis patch, based on the following table:

Table 12-5 Genesis Patches[1] to Upgrade Directory Preparation Tool on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Directory Preparation Tool (DSSETUP)	118242-03	118243-03

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

   Patches can be downloaded to `/tmp` from:
   http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Apply the Directory Preparation Tool genesis patch in Table 12-5.

   ```
   patchadd patch_ID
   ```

4. Obtain the required upgrade patch, based on Table 12-3.

   Patches can be downloaded to `/tmp` from:
   http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

5. Apply the appropriate Directory Preparation Tool upgrade patch in Table 12-3.

   ```
   patchadd patch_ID
   ```

6. Confirm that the patch upgrade was successful:

   ```
   showrev -p | grep patch_ID
   ```

   The output should return the versions of patch IDs applied in Step 5.

7. Run the Directory Preparation Tool against Directory Server.

   a. Confirm that Directory Server is running.

   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`

   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      ```
      perl comm_dssetup.pl
      ```

      Provide the parameters requested by the script.

Release 2 Upgrade Procedure (Linux)

The procedure documented below applies to the Directory Preparation Tool installed on the computer where Directory Server resides.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Obtain the required upgrade patch using the patch number and RPM name from Table 12-4 on page 238.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

3. Apply the upgrade RPMs for Directory Preparation Tool in Table 12-4 on page 238.

   ```
   rpm -Uvh sun-comms-dssetup-6.3-2.5.i386.rpm
   ```

4. Confirm that the upgrade was successful:

   ```
   rpm -q sun-comms-dssetup
   ```

   The new version number of the RPM should be returned.

5. Run the Directory Preparation Tool against Directory Server.
   a. Confirm that Directory Server is running.
   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`
   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      `perl comm_dssetup.pl`

      Provide the parameters requested by the script.

Chapter 13

Messaging Server

This chapter describes how to upgrade Messaging Server to Java ES 2005Q4 (Release 4): Sun Java System Messaging Server 6.2 2005Q4.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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

The default value of *MessagingServer-base* depends on operating system platform:

- Solaris: `/opt/SUNWmsgsr`
- Linux: `/opt/sun/messaging`

Overview of Messaging Server Upgrades

This section describes the following general aspects of Messaging Server that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Messaging Server

Java ES Release 4 Messaging Server mostly represents bug fixes. There is no major new functionality with respect to Release 3.

Messaging Server Upgrade Roadmap

Table 13-1 shows the supported Messaging Server upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 13-1 Upgrade Paths to Java ES Release 4: Sun Java System Messaging Server 6.2 2005Q4

Java ES Release	Messaging Server Version	General Approach	Re-configuration Required
Release 3	Sun Java System Messaging Server 6.2 2005Q1	Direct upgrade: Performed by applying patches.	Configuration files and configuration directory data
Release 2	Sun Java System Messaging Server 6.1 2004Q2	Direct upgrade: Performed by applying patches.	Configuration files and configuration directory data
Release 1	Sun ONE Messaging Server 6.0 (2003Q4)	No direct upgrade: But you can upgrade first to Release 3 using procedures in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`). Then upgrade from Release 3 to Release 4.	Configuration files and configuration directory data

Table 13-1 Upgrade Paths to Java ES Release 4: Sun Java System Messaging Server 6.2 2005Q4 *(Continued)*

Java ES Release	Messaging Server Version	General Approach	Re-configuration Required
Pre-dates Java ES releases	Sun ONE Messaging Server 5.2	No direct upgrade: But you can upgrade first to Release 3 using procedures in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062). Then upgrade from Release 3 to Release 4.	Configuration files and configuration directory data

Messaging Server Data

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

Table 13-2 Messaging Server Data Usage

Type of Data	Location	Usage
Configuration data	Local configuration directory: *MessagingServer-base*/config/msg.conf and many other configuration files for configuring Messaging Server Store, MTA, MMP, MEM (webmail)	Configuration of Messaging Server components
Configuration data	Directory Server configuration directory	Configuration of Messaging Server components
User data	Directory Server user/group directory	Storing user attributes needed to support messaging for end users
Dynamic application data	Messaging Server store: *MessagingServer-base*/	Store email messages, message transfer queues, and related information on behalf of users
Directory schema	Directory Server /var/opt/mps/serverroot	For user attributes needed to support end users

Compatibility Issues

Release 4 Messaging Server does not introduce any interface changes. The Messaging Server Store, MTA, MMP, and MEM components, logically distinct configurations of Messaging Server, are backwardly compatible with earlier versions.

Messaging Server Dependencies

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

Messaging Server has dependencies on the following Java ES components:

- **Shared components.** Messaging Server has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Directory Server.** Messaging Server stores configuration data and user data needed for messaging in Directory Server. As a result, Messaging Server upgrades might require extensions of directory schema.
- **Directory Preparation Tool.** Messaging Server uses the Directory Preparation Tool to prepare Directory Server to support Messaging Server functions.
- **Access Manager (optional).** For software solutions that support single user sign-on for web-based services, Messaging Server can be configured to use Access Manager single sign-on capability.
- **Delegated Administrator (optional).** Delegated Administrator is the preferred utility to use for provisioning users in Directory Server so that Messaging Server has access to the user data needed to provide messaging services.

Upgrading Messaging Server from Java ES Release 3

This section includes information about upgrading Messaging Server from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Messaging Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. Re-configuration is achieved by running two data configuration utilities and by importing configuration data into Directory Server.
- **Upgrade Dependencies.** While Messaging Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Messaging Server requires that NSS, NSPR, LDAP C SDK, ICU, and SASL be upgraded to Release 4. Upgrade of J2SE is optional with respect to upgrade of Messaging Server to Release 4.

  In addition, Release 4 Messaging Server is dependent upon Directory Server and optionally dependent on Access Manager, as described in "Messaging Server Dependencies" on page 248. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Messaging Server to Release 4.

  However, Release 4 Messaging Server has a hard upgrade dependency on Directory Preparation Tool; Release 4 Directory Preparation Tool is required to prepare Directory Server for messaging operations.
- **Backward Compatibility.** Release 4 Messaging Server is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Messaging Server to Release 3 is achieved by first removing the changes made to Directory Server, removing changes to local configuration files, and removing the patches applied during the upgrade.

- **Platform Issues.** The general approach for upgrading Messaging Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Messaging Server Upgrade

This section describes how to perform an upgrade of Messaging Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Messaging Server by entering the following command:

MessagingServer-base`/sbin/imsimta version`

Table 13-3 Messaging Server Version Verification Outputs

Java ES Release	Messaging Server Version Number
Release 2	`6.1`
Release 3	`6.2`
Release 4	`6.2p3`

Upgrade Messaging Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, Messaging Server has hard upgrade dependencies only on the SASL shared component and on Directory Preparation Tool. Upgrading of otherJava ES Release 3 components upon which Messaging Server depends is therefore optional.

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

1. **Shared Components.** Instructions for upgrading NSS, NSPR, LDAP C SDK, ICU, and SASL shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, “Directory Server and Administration Server” on page 103.
3. **Access Manager (optional).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, “Access Manager” on page 203.
4. **Directory Preparation Tool.** Release 4 Directory Preparation Tool needs to have been run against Directory Server before configuring Release 4 Messaging Server. If Release 4 Directory Preparation Tool has not already been run against Directory Server, upgrade Directory Preparation Tool to Release 4 and use it to modify and extend the schema of Directory Server (see Chapter 12, “Directory Preparation Tool” on page 231 for procedures).

Back Up Messaging Server Data

The Messaging Server upgrade from Release 3 to Release 4 requires re-configuration of Messaging Server in local configuration files and in the Directory Server configuration directory. The local changes can be rolled back, but it is a good idea to back up the configuration directory in case you want to roll back the Release 4 upgrade at a future point.

Obtain Required Configuration Information and Passwords

Messaging Server upgrade requires knowing the following information:

- Superuser password
- Directory Manager DN and password

Upgrading Release 3 Messaging Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Messaging Server followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Messaging Server software to Java ES Release 4 takes into account the following considerations:

- All Messaging Server components, such as Messaging Server Store, MTA, MMP, or MEM, that correspond to the same installed Messaging Server image, are upgraded at the same time. All such components should be shut down before patches are applied to the installed image.
- The Release 4 Messaging Server upgrade patches for Solaris OS are shown in the following table:

Table 13-4 Patches[1] to Upgrade Messaging Server on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Messaging Server core	118207-38	118208-38
Messaging Server localization	117784 -15	117785 -15

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to all Messaging Server components that correspond to the same installed Messaging Server image on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 13-4.

Patches can be downloaded to `/tmp` from:
`http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

```
su -
```

3. Stop all running Messaging Server components.

MessagingServer-base`/sbin/stop-msg`

4. If you have not already done so, upgrade the required shared components to Release 4.

   "Upgrade Messaging Server Dependencies" on page 251.

5. Apply the appropriate Messaging Server patches in Table 13-4.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Migrate configuration data from existing configuration files to Release 4 configuration files.

   a. Create candidate configuration files.

      `cd` *MessagingServer-base*`/sbin`
      `./patch-config` *MessagingServer-base*`/install/patch/`*patch_ID*

      This command backs up existing configuration files. Then it merges configuration parameter values in these files with Release 4 template configuration files to create new Release 4 candidate configuration files. You should examine these new files for possible conflicts, as described in the Special Installation Instructions section of the patch `118209` `readme` file.

      This command also generates the following `ldif` files (LDAP directory import files):

      MessagingServer-base`/lib/patch/cfgdir_diff.ldif`
      MessagingServer-base`/lib/patch/ugdir_diff.ldif`

   b. Install the Release 4 candidate configuration files, making them the active configuration.

      `./install-newconfig` *MessagingServer-base*`/install/patch/`*patch_ID*

      This command installs the new Release 4 configuration files in their correct Release 4 locations.

      Note: If the `install-newconfig` command fails on the Solaris 10 platform, set the library path to null when running the command:

      ```
      LD_LIBRARY_PATH= ./install-newconfig
          MessagingServer-base/install/patch/patch_ID
      ```

c. Import the new configuration data generated in Step a on page 253 into the Directory Server configuration directory being used by Messaging Server.

Change to the configuration directory and import the `ldif` files using the `ldapmodify` command:

```
cd /MessagingServer-base/config/lib
```

```
./ldapmodify -D bind_dn -w password -c
    -e patch/cfgdir_diff.rej -f patch/cfgdir_diff.ldif
```

```
./ldapmodify -D bind_dn -w password -c
    -e patch/ugdir_diff.rej -f patch/ugdir_diff.ldif
```

8. Restart the Messaging Server components that were stopped in Step 3.

MessagingServer-base`/sbin/start-msg`

Upgrading Release 3 Messaging Server (Linux)

This section discusses considerations that impact the upgrade procedure for Messaging Server followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Messaging Server software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 252), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Messaging Server upgrade patches for Linux OS are shown in the following table:

Table 13-5 Patches[1] to Upgrade Messaging Server on Linux

Description	Patch ID and RPM names
Messaging Server core software with S/MIME	118209-38 • `sun-messaging-server-6.1-12.38.i386.rpm`
Messaging Server localization	117786-15 • `sun-messaging-l10n-`*Locale*`-6.1-8.15.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to all Messaging Server components that correspond to the same installed Messaging Server image on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 13-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop all running Messaging Server components.

   MessagingServer-base`/sbin/stop-msg`

4. If you have not already done so, upgrade the required shared components to Release 4.

   "Upgrade Messaging Server Dependencies" on page 251.

5. Apply the RPMs for Messaging Server in Table 13-5.

   For example:

   ```
   rpm -Fvh sun-messaging-server-6.1-12.38.i386.rpm
   ```

6. Confirm that the patch upgrade was successful:

   ```
   rpm -q sun-messaging-server
   ```

   The new version number of the RPM should be returned.

7. Migrate configuration data from existing configuration files to Release 4 configuration files.

   a. Create candidate configuration files.

      cd *MessagingServer-base*/sbin
      ./patch-config *MessagingServer-base*/install/patch/*patch_ID*

      This command backs up existing configuration files. Then it merges configuration parameter values in these files with Release 4 template configuration files to create new Release 4 candidate configuration files. You should examine these new files for possible conflicts, as described in the Special Installation Instructions section of the patch 118209 `readme` file.

      This command also generates the following `ldif` files (LDAP directory import files):

      MessagingServer-base/lib/patch/cfgdir_diff.ldif
      MessagingServer-base/lib/patch/ugdir_diff.ldif

   b. Install the Release 4 candidate configuration files, making them the active configuration.

      ./install-newconfig *MessagingServer-base*/install/patch/*patch_ID*

      This command installs the new Release 4 configuration files in their correct Release 4 locations.

   c. Import the new configuration data generated in Step a on page 256 into the Directory Server configuration directory being used by Messaging Server.

      Change to the configuration directory and import the `ldif` files using the `ldapmodify` command:

      cd /*MessagingServer-base*/config/lib

      ./ldapmodify -D *bind_dn* -w *password* -c
          -e patch/cfgdir_diff.rej -f patch/cfgdir_diff.ldif

      ./ldapmodify -D *bind_dn* -w *password* -c
          -e patch/ugdir_diff.rej -f patch/ugdir_diff.ldif

8. Restart the Messaging Server components that were stopped in Step 3.

   MessagingServer-base/sbin/start-msg

Verifying the Upgrade

You can verify the current version of Messaging Server by entering the following command:

MessagingServer-base/sbin/imsimta version

You can also check the banner displayed when starting up Messaging Server components

See Table 13-3 on page 250 for output values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 252 and "Upgrade Procedure (Linux)" on page 255.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Messaging Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Messaging Server is pretty much the reverse of the procedure for upgrading to Release 4. The re-configurations are rolled back and the patches are removed.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop all running Messaging Server components.

   MessagingServer-base/sbin/stop-msg

3. Roll back the changes made to the Directory Server configuration directory being used by Messaging Server.

   Replace the directory with the pre-upgrade directory that you backed up before beginning the upgrade procedure (see "Back Up Messaging Server Data" on page 251).

4. Roll back the re-configuration performed in Step 7 on page 253.

   ```
   cd MessagingServer-base/sbin
   ./uninstall-newconfig MessagingServer-base/install/patch/patch_ID
   ```

5. Remove the patches in Table 13-4 on page 252.

   ```
   patchrm patch_ID
   ```

6. Restart the Messaging Server components that were stopped in Step 2.

   ```
   MessagingServer-base/sbin/start-msg
   ```

Multiple Instance Upgrades

In some deployment architectures Messaging Server is deployed on multiple computer systems to provide for high availability and scalability. For example, you might have Messaging Server MTA or Messaging Server MMP components running on multiple computers with a load balancer to distribute the load. You might also have the Messaging Server Store component running in a Sun Cluster environment to provide high availability.

In the case of load-balanced instances of Messaging Server, you can perform a rolling upgrade in which you upgrade the Messaging Server instances sequentially without interrupting service. You upgrade each instance of Messaging Server while the others remain running. In deployment architectures in which various MS subcomponents (MS Store, MTA, MMP, MEM) are deployed on different computers, upgrade components beginning in the back-end tier (MS Store) and working toward the front-end tier (such as MEM). You perform the upgrade of each instance as described in "Release 3 Messaging Server Upgrade" on page 250.

In the case of Messaging Server instances running in a cluster environment, those instances share the same configuration. You therefore need to apply Messaging Server upgrade patches to each of the instances, but you only need to perform the re-configuration part of the upgrade procedure once, after patches have been applied to all the instances.

Upgrading Messaging Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Messaging Server to Release 4 is the same as that for upgrading Release 3 Messaging Server to Release 4, with a couple of exceptions, noted below.

Upgrade Messaging Server Dependencies

As compared to the upgrade from Release 3, the Release 2 to Release 4 pre-upgrade tasks should include the upgrading to Release 4 of all shared components (see Table 1-6 on page 40) and all locally-resident product components upon which Messaging Server depends:

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Directory Server rarely resides on the same computer as Messaging Server, however, instructions for upgrading Directory Server to Release 4 are provided in "Upgrading Directory Server and Administration Server from Java ES Release 2" on page 122.
3. **Access Manager (optional).** Instructions for upgrading Access Manager to Release 4 are provided in "Upgrading Access Manager from Java ES Release 2" on page 224.
4. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Messaging Server, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in "Upgrading Directory Preparation Tool from Java ES Release 2" on page 240.

Release 2 Messaging Server Upgrade

The procedure for upgrading Messaging Server from Release 2 to Release 4 depends on operating system platform.

Upgrading Release 2 Messaging Server (Solaris)

To upgrade Release 2 Messaging Server to Release 4, use the instructions in "Upgrading Release 2 Messaging Server (Solaris)" on page 260, except substitute Release 2 wherever Release 3 is referenced.

Upgrading Release 2 Messaging Server (Linux)

The procedure documented below applies to all Messaging Server components that correspond to the same installed Messaging Server image on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

---

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop all running Messaging Server components.

   MessagingServer-base`/sbin/stop-msg`

3. If you have not already done so, upgrade the required shared components to Release 4.

   See "Upgrade Messaging Server Dependencies" on page 259.

4. Uninstall the Release 2 RPM packages.

   ```
   rpm -e --noscripts sun-messaging-lib-6.1-9 \
                      sun-messaging-store-6.1-9 \
                      sun-messaging-install-6.1-9 \
                      sun-messaging-core-6.1-9 \
                      sun-messaging-mmp-6.1-9 \
                      sun-messaging-sieveui-6.1-9 \
                      sun-messaging-webmail-6.1-9 \
                      sun-messaging-core-en-6.1-9 \
                      sun-messaging-mta-6.1-9
   ```

5. Install the RPM for Messaging Server in Table 13-5 on page 254.

   ```
   rpm -i sun-messaging-server-6.1-12.38.i386.rpm
   ```

6. Confirm that the patch upgrade was successful:

   ```
   rpm -q sun-messaging-server
   ```

   The version number of the newly installed RPM should be returned.

7. Save off your old Release 2 configuration.

   The configuration files are located at: *MessagingServer-base*`/config`

8. Run the Messaging Server configuration program.

   ```
   cd MessagingServer-base/sbin
   ./configure
   ```

9. Perform a manual merge of the Release 2 configuration values with the new Release 4 configuration entries.

10. Restart the Messaging Server components that were stopped in Step 2.

    MessagingServer-base`/sbin/start-msg`

For further details, for example to change the HTTP port using the `configutil` command, see the Special Installation Instructions section of the patch 118209-38 `readme` file.

Verifying the Upgrade

You can verify the current version of Messaging Server by entering the following command:

MessagingServer-base`/sbin/imsimta version`

You can also check the banner displayed when starting up Messaging Server components

See Table 13-3 on page 250 for output values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure (Solaris)” on page 252 and “Upgrade Procedure (Linux)” on page 255.

Chapter 14

Calendar Server

This chapter describes how to upgrade Calendar Server to Java ES 2005Q4 (Release 4): Sun Java System Calendar Server 6.2 2005Q4. The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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

The default value of *CalendarServer-base* depends on operating system platform:

- Solaris: `/opt/SUNWics5`
- Linux: `/opt/sun/calendar`

Overview of Calendar Server Upgrades

This section describes the following general aspects of Calendar Server that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Calendar Server

Java ES Release 4 Calendar Server mostly represents bug fixes. There is no major new functionality with respect to Release 3.

Calendar Server Upgrade Roadmap

Table 14-1 shows the supported Calendar Server upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 14-1 Upgrade Paths to Java ES Release 4: Sun Java System Calendar Server 6.2 2005Q4

Java ES Release	Calendar Server Version	General Approach	Re-configuration Required
Release 3	Sun Java System Calendar Server 6 2005Q1	Direct upgrade: Perform by applying patches.	None
Release 2	Sun Java System Calendar Server 6 2004Q2	Direct upgrade: Perform by applying patches.	Configuration files
Release 1	Sun ONE Calendar Server 6.0 (2003Q4)	Direct upgrade not certified: But can be performed by applying patches.	Configuration files
Pre-dates Java ES releases	All previous versions	No direct upgrade.	

Calendar Server Data

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

Table 14-2 Calendar Server Data Usage

Type of Data	Location	Usage
Configuration data	`etc/`*CalendarServer-base*`/cal/config/ics.conf`	Configuration of Calendar Server
Dynamic application data	Calendar Server database: `/var/`*CalendarServer-base*`/csdb`	Store calendar entries on behalf of users.
Directory schema	Directory Server user/group directory	For user attributes needed to support end users

Compatibility Issues

Release 4 Calendar Server does not introduce any interface changes. Calendar Server is backwardly compatible with earlier versions.

Calendar Server Dependencies

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

Calendar Server has dependencies on the following Java ES components:

- **Shared components.** Calendar Server has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Directory Server.** Calendar Server accesses user data stored in Directory Server. As a result, Calendar Server upgrades might require extensions of directory schema.
- **Directory Preparation Tool.** Calendar Server uses the Directory Preparation Tool to prepare the directory to support Calendar Server functions.

- **Access Manager (optional).** For software solutions that support single user sign-on for web-based services, Calendar Server can be configured to use Access Manager single sign-on capability.
- **Messaging Server (optional).** Calendar Server can be configured to use Messaging Server to provide messaging notifications of calendar events.
- **Delegated Administrator (optional).** Delegated Administrator is the preferred utility to use for provisioning users in Directory Server so that Calendar Server has access to the user data needed to provide calendar services.

Upgrading Calendar Server from Java ES Release 3

This section includes information about upgrading Calendar Server from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Calendar Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version.
- **Upgrade Dependencies.** While Calendar Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Calendar Server requires only that SASL be upgraded to Release 4. Upgrade of other shared components is optional with respect to upgrade of Calendar Server.

In addition, Release 4 Calendar Server is dependent upon Directory Server and optionally dependent on Access Manager, as described in "Calendar Server Dependencies" on page 265. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Calendar Server to Release 4.

However, Release 4 Calendar Server has a hard upgrade dependency on Directory Preparation Tool; Release 4 Directory Preparation Tool is required to prepare Directory Server for calendaring operations.

- **Backward Compatibility.** Release 4 Calendar Server is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Calendar Server to Release 3 is achieved by removing the patches applied during the upgrade.
- **Platform Issues.** The general approach for upgrading Calendar Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Calendar Server Upgrade

This section describes how to perform an upgrade of Calendar Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Calendar Server using the following command:

Solaris:

```
cd CalendarServer-base/cal/bin
./cshttpd -#
```

Linux:

```
cd CalendarServer-base/bin
./cshttpd -#
```

Note: If the `cshttpd` command fails on the Solaris 10 platform, set the library path to null when running the command:

```
LD_LIBRARY_PATH= ./cshttpd -#
```

Table 14-3 Calendar Server Version Verification Outputs

Java ES Release	Calendar Server Version Number
Release 2	2004Q2
Release 3	2005Q1

Apply Necessary Operating System Patches

On Solaris 10 operating system platforms, you need to apply an operating system patch to perform the Delegated Administrator upgrade procedure (see "Required Operating System Patches" on page 31).

Upgrade Calendar Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, Calendar Server has a hard upgrade dependency only on Directory Preparation Tool. Upgrading of other Java ES Release 3 components upon which Calendar Server depends is therefore optional.

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

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, “Directory Server and Administration Server” on page 103.
3. **Access Manager (optional).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, “Access Manager” on page 203.
4. **Directory Preparation Tool.** Release 4 Directory Preparation Tool needs to have been run against Directory Server before configuring Release 4 Calendar Server. If Release 4 Directory Preparation Tool has not already been run against Directory Server, upgrade Directory Preparation Tool to Release 4 and use it to modify and extend the schema of Directory Server (see Chapter 12, “Directory Preparation Tool” on page 231 for procedures).

Back Up Calendar Server Data

The Calendar Server upgrade from Release 3 to Release 4 requires no re-configuration of Calendar Server. However, as a safety precaution, you might back up your Calendar Server store, located at

`/var/`*CalendarServer-base*`/csdb`

Obtain Required Configuration Information and Passwords

No special information about your currently installed version is needed. However you will have to log in as superuser to perform the upgrade.

Upgrading Release 3 Calendar Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Calendar Server followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Calendar Server software to Java ES Release 4 takes into account the following considerations:

- Calendar Server should be shut down when patches are being applied to the installed image.
- In architectures in which different Calendar Server subcomponents reside on different computers, for example Calendar Server back-end store on one computer, and Calendar Server front-end processes (such as cshttpd) on another, the upgrade must be performed on all such computers.

- The Calendar Server upgrade applies to multiple subcomponents of Calendar Server on one computer using the same installed image.
- The Release 4 Calendar Server upgrade patches for Solaris OS are shown in the following table:

Table 14-4 Patches[1] to Upgrade Calendar Server on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Calendar Server core	116577-24	116578-24
Calendar Server localization	117010 -23	117011 -23

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Calendar Server on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 14-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Calendar Server if it is running.

   CalendarServer-base`/cal/sbin/stop-cal`

4. If you have not already done so, upgrade the SASL shared component to Release 4 and any other shared components you wish to upgrade.

   See "Upgrade Calendar Server Dependencies" on page 268.

5. Apply the appropriate Calendar Server patches in Table 14-4.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrade was successful:

   ```
   showrev -p | grep ics
   ```

   The output should return the versions of patch IDs applied in Step 5.

7. Restart the Calendar Server that was stopped in Step 3.

   CalendarServer-base`/cal/sbin/start-cal`

Upgrading Release 3 Calendar Server (Linux)

This section discusses considerations that impact the upgrade procedure for Calendar Server followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Calendar Server software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 269), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Calendar Server upgrade patches for Linux OS are shown in the following table:

Table 14-5 Patches[1] to Upgrade Calendar Server on Linux

Description	Patch ID and RPM names
Calendar Server core	117851-24
	• `sun-calendar-core-6.2-10.7.i386.rpm`
	• `sun-calendar-api-6.2-10.7.i386.rpm`
Calendar Server Locale	117852-23
	• `sun-calendar-core-`*Locale*`-6.2-10.3.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to Calendar Server on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 14-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Calendar Server if it is running.

   CalendarServer-base`/sbin/stop-cal`

4. If you have not already done so, upgrade the SASL shared component to Release 4 and any other shared components you wish to upgrade.

   See "Upgrade Calendar Server Dependencies" on page 268.

5. Apply the RPMs for Calendar Server in Table 14-5.

   ```
   rpm -Fvh sun-calendar-core-Locale-6.2-10.3.i386.rpm
   rpm -Fvh sun-calendar-core-6.2-10.7.i386.rpm
   rpm -Fvh sun-calendar-api-6.2-10.7.i386.rpm
   ```

6. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-calendar
   ```

   The new version numbers of the RPMs should be returned.

7. Restart the Calendar Server that was stopped in Step 3.

   CalendarServer-base`/sbin/start-cal`

Verifying the Upgrade

The upgrade of Calendar Server to Release 4 is verified by confirming that the upgrade patches have been properly applied. The steps for this verification were included in "Upgrade Procedure (Solaris)" on page 270 and "Upgrade Procedure (Linux)" on page 271.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 270 and "Upgrade Procedure (Linux)" on page 271.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Calendar Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Calendar Server is pretty much the reverse of the procedure for upgrading to Release 4.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   `su -`

2. Stop Calendar Server.

   CalendarServer-base`/cal/sbin/stop-cal`

3. Remove the patches in Table 14-4 on page 270.

   `patchrm` *patch_ID*

4. Restart Calendar Server.

   CalendarServer-base`/cal/sbin/start-cal`

Multiple Instance Upgrades

In some deployment architectures Calendar Server is deployed on multiple computer systems to provide for high availability. For example, you have the Calendar Server Store component running in a Sun Cluster environment to provide high availability.

For Calendar Server instances running in a cluster environment, those instances share the same configuration. You need to apply Calendar Server upgrade patches to each of the instances, and for a Release 3 to Release 4 upgrade there is no re-configuration required.

Upgrading Calendar Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Calendar Server to Release 4 is very similar to that for upgrading Release 3 Calendar Server to Release 4, with the exception that the pre-upgrade tasks should include the upgrading to Release 4 of all shared components (see Table 1-6 on page 40) and all locally-resident product components upon which Calendar Server depends:

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Directory Server rarely resides on the same computer as Calendar Server, however, instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, "Directory Server and Administration Server" on page 103.
3. **Access Manager (optional).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, "Access Manager" on page 203.
4. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Calendar Server, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in Chapter 12, "Directory Preparation Tool" on page 231.

To upgrade Release 2 Calendar Server to Release 4, use the instructions in "Upgrading Calendar Server from Java ES Release 3" on page 266, except substitute Release 2 wherever Release 3 is referenced.

In addition, the Release 2 to Release 4 upgrade requires the post-upgrade task of configuring Calendar Server hot backup, accomplished by adding hotbackup parameters to the Calendar Server `ics.conf` configuration file. The instructions for this post-upgrade re-configuration can be found at the following location:
`http://docs.sun.com/doc/819-2433/6n4nlfjnq?a=view`

Chapter 15

Communications Express

This chapter describes how to upgrade Communications Express to Java ES 2005Q4 (Release 4): Sun Java System Communications Express 6.2 2005Q4. The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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

The default value of *CommsExpress-base* depends on operating system platform:

- Solaris: `/opt/SUNWuwc`
- Linux: `/opt/sun/uwc`

Overview of Communications Express Upgrades

This section describes the following general aspects of Communications Express that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Communications Express

Java ES Release 4 Communications Express mostly represents bug fixes. There are a few new features with respect to Release 3: mail filter support, address book sharing, and proxy authentication.

Communications Express Upgrade Roadmap

Table 15-1 shows the supported Communications Express upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 15-1 Upgrade Paths to Java ES Release 4: Sun Java System Communications Express 6.2 2005Q4

Java ES Release	Communications Express Version	General Approach	Re-configuration Required
Release 3	Sun Java System Communications Express 6 2005Q1	Direct upgrade: Perform by applying patches and re-configuring Messaging Server component.	Configuration files
Release 2	Sun Java System Communications Express 6 2004Q2	Direct upgrade: Perform by applying patches and re-configuring Messaging Server component.	Configuration files
Release 1	None	No upgrade:	

Table 15-1 Upgrade Paths to Java ES Release 4: Sun Java System Communications Express 6.2 2005Q4 *(Continued)*

Java ES Release	Communications Express Version	General Approach	Re-configuration Required
Pre-dates Java ES releases	None	No upgrade:	

Communications Express Data

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

Table 15-2 Communications Express Data Usage

Type of Data	Location	Usage

Configuration data:	Local configuration directory
	`var/`*CommsExpress-base*`/WEB-INF/config/uwcauth.properties`
	`var/`*CommsExpress-base*`/WEB-INF/config/uwcconfig.properties`
	`var/`*CommsExpress-base*`/WEB-INF/config/uwclogging.properties`
	MessagingServer-base`/config/msg.conf`
	and other configuration files for configuring Messaging Server MEM (webmail)
		Configuration of Communications Express, including Messaging Server MEM (webmail)

Web container configuration	Web Server: `server.policy` and `server.xml` files in *WebServer-base*`/https-`*hostname*`/config` Application Server (Java ES Release 3 and 4): `server.policy` and `domain.xml` files in *AppServer8Config-base*`/domains/`*domainName*`/config` Application Server (Java ES Release 2): `server.policy` and `server.xml` files in *AppServer7Config-base*`/domains/`*domainName*`/config`	Configuration of Communications Express web container instance.
Directory schema	Directory Server user/group directory	For user attributes needed to support end users

Compatibility Issues

Release 4 Communications Express does not introduce any interface changes and is backwardly compatible with earlier versions.

Communications Express Dependencies

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

Communications Express has dependencies on the following Java ES components:

- **Shared components.** Communications Express has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Web Container.** Communications Express depends upon web container services, which can be provided either by Java ES Web Server or Java ES Application Server.
- **Access Manager (or Access Manager SDK).** Communications Express depends upon Access Manager to provide authentication and authorization services for end users, including single sign-on. If Access Manager is run on a remote computer, then Access Manager SDK must be available locally.
- **Messaging Server.** Communications Express is used to provide web-based access to Messaging Server. In fact Communications Express directly employs the Messaging Server MEM component to access other Messaging Server back-end components, such as the Messaging Server Store and MTA components.
- **Calendar Server.** Communications Express is used to provide web-based access to Calendar Server.
- **Directory Server.** Communications Express stores configuration data and also accesses user data stored in Directory Server. As a result, Communications Express upgrades might require upgrades of Directory Server or extensions of directory schema.
- **Directory Preparation Tool.** Communications Express uses the Directory Preparation Tool to prepare Directory Server to support Communications Express functions. As a result, Communications Express upgrades might depend upon preparation of the directory to support new functions.

Upgrading Communications Express from Java ES Release 3

This section includes information about upgrading Communications Express from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4). The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Communications Express to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. Re-configuration of the included Messaging Server MEM component is achieved using two configuration utilities and by importing configuration data into Directory Server.
- **Upgrade Dependencies.** While Calendar Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Calendar Server is compatible with the Release 3 versions of these components. Upgrade of these shared components is therefore optional with respect to upgrade of Calendar Server to Release 4.

  In addition, Release 4 Communications Express is dependent upon a web container and on Access Manager, as described in “Communications Express Dependencies” on page 278. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Communications Express to Release 4.

  However, Communications Express has hard upgrade dependencies on both Calendar Server, for which it provides web-based access, and on Messaging Server for which it also provides web-based access using the Messaging Server MEM component. Both Calendar Server and Messaging Server must therefore be upgraded to Release 4 before Communications Express can be upgraded to Release 4.

In addition, Release 4 Communications Express has a hard upgrade dependency on Directory Preparation Tool; Release 4 Directory Preparation Tool is required to prepare Directory Server for Communications Express functions.

- **Backward Compatibility.** Release 4 Communications Express is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Communications Express to Release 3 is achieved by first removing the changes made to Directory Server, removing changes to local configuration files, and removing the patches applied during the upgrade.
- **Platform Issues.** The general approach for upgrading Communications Express is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Communications Express Upgrade

This section describes how to perform an upgrade of Communications Express from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Communications Express you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Communications Express by accessing the Communications Express login page, which shows the current version number.

```
http://hostName:port/uwc/auth
```

Table 15-3 Communications Express Version Verification Outputs

Java ES Release	Communications Express Version Number
Release 2	`Sun Java System Communications Express 6 2004Q2`
Release 3	`Sun Java System Communications Express 6 2005Q1`
Release 4	`Sun Java System Communications Express 6 2005Q4`

Upgrade Communications Express Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, Communications Express has hard upgrade dependencies only on Messaging Server, Calendar Server, and Directory Preparation Tool. Upgrading of other Java ES Release 3 components upon which Communications Express depends is therefore optional.

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

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, “Directory Server and Administration Server” on page 103.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 6, “Web Server” on page 137 and Chapter 9, “Application Server” on page 175, respectively.

4. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, “Access Manager” on page 203.

5. **Directory Preparation Tool.** Release 4 Directory Preparation Tool needs to have been run against Directory Server before configuring Release 4 Communications Express. If Release 4 Directory Preparation Tool has not already been run against Directory Server, upgrade Directory Preparation Tool to Release 4 and use it to modify and extend the schema of Directory Server (see Chapter 12, “Directory Preparation Tool” on page 231 for procedures).

6. **Messaging Server.** Messaging Server components need to be upgraded to Release 4 to support Release 4 Communications Express. Instructions for upgrading Messaging Server to Release 4 are provided in Chapter 13, “Messaging Server” on page 245.

7. **Calendar Server.** Calendar Server components need to be upgraded to Release 4 to support Release 4 Communications Express. Instructions for upgrading Calendar Server to Release 4 are provided in Chapter 14, “Calendar Server” on page 263.

Back Up Communications Express Data

The Communications Express upgrade from Release 3 to Release 4 requires re-configuration of Messaging Server MEM. The local changes can be rolled back, so there is no need to back up any data.

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. If you are using Web Server as a web container, no configuration information is needed. However if you are using Application Server as a web container, you will need the Application Server administrator user ID and password.

Upgrading Release 3 Communications Express (Solaris)

This section discusses considerations that impact the upgrade procedure for Communications Express followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Communications Express software to Java ES Release 4 takes into account the following considerations:

- Communications Express includes components used to provide web-based access to Calendar Server and a Messaging Server MEM component used to provide web-based access to Messaging Server back-end components. The upgrade patches encompass all these components.

- All Communications Express components should be deployed to the same web container. The web container should be shut down before patches are applied to the installed image.
- The Release 4 Communications Express upgrade patches for Solaris OS are shown in the following table:

Table 15-4 Patches[1] to Upgrade Communications Express on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Communications Express core	118540-21	118541-21
Communications Express localization	118042-16	118042-16

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to all Communications Express components on the computer being updated.

1. Obtain the required patches, based on Table 15-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   `su -`

3. Stop Communications Express by stopping its web container.

   Web Server:
   WebServer-base`/https-`*instanceName*`/stop`

   Application Server:
   AppServer8-base`/bin/asadmin stop-domain` *domainName*

4. If you have not already done so, upgrade any shared components you wish to upgrade to Release 4.

   See "Upgrade Communications Express Dependencies" on page 281.

5. Apply the appropriate Communications Express patches in Table 15-4.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrade was successful:

   ```
   showrev -p | grep uwc
   ```

   The output should return the versions of patch IDs applied in Step 5.

7. Migrate configuration data from existing configuration files to Release 4 candidate configuration files.

   ```
   cd CommsExpress-base/sbin
   ./patch-config CommsExpress-base/install/patch/patch_ID
   ```

   This command prompts you for the current configuration directory and then backs up the existing configuration files. Then it merges configuration parameter values in these files with Release 4 template configuration files to create new Release 4 candidate configuration files. You should check these new files for possible conflicts, as described in the Special Installation Instructions section of the patch `readme` file.

8. Install the Release 4 candidate configuration files, making them the active configuration.

   ```
   ./install-newconfig CommsExpress-base/install/patch/patch_ID
   ```

   This command installs the new Release 4 configuration files in their correct Release 4 locations.

9. Remove the JSP class cache for Communications Express that is maintained by the web container.

   For the procedure, see the documentation for your web container (Web Server or Application Server).

10. Restart Communications Express by restarting its web container.

    Web Server:
    WebServer-base`/https-`*instanceName*`/start`

    Application Server:
    AppServer8-base`/bin/asadmin start-domain` *domainName*
    `--user` *admin_ID* `--password` *password*

Upgrading Release 3 Communications Express (Linux)

This section discusses considerations that impact the upgrade procedure for Communications Express followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Communications Express software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 282), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Communications Express upgrade patches for Linux OS are shown in the following table:

Table 15-5 Patches[1] to Upgrade Communications Express on Linux

Description	Patch ID and RPM names
Communications Express core	118542-21 • `sun-uwc-6.1-7.21.i386.rpm`
Communications Express localization	118044-14 • `sun-uwc-l10n-`*`Locale`*`-6.1-11.9.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to all Communications Express components on the computer being updated.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 15-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Communications Express by stopping its web container.

   Web Server:
   WebServer-base/`https-`*instanceName*/`stop`

   Application Server*:*
   AppServer8-base/`bin/asadmin stop-domain` *domainName*

4. If you have not already done so, upgrade any shared components you wish to upgrade to Release 4.

   See "Upgrade Communications Express Dependencies" on page 281.

5. Apply the RPMs for Communications Express in Table 15-5.

   For example:

   ```
   rpm -Fvh sun-uwc-6.1-7.21.i386.rpm
   ```

6. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep uwc
   ```

   The output should return the version of RPM in Step 5.

7. Migrate configuration data from existing configuration files to Release 4 candidate configuration files.

   `cd` *CommsExpress-base*`/sbin`
   `./patch-config` *CommsExpress-base*`/install/patch/`*patch_ID*

   This command prompts you for the current configuration directory and then backs up the existing configuration files. Then it merges configuration parameter values in these files with Release 4 template configuration files to create new Release 4 candidate configuration files. You should check these new files for possible conflicts, as described in the Special Installation Instructions section of the patch `readme` file.

8. Install the Release 4 candidate configuration files, making them the active configuration.

   `./install-newconfig` *CommsExpress-base*`/install/patch/`*patch_ID*

   This command installs the new Release 4 configuration files in their correct Release 4 locations.

9. Remove the JSP class cache for Communications Express that is maintained by the web container.

   For the procedure, see the documentation for your web container (Web Server or Application Server).

10. Restart Communications Express by restarting its web container.

Web Server:
WebServer-base/`https-`*instanceName*`/start`

Application Server:
AppServer8-base`/bin/asadmin start-domain` *domainName*
`--user` *admin_ID* `--password` *password*

Verifying the Upgrade

You can verify the current version of Communications Express by accessing the Communications Express login page, which shows the current version number.

`http://`*hostName*`:`*port*`/uwc/auth`

Once logged in, check the upgraded user interface for the new mail tab and old email and calendar events (if you are using those channels).

Also, you can check the log files for the various steps in the upgrade procedure:

CommsExpress-base`/install/patch/118540-21.`
CommsExpress-base`/patch-config_20050729164754.log`
CommsExpress-base`/install-newconfig_20050729164838.log`

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 283 and "Upgrade Procedure (Linux)" on page 285.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Communications Express followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Communications Express is pretty much the reverse of the procedure for upgrading to Release 4. The re-configurations are rolled back and the patches are removed.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop Communications Express by stopping its web container.

   Web Server:
   WebServer-base/https-*instanceName*/stop

   Application Server:
   AppServer8-base/bin/asadmin stop-domain *domainName*

3. Roll back the changes made to the Directory Server configuration directory being used by Communications Express.

4. Roll back the re-configuration performed in Step 8 on page 284.

   ```
   cd CommsExpress-base/sbin
   ./uninstall-newconfig CommsExpress-base/install/patch/patch_ID
   ```

5. Remove the patches in Table 15-4 on page 283.

   ```
   patchrm patch_ID
   ```

6. Restart Communications Express by restarting its web container.

   Web Server:
   WebServer-base/https-*instanceName*/start

   Application Server:
   AppServer8-base/bin/asadmin start-domain *domainName*
   --user *admin_ID* --password *password*

Multiple Instance Upgrades

In some deployment architectures Communications Express is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Communications Express components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Communications Express, you can perform a rolling upgrade in which you upgrade the Communications Express instances sequentially without interrupting service. You upgrade each instance of Communications Express while the others remain running. You perform the upgrade of each instance as described in "Release 3 Communications Express Upgrade" on page 280.

Upgrading Communications Express from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Communications Express to Release 4 is the same as that for upgrading Release 3 Communications Express to Release 4, with a couple of exceptions, noted below.

NOTE This section applies to the case in which Communications Express is deployed in a Release 2 Web Server web container, but does not apply to the case in which Communications Express is deployed in a Release 2 Application Server web container. The latter case is not currently supported.

Upgrade Communications Express Dependencies

The pre-upgrade tasks for upgrading Java ES Release 2 Communications Express to Release 4 are similar to those for upgrading Release 3 Communications Express to Release 4, with the exception that the upgrade of Communications Express dependencies should include the upgrading to Release 4 of all shared components (see Table 1-6 on page 40) and all locally-resident product components upon which Communications Express depends.

When upgrading Communications Express dependencies, they should be upgraded in the following order, all before you upgrade Communications Express. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51.

2. **Directory Server.** Directory Server rarely resides on the same computer as Communications Express, however, instructions for upgrading Directory Server to Release 4 are provided in “Upgrading Directory Server and Administration Server from Java ES Release 2” on page 122.

3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in “Upgrading Web Server from Java ES Release 2” on page 147 and “Upgrading Application Server from Java ES Release 2” on page 188, respectively.

4. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 4 are provided in "Upgrading Access Manager from Java ES Release 2" on page 224.

5. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Communications Express, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in "Upgrading Directory Preparation Tool from Java ES Release 2" on page 240.

6. **Messaging Server.** Messaging Server MTA needs to be upgraded to Release 4 to support Release 4 Communications Express. Instructions for upgrading Messaging Server to Release 4 are provided in "Upgrading Messaging Server from Java ES Release 2" on page 259

7. **Calendar Server.** Calendar Server rarely resides on the same computer as Communications Express, however, instructions for upgrading Calendar Server to Release 4 are provided in "Upgrading Calendar Server from Java ES Release 2" on page 274

Release 2 Communications Express Upgrade

The procedure for upgrading Communications Express from Release 2 to Release 4 depends on the web container in which you are deploying Communications Express software.

Upgrading Release 2 Communications Express: Web Server Web Container

To upgrade Release 2 Communications Express to Release 4, when deploying into a Web Server web container that has been upgraded to Release 4, follow the instructions in "Upgrading Release 3 Communications Express (Solaris)" on page 282 or "Upgrading Release 3 Communications Express (Linux)" on page 284, except substitute Release 2 wherever Release 3 is referenced.

Upgrading Release 2 Communications Express: Application Server Web Container

The upgrade of Release 2 Communications Express to Release 4, when deploying into an Application Server web container that has been upgraded to Release 4, is not currently supported.

Chapter 16

Instant Messaging

This chapter describes how to upgrade Instant Messaging to Java ES 2005Q4 (Release 4): Sun Java System Instant Messaging 7.0.1 2005Q4. The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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

The default value of *InstantMessaging-base* depends on operating system platform:

- Solaris: `/opt/SUNWiim`
- Linux: `/opt/sun/im`

Overview of Instant Messaging Upgrades

This section describes the following general aspects of Instant Messaging that impact upgrading to Java ES 2005Q4 (Release 4):

- About Java ES Release 4 Instant Messaging
- Instant Messaging Upgrade Roadmap
- Instant Messaging Data
- Compatibility Issues
- Instant Messaging Dependencies

About Java ES Release 4 Instant Messaging

Java ES Release 4 Instant Messaging mostly represents bug fixes. There is no major new functionality with respect to Release 3.

Instant Messaging Upgrade Roadmap

Table 16-1 shows the supported Instant Messaging upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 16-1 Upgrade Paths to Java ES Release 4: Sun Java System Instant Messaging 7.0.1 2005Q4

Java ES Release	Instant Messaging Version	General Approach	Re-configuration Required
Release 3	Sun Java System Instant Messaging 7.0 2005Q1	Direct upgrade: Performed by applying patches.	None
Release 2	Sun Java System Instant Messaging 6 2004Q2	Direct upgrade: Performed using the `upgrade` utility.	Configuration data
Release 1	Sun Java System Instant Messaging 6.1 (2003Q4)	Direct upgrade not certified: But can be performed using the `upgrade` utility.	Configuration data
Pre-dates Java ES releases		No direct upgrade	

Instant Messaging Data

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

Table 16-2 Instant Messaging Data Usage

Type of Data	Location	Usage

Configuration data:	Local configuration directory
	Solaris:
	`/etc/opt/SUNWiim/default/config/iim.conf`
	`/etc/opt/SUNWiim/default/config/registration.properties`
	Linux:
	`/etc/opt/sun/im/default/config/iim.conf`
	`/etc/opt/sun/im/default/config/registration.properties`
		Configuration of Instant Messaging processes and registration attributes

Instant Messaging Server Resources	Local configuration directory *InstantMessaging-base*`/html`	customized client files downloaded by end users to launch the Messenger client.
Dynamic data	*runtimeFilesDir*`/default/db` where *runtimeFilesDir* is specified at installation: Solaris: default *runtimeFilesDir*: `/var/opt/SUNWiim` Linux: default *runtimeFilesDir*: `/var/opt/sun/im`	All variable data, such as the Instant Messaging database, log files, and lock files.

Compatibility Issues

Release 4 Instant Messaging does not introduce any interface changes and is backwardly compatible with earlier versions.

Instant Messaging Dependencies

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

Instant Messaging has dependencies on the following Java ES components:

- **Shared components.** Instant Messaging has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Web Container.** Instant Messaging depends upon web container services, which can be provided either by Java ES Web Server or Java ES Application Server.
- **Directory Server (optional).** Instant Messaging can be configured to store and access user data in Directory Server. As a result, Instant Messaging upgrades might require extensions of directory schema.
- **Access Manager (optional).** For software solutions that support single user sign-on for web-based services, Instant Messaging can be configured to use Access Manager single sign-on capability.

Upgrading Instant Messaging from Java ES Release 3

This section includes the following information about upgrading Instant Messaging from Java ES Release 3 to Java ES Release 4. The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Instant Messaging to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. There is no re-configuration of Instant Messaging required.

- **Upgrade Dependencies.** While Instant Messaging has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Instant Messaging requires only that IM-SDK be upgraded to Release 4. Upgrade of other shared components is optional with respect to upgrade of Instant Messaging to Release 4.

  In addition, Release 4 Instant Messaging is dependent on a web container and optionally dependent on Access Manager, as described in "Instant Messaging Dependencies" on page 293. However, these are soft upgrade dependencies; upgrade the web container and Access Manager is optional with respect to upgrade of Instant Messaging to Release 4.

- **Backward Compatibility.** Release 4 Instant Messaging is backwardly compatible with the Release 3 version.

- **Upgrade Rollback.** Rollback of the Release 4 upgrade to Release 3 is achieved on Solaris platforms by removing the patches applied during the upgrade.

- **Platform Issues.** The general approach for upgrading Instant Messaging is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Instant Messaging Upgrade

This section describes how to perform an upgrade of Instant Messaging from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Instant Messaging you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Instant Messaging using standard version checking utilities

Solaris:

```
pkginfo -l SUNWiimin
```

Linux:

```
rpm -qa | grep sun-im
```

Table 16-3 Instant Messaging Version Verification Outputs

Java ES Release	Instant Messaging Version Number
Release 2	Version numbers `6.x`
Release 3	Version numbers `7.0`
Release 4	Version numbers `7.0.1`

Upgrade Instant Messaging Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, Instant Messaging has a hard upgrade dependency only on the IM-SDK shared component. Upgrading of otherJava ES Release 3 components upon which Instant Messaging depends is therefore optional.

If you choose to upgrade all Instant Messaging dependencies, they should be upgraded in the following order, all before you upgrade Instant Messaging. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading IM-SDK and other Java ES shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51.
2. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 6, “Web Server” on page 137 and Chapter 9, “Application Server” on page 175, respectively.
3. **Access Manager (optional).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, “Access Manager” on page 203.

Back Up Instant Messaging Data

The Instant Messaging upgrade from Release 3 to Release 4 does not modify configuration data. However, as a safety measure it is a good idea to back up the Instant Messaging database and any existing resource and configuration files you have customized before upgrading Instant Messaging. For more information, see the *Sun Java System Instant Messaging 7 2005Q4 Administration Guide* (http://docs.sun.com/doc/819-2503).

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. If you are using Web Server as a web container, no configuration information is needed. However if you are using Application Server as a web container, you will need the Application Server administrator user ID and password.

Upgrading Release 3 Instant Messaging (Solaris)

This section discusses considerations that impact the upgrade procedure for Instant Messaging followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Instant Messaging software to Java ES Release 4 takes into account the following considerations:

- Instant Messaging components should be shut down when patches are being applied to the installed image.
- In architectures in which different Instant Messaging subcomponents reside on different computers, for example messenger resources on one computer, Instant Messaging server on another, and Instant Messaging Multiplexor on yet another, the upgrade must be performed on all such computers.
- The Instant Messaging upgrade applies to multiple subcomponents of Instant Messaging on one computer using the same installed image.
- The Release 4 Instant Messaging upgrade patch for Solaris OS are shown in the following table:

Table 16-4 Patches[1] to Upgrade Instant Messaging on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Instant Messaging	118786-08	118787-08
Instant Messaging localization	119707-06	119707-06

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 16-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   `su -`

3. Stop Instant Messaging if it is running.

   InstantMessaging-base`/sbin/imadmin stop`

4. If you have not already done so, upgrade the IM-SDK shared component to Release 4 and any other shared components you wish to upgrade.

   See "Upgrade Instant Messaging Dependencies" on page 296.

5. Apply the appropriate Instant Messaging patches in Table 16-4.

   Be sure to apply the Instant Messaging localization patch (`119707`) before applying the Instant Messaging base patch.

   `patchadd` *patch_ID*

6. Confirm that the upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Restart the Instant Messaging service that was stopped in Step 3.

   InstantMessaging-base`/sbin/imadmin start`

Upgrading Release 3 Instant Messaging (Linux)

This section discusses considerations that impact the upgrade procedure for Instant Messaging followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Instant Messaging software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 297), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Instant Messaging upgrade patch for Linux OS is shown in the following table:

Table 16-5 Patches[1] to Upgrade Instant Messaging on Linux

Description	Patch ID and RPM names
Instant Messaging	118788-11 • `sun-im-client-7.0-13.8.i386.rpm` • `sun-im-server-7.0-13.8.i386.rpm` • `sun-im-mux-7.0-13.8.i386.rpm` • `sun-im-olh-7.0-13.8.i386.rpm` • `sun-im-install-7.0-13.8.i386.rpm` • `sun-im-ident-7.0-13.8.i386.rpm` • `sun-im-apidoc-7.0-13.8.i386.rpm`
Instant Messaging localization	119708-06 • `sun-im-client-`*`Locale`*`-7.0-12.i386.rpm` • `sun-im-ident-`*`Locale`*`-7.0-12.i386.rpm` • `sun-im-install-`*`Locale`*`-7.0-12.i386.rpm` • `sun-im-olh-`*`Locale`*`-7.0-12.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patch using the patch number and RPM name from Table 16-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Instant Messaging if it is running.

   InstantMessaging-base`/sbin/imadmin stop`

4. If you have not already done so, upgrade the IM-SDK shared component to Release 4 and any other shared components you wish to upgrade.

   See "Upgrade Instant Messaging Dependencies" on page 296.

5. Apply the RPMs for Instant Messaging in Table 16-5.

   Be sure to apply the Instant Messaging localization patch (`119708`) before applying the Instant Messaging base patch.

   ```
   rpm -Fvh sun-im-module-Locale-7.0-2.8.i386.rpm
   rpm -Fvh sun-im-module-7.0-13.8.i386.rpm
   ```

6. Confirm that the upgrade was successful:

   ```
   rpm -qa | grep sun-im
   ```

   The new version numbers of the RPMs should be returned.

7. Restart the Instant Messaging service that was stopped in Step 3.

   InstantMessaging-base`/sbin/imadmin start`

Verifying the Upgrade

The upgrade of Instant Messaging to Release 4 is verified by confirming that the upgrade patches have been properly applied. The steps for this verification were included in "Upgrade Procedure (Solaris)" on page 298 and "Upgrade Procedure (Linux)" on page 299.

You can also check the status of the various Instant Messaging subcomponents using the following command:

InstantMessaging-base/sbin/imadmin status

Or you can check the log file located at *iim.instancevardir* / log,

where *instancevardir* is specified in the iim.conf file (for path, see Table 16-2 on page 293).

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 298 and "Upgrade Procedure (Linux)" on page 299.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Instant Messaging followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Instant Messaging is pretty much the reverse of the procedure for upgrading to Release 4.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   su -

2. Stop Instant Messaging if it is running.

   InstantMessaging-base/sbin/imadmin stop

3. Remove the patches in Table 16-4 on page 298.

   patchrm *patch_ID*

4. Restart the Instant Messaging service that was stopped in Step 2.

   InstantMessaging-base/sbin/imadmin start

Multiple Instance Upgrades

Multiple instance upgrades are not applicable to Release 4 Instant Messaging.

Upgrading Instant Messaging from Java ES Release 2

This section includes information about upgrading Instant Messaging from Java ES 2004Q2 (Release 2) to Java ES 2005Q4 (Release 4). The procedure for upgrading Release 2 Instant Messaging to Release 4 is quite different from that for upgrading from Release 3 Instant Messaging.

The section covers the following topics:

Introduction

When upgrading Java ES Release 2 Instant Messaging to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed using an upgrade utility that performs all operations needed to upgrade Instant Messaging software.
- **Upgrade Dependencies.** Upgrade of any Java ES component on a computer from Release 2 requires the upgrade of all other Java ES components hosted by the computer; selective upgrade of Java ES components from Release 2 to Release 4 is not supported. In particular, all Java ES shared components used by Instant Messaging, the web container, and Access Manager need to be upgraded to Release 4.
- **Backward Compatibility.** Release 4 Instant Messaging is backwardly compatible with the Release 2 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade to Release 2 is achieved by saving all Release 2 software and data and manually reverting back to the Release 2 version. There is no utility for rolling back the upgrade.
- **Platform Issues.** The general approach for upgrading Instant Messaging is the same on both Solaris and Linux operating systems. The upgrade process includes any platform-specific details.

Release 2 Instant Messaging Upgrade

This section describes how to perform an upgrade of Instant Messaging from Java ES Release 2 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Instant Messaging you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Instant Messaging using standard version checking utilities

Solaris:

```
pkginfo -l SUNWiimin
```

Linux:

```
rpm -qa | grep sun-im
```

See Table 16-3 on page 296 for output values.

Upgrade Instant Messaging Dependencies

Java ES Release 4 does not support the coexistence of Release 2 and Release 4 shared components on a single computer.

You are therefore required to upgrade all local Java ES Release 2 components on which Instant Messaging depends to Release 4. When you upgrade all Instant Messaging dependencies on a computer, they should be upgraded in the following order, all before you upgrade Instant Messaging.

1. **Shared Components.** All shared components upon which Instant Messaging depends must be upgraded to Release 4. If other Java ES product components coexist with Instant Messaging on a single computer, you have to upgrade all Java ES shared components residing on that computer. Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 51).
2. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in "Upgrading Web Server from Java ES Release 2" on page 147 and "Upgrading Application Server from Java ES Release 2" on page 188, respectively.
3. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 4 are provided in "Upgrading Access Manager from Java ES Release 2" on page 224.
4. **Directory Server.** Instant Messaging is rarely dependent on a local Directory Server, however, instructions for upgrading Directory Server to Release 4 are provided in "Upgrading Directory Server and Administration Server from Java ES Release 2" on page 122

Back Up Instant Messaging Data

The Instant Messaging upgrade from Release 2 to Release 4 modifies configuration data and customizations. Before upgrading Instant Messaging, it is a good idea to back up the Instant Messaging database and any existing resource and configuration files. For more information, see the *Sun Java System Instant Messaging 7 2005Q4 Administration Guide* (`http://docs.sun.com/doc/819-2503`).

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. If you are using Web Server as a web container, no configuration information is needed. However if you are using Application Server as a web container, you will need the Application Server administrator user ID and password.

Upgrading Release 2 Instant Messaging (Solaris)

This section discusses considerations that impact the upgrade procedure for Instant Messaging followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Instant Messaging software to Java ES Release 4 takes into account the following considerations:

- In architectures in which different Instant Messaging subcomponents reside on different computers, for example messenger resources on one computer, Instant Messaging server on another, and Instant Messaging Multiplexor on yet another, the upgrade must be performed on all such computers. However, the upgrade applies to multiple subcomponents of Instant Messaging on one computer using the same installed image.
- The upgrade of Release 2 Instant Messaging software to Java ES Release 4 makes use of an upgrade utility that performs the following operations:
  - Creates a temporary directory where it stores working files.
  - Gathers and temporarily stores previous package installation parameters for all packages installed on the system.
  - Shuts down the previous version of the Instant Messaging server.
  - Installs new packages and patches existing packages.
  - Installs any new shared component packages needed by Instant Messaging.
  - Saves the previous graphics contents from IIM_DOCROOT and restores them to the new resource files location.
  - Restarts all services.
  - Deletes the temporary directory and its contents.

Upgrade Procedure (Solaris)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. If you have not already done so, upgrade the IM-SDK shared component to Release 4 and any other shared components you wish to upgrade.

   See "Upgrade Instant Messaging Dependencies" on page 303.

3. Run the `upgrade` utility from the Instant Messaging tools directory of the Java ES distribution.

   ```
   cd /Solaris_arch/Product/instant_messaging/Tools
   ./upgrade
   ```

   The `upgrade` utility creates a log file that shows the progression of the upgrade process. The log file resides in the following location:

   `/var/sadm/install/logs/Instant_Messaging_Upgrade.`*timestamp*

   Where *timestamp* is in the format yyyymmddhhss.

4. (Optional) Change configuration as necessary to use new features introduced after Release 2. For configuration information, see *Sun Java System Instant Messaging Administration Guide*, `http://docs.sun.com/doc/819-0430`.

5. Re-customize messenger resources.

   If you had customized your messenger resources, you need to reapply those customizations to the following files:

   InstantMessaging-base`/html/`*Locale*`/imbrand.jar`
   InstantMessaging-base`/html/`*Locale*`/imb[ssl].html|jnlp`

   Consult the customized files that you saved under "Back Up Instant Messaging Data" on page 304.

6. Re-deploy messenger resources into the web container.

   InstantMessaging-base / lib / deployHtml -f *webcontainerDeployLocation*

   where *webcontainerDeployLocation* is the directory location where you want to deploy the messenger resources. The location generally depends on the web container being used, for example:

   Web Server
   WebServer-base/docs/im

   Application Server
   AppServer8Config-base / nodeagents / *hostName_domainName* / *instanceName* / docroot / iim

Upgrading Release 2 Instant Messaging (Linux)

This section discusses considerations that impact the upgrade procedure for Instant Messaging followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Instant Messaging software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 305).

Upgrade Procedure (Linux)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. If you have not already done so, upgrade the IM-SDK shared component to Release 4 and any other shared components you wish to upgrade.

   See "Upgrade Instant Messaging Dependencies" on page 303.

3. Run the upgrade utility from the Instant Messaging tools directory of the Java ES distribution.

   ```
   cd /Linux_x86/Product/instant_messaging/Tools
   ./upgrade
   ```

The `upgrade` utility creates a log file that shows the progression of the upgrade process. The log file resides in the following location:

`/var/sadm/install/logs/Instant_Messaging_Upgrade.`*timestamp*

Where *timestamp* is in the format yyyymmddhhss.

4. (Optional) Change configuration as necessary to use new features introduced after Release 2. For configuration information, see *Sun Java System Instant Messaging Administration Guide*, `http://docs.sun.com/doc/819-0430`.

5. Re-customize messenger resources.

   If you had customized your messenger resources, you need to reapply those customizations to the following files:

   InstantMessaging-base`/html/`*Locale*`/imbrand.jar`

   InstantMessaging-base`/html/`*Locale*`/imb[ssl].html|jnlp`

   Consult the customized files that you saved under "Back Up Instant Messaging Data" on page 304.

6. Re-deploy messenger resources into the web container.

   InstantMessaging-base / lib / deployHtml -f *webcontainerDeployLocation*

   where *webcontainerDeployLocation* is the directory location where you want to deploy the messenger resources. The location generally depends on the web container being used, for example:

   Web Server
   WebServer-base`/docs/im`

   Application Server
   AppServer8Config-base / nodeagents / *hostName_domainName* / *instanceName* / docroot / iim

Verifying the Upgrade

You can check the status of the various Instant Messaging subcomponents using the following command:

InstantMessaging-base`/sbin/imadmin status`

Or you can check the log file located at *iim.instancevardir* / log,

where *instancevardir* is specified in the `etc/`*InstantMessaging-base*`/config/iim.conf` file.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure (Solaris)” on page 306 and “Upgrade Procedure (Linux)” on page 307. However, if you wish to use the feature enhancements of Release 4 over those of Release 2, you need to re-configure Instant Messaging and redeploy to the web container.

Rolling Back the Upgrade

Rollback of the Release 4 upgrade to Release 2 is achieved by saving all Release 2 software and data (see “Back Up Instant Messaging Data” on page 304) and manually reverting back to the Release 2 version. There is no utility for rolling back the upgrade.

Multiple Instance Upgrades

Multiple instance upgrades are not applicable to Release 4 Instant Messaging.

Chapter 17

Portal Server

This chapter describes how to upgrade Portal Server to Java ES 2005Q4 (Release 4): Sun Java System Portal Server 6.3.1 2005Q4. The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

NOTE File locations in this chapter are specified with respect to two directory paths referred to as *PortalServer-base and PortalServerConfig-base*. At least part of these paths might have been specified as an installation directory when Portal Server was initially installed. If not, the Java ES installer assigned a default value.

The default value of *PortalServer-base* depends on operating system platform:

- Solaris: `/opt/SUNWps`
- Linux: `/opt/sun/portal`

The default value of *PortalServerConfig-base* depends on operating system platform:

- Solaris: `/etc/opt/SUNWps`
- Linux: `/etc/opt/sun/portal`

Overview of Portal Server Upgrades

This section describes the following general aspects of Portal Server that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Portal Server

Java ES Release 4 Portal Server is functionally the same as Release 3, but contains bug fixes made since Release 3.

Portal Server Upgrade Roadmap

Table 17-1 shows the supported Portal Server upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 17-1 Upgrade Paths to Java ES Release 4: Sun Java System Portal Server 6.3.1 2005Q4

Java ES Release	Portal Server Version	General Approach	Re-configuration Required
Release 3	Sun Java System Portal Server 6.3.1 2005Q1	Direct upgrade: Performed by applying patches. Some limitations apply (see procedures).	None
Release 2	Sun Java System Portal Server 6.3 2004Q2	Direct upgrade: Performed by applying patches to upgrade to Release 4, reconfiguring the software, and redeploying to the web container.	Configuration data

Table 17-1 Upgrade Paths to Java ES Release 4: Sun Java System Portal Server 6.3.1 2005Q4 *(Continued)*

Java ES Release	Portal Server Version	General Approach	Re-configuration Required
Release 1	Sun ONE Portal Server 6.2 (2003Q4)	No direct upgrade: But can be performed by upgrading first to Release 3 and then applying patches to upgrade to Release 4. Some limitations apply (see procedures).	Configuration data
Pre-dates Java ES releases		No direct upgrade.	

Portal Server Data

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

Table 17-2 Portal Server Data Usage

Type of Data	Location	Usage
Configuration data	*PortalServerConfig-base*/	Configuration of Portal Server.
Web container configuration	Web Server: `server.policy` and `server.xml` files in *WebServer-base*/`https-`*hostname*/`config`	Configuration of Portal Server web container instance.
	Application Server (Java ES Release 3 and 4): `server.policy` and `domain.xml` files in *AppServer8Config-base*/`domains/`*domainName*/`config`	
	Application Server (Java ES Release 2): `server.policy` and `server.xml` files in *AppServer7Config-base*/`domains/`*domainName*/`config`	
Customization data	*PortalServerConfig-base*/`desktop`	JAR files for customized modules
		Customized sample Portal Server desktop
Directory schema Services configuration User data	Directory Server	Portal Server depends on services configurations, such as the portal desktop, and user profile data that is stored in a directory.

Table 17-2 Portal Server Data Usage *(Continued)*

Type of Data	Location	Usage
Dynamic application data	None	Portal Server does not persistently store application data such as session state.

Compatibility Issues

Release 4 Portal Server does not introduce any interface changes. Portal Server components, including the mobile access component, are backwardly compatible with earlier versions.

Portal Server Dependencies

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

Portal Server has dependencies on the following Java ES components:

- **Shared components.** Portal Server has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Web Container.** Portal Server depends upon web container services, which can be provided either by Java ES Web Server or Java ES Application Server (or by third-party web containers from Weblogic and WebSphere).
- **Access Manager (or Access Manager SDK).** Portal Server depends upon Access Manager to provide authentication and authorization services for end users, including single sign-on. If Access Manager is run on a remote computer, then Access Manager SDK must be available locally.
- **Directory Server.** Portal Server accesses user data stored in Directory Server. As a result, Portal Server upgrades might require extensions of directory schema.

Upgrading Portal Server from Java ES Release 3

This section includes information about upgrading Portal Server from Java ES 2005Q1 (Release 3) to Java ES 2005Q4 (Release 4).

NOTE This section does not apply to the special case in which Portal Server is deployed in an Application Server web container and has been upgraded from Release 2 to Release 3 prior to being upgraded to Release 4. The aforementioned upgrade path is not currently supported.

The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Portal Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version and redeploying Portal Server to a web container.
- **Upgrade Dependencies.** While Portal Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Portal Server is compatible with the Release 3 version of these components. Upgrade of these shared components, except for Mobile Access Core (MA Core), is therefore optional with respect to upgrade of Portal Server to Release 4.

  In addition, Release 4 Portal Server is dependent upon a web container, Access Manager, and Directory Server as described in "Portal Server Dependencies" on page 314. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Portal Server to Release 4.
- **Backward Compatibility.** Release 4 Portal Server is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Portal Server to Release 3 is achieved by rolling back the patches applied during the upgrade. Patch rollback is not available on the Linux platform.

- **Platform Issues.** The general approach for upgrading Portal Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Portal Server Upgrade

This section describes how to perform an upgrade of Portal Server from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Portal Server using the following command:

PortalServer-base`/bin/version`

Table 17-3 Portal Server Version Verification Outputs

Java ES Release	Portal Server Version Number
Release 2	`6.3`
Release 3	`6.3.1`
Release 4	`6.3.1`[1]

1. The only difference between Release 3 and Release 4 is a patch. You can check for the Release 4 patches shown in Table 17-5 on page 321 and Table 17-7 on page 336 using the Solaris `showrev -p | grep` *patch_ID* command and the Linux `rpm -qa sun-portal-core` command and look for the string "`25.12`" or greater.

Upgrade Portal Server Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, Portal Server has a hard upgrade dependency only on the Mobile Access Core (MA Core) shared component. Upgrading of otherJava ES Release 3 components upon which Portal Server depends is therefore optional.

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

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, "Directory Server and Administration Server" on page 103.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 6, "Web Server" on page 137 and Chapter 9, "Application Server" on page 175, respectively.

NOTE Upgrading third-party web containers, such as those from WebLogic and WebSphere can cause Portal Server to break because customizations made to these containers to support Portal Server are overwritten by the container upgrade.

In these cases you have to reinstall and re-configure Portal Server for the upgraded web container environments.

4. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, “Access Manager” on page 203.

Back Up Release 3 Portal Server Configuration Information

Upgrade of Portal Server to Release 4 does not require the re-configuration of Portal Server software. However, as a safety measure you can back up the following directories where configuration information is stored:

PortalServerConfig-base /

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. If you are using Web Server as a web container, no configuration information is needed. However if you are using Application Server as a web container, you will need the Application Server administrator user ID and password.

Upgrading Release 3 Portal Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Portal Server followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Portal Server software to Java ES Release 4 takes into account the following considerations:

- All Portal Server instances corresponding to the same installed Portal Server image are upgraded at the same time. All such instances should be shut down by shutting down the web container when patches are being applied to the installed image.
- The Release 4 Portal Server upgrade patches for Solaris OS are shown in the following table:

Table 17-4 Patches[1] to Upgrade Portal Server on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Portal Server core	118950-12	118951-12
Portal Server localization	119425-08	119425-08
Portal Server localization configurator	118115-11	118115-11

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 17-4.

   Always use the latest patch revision available, unless directed to use a specific revision.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Portal Server by stopping its web container.

   Web Server:
   WebServer-base`/https-`*instanceName*`/stop`

   Application Server:
   AppServer8-base`/bin/asadmin stop-domain` *domainName*

4. If you have not already done so, upgrade the MA Core shared component and any others you wish to upgrade.

   See “Upgrade Portal Server Dependencies” on page 317.

5. Apply the appropriate Portal Server patch in Table 17-4.

   Be sure to apply the Portal Server core patch before applying the two Portal Server localization patches.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Restart Portal Server by restarting its web container.

   Web Server:
   WebServer-base`/https-`*instanceName*`/start`

   Application Server:
   AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
   `--password` *password domainName*

8. Re-deploy the Portal Server web application to your web container.

   PortalServer-base`/bin/deploy redeploy`

   The redeploy command redeploys content from *PortalServer-base*`/web-src` to `/var/`*PortalServer-base*`/https-`*hostName*`/`*deploy-dir*`/web-apps`. Any customizations to the Portal Server web application should therefore be first made to `/web-src` and then deployed to `/web-apps`. Any changes you might make under `/web-apps` should be replicated in `/web-src` *before* running the `deploy` command, or such changes will be overwritten.

9. Stop and restart the web container.

   While not required in all situations, restarting the web container ensures that Portal Server starts in a clean state.

Upgrading Release 3 Portal Server (Linux)

This section discusses considerations that impact the upgrade procedure for Portal Server followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Portal Server software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 318), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Portal Server upgrade patches for Linux OS are shown in the following table:

Table 17-5 Patches[1] to Upgrade Portal Server on Linux

Description	Patch ID and RPM names
Portal Server core	118952-12 • `sun-portal-core-6.3-25.12.i386.rpm` and a number of other RPMs for the Portal desktop and Portal Server mobile access.
Portal Server localization	119426-07 • `sun-portal-core-`*`Locale`*`-6.3-24.i386.rpm` and a large number of other RPMs for the Portal Server mobile access, configuration, identity, and other components.
Portal Server localization configurator	118116-08 • `sun-portal-l10n-configurator-6.3-24.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 17-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Portal Server by stopping its web container.

   Web Server:
   WebServer-base/https-*instanceName*/stop

   Application Server:
   AppServer8-base/bin/asadmin stop-domain *domainName*

4. If you have not already done so, upgrade the MA Core shared component and any others you wish to upgrade.

   See "Upgrade Portal Server Dependencies" on page 317.

5. Apply the RPMs for the Portal Server core patch in Table 17-5.

   ```
   cd /tmp
   ```

   where /tmp is the directory to which you downloaded the patch in Step 1.

   ```
   ./update
   ```

   The update script installs the RPMs and also ensures that the correct configurational changes occur as the result of the patch.

6. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal-core-6.3-25
   ```

   The upgrade revision numbers of the RPMs should be returned.

7. Apply the RPMs for the two Portal Server localization patches in Table 17-5.

   ```
   rpm -Fvh --replacefiles sun-portal-*-Locale-6.3-24.i386.rpm
   rpm -Fvh --replacefiles
        sun-portal-l10n-configurator-6.3-24.i386.rpm
   ```

8. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal-l10n-configurator-6.3-24
   ```

   The upgrade revision numbers of the RPMs should be returned.

9. Restart Portal Server by restarting its web container.

   Web Server:
   WebServer-base/https-*instanceName*/start

   Application Server:
   AppServer8-base/bin/asadmin start-domain --user *admin_ID*
   --password *password domainName*

10. Re-deploy the Portal Server web application to your web container.

    PortalServer-base/bin/deploy redeploy

    The redeploy command redeploys content from *PortalServer-base*/web-src to /var/*PortalServer-base*/https-*hostName*/*deploy-dir*/web-apps. Any customizations to the Portal Server web application should therefore be first made to /web-src and then deployed to /web-apps. Any changes you might make under /web-apps should be replicated in /web-src *before* running the deploy command, or such changes will be overwritten.

11. Stop and restart the web container.

    While not required in all situations, restarting the web container ensures that Portal Server starts in a clean state.

Verifying the Upgrade

The upgrade of Portal Server to Release 4 is verified by confirming that the upgrade patches have been properly applied. The steps for this verification were included in "Upgrade Procedure (Solaris)" on page 319 and "Upgrade Procedure (Linux)" on page 321.

In addition, you can use the following command:

PortalServer-base/bin/version

See Table 17-3 on page 317 for output values.

Beyond these tests of the patch upgrade you can verify that what previously worked still works and that bug fixes of interest have actually been fixed.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 319 and "Upgrade Procedure (Linux)" on page 321.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Portal Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Portal Server is pretty much the reverse of the procedure for upgrading to Release 4. The re-configurations are rolled back and the patches are removed.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop Portal Server by stopping its web container.

   Web Server*:*
   WebServer-base/https-*instanceName*/stop

   Application Server*:*
   AppServer8-base/bin/asadmin start-domain --user *admin_ID*
   --password *password domainName*

3. Remove the patches in Table 17-4 on page 319.

   ```
   patchrm patch_ID
   ```

4. Restart Portal Server by restarting its web container.

   Web Server*:*
   WebServer-base/https-*instanceName*/start

   Application Server*:*
   AppServer8-base/bin/asadmin start-domain *domainName*
   --user *admin_ID* --password *password*

5. Re-deploy the Portal Server web application to your web container.

   PortalServer-base/bin/deploy redeploy

   The redeploy command redeploys content from *PortalServer-base*/web-src to /var/*PortalServer-base*/https-*hostName*/*deploy-dir*/web-apps. Any customizations to the Portal Server web application should therefore be first made to /web-src and then deployed to /web-apps. Any changes you might make under /web-apps should be replicated in /web-src *before* running the deploy command, or such changes will be overwritten.

6. Stop and restart the web container.

   While not required in all situations, restarting the web container ensures that Portal Server starts in a clean state.

Multiple Instance Upgrades

In some deployment architectures Portal Server is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Portal Server components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Portal Server, you can perform a rolling upgrade in which you upgrade the Portal Server instances sequentially without interrupting service. You upgrade each instance of Portal Server while the others remain running. You perform the upgrade of each instance as described in "Release 3 Portal Server Upgrade" on page 316.

Upgrading Portal Server from Java ES Release 2

This section includes information about upgrading Portal Server from Java ES 2004Q2 (Release 2) to Java ES 2005Q4 (Release 4).

Because of the complexity involved in a Release 2 Portal Server upgrade, and the likelihood of considerable down time, you might choose to perform a parallel upgrade on another computer rather than an in-place upgrade on a production system. Such an approach is advisable for business critical or complex Portal Server solutions where only limited downtime is acceptable. The duration of the upgrade procedure will also depend on the time required for you to re-implement and test any required Portal Server customizations.

It might also be necessary to modify or adapt the instructions in this section to accommodate your particular upgrade scenario. In such cases it would be advisable to contact Sun Microsystems support services for assistance in performing the upgrade.

This section covers the following topics regarding the upgrade from Release 2 to Release 4:

Introduction

When upgrading Java ES Release 2 Portal Server to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 2 version. Re-configuration of Portal Server is also required and performed using an upgrade utility.
- **Upgrade Dependencies.** Portal Server has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), and all these need to be Upgraded to Release 4 because Java ES does not support a mixture of Release 2 and Release 4 components on a single computer.

  In addition, Release 4 Portal Server is dependent upon a web container, Access Manager, and Directory Server as described in "Portal Server Dependencies" on page 314. Portal Server has a hard upgrade dependency on the web container and Access Manager (or Access Manager SDK) because they reside locally, and has a soft upgrade dependency on Directory Server, since it rarely resides locally.
- **Backward Compatibility.** Release 4 Portal Server is backwardly compatible with the Release 2 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Portal Server to Release 2 can not be achieved once Portal Server re-configuration has been performed.
- **Platform Issues.** The general approach for upgrading Portal Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 2 Portal Server Upgrade

This section describes how to perform an upgrade of Portal Server from Java ES Release 2 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Portal Server using the following command:

PortalServer-base`/bin/version`

See Table 17-3 on page 317 for output values.

Upgrade Portal Server Dependencies

Java ES Release 4 does not support the coexistence of Release 2 and Release 4 shared components on a single computer.

You are therefore required to upgrade all local Java ES Release 2 components on which Portal Server depends to Release 4. When you upgrade all local Portal Server dependencies on a computer, they should be upgraded in the following order, all before you upgrade Portal Server. Note that there are special requirements for specific upgrade scenarios.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Portal Server is rarely dependent on a local Directory Server. However, instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, "Directory Server and Administration Server" on page 103.

3. **Web Container Software.** Portal Server can be running in a web container provided by either Web Server or Application Server.

   - Web Server: Upgrade to Release 4 Web Server using the procedure in "Upgrading Web Server from Java ES Release 2" on page 147.
   - Application Server: Upgrade to Release 4 Application Server by performing a fresh install of Application Server using the Java ES installer rather than by using the procedure in "Upgrading Application Server from Java ES Release 2" on page 188. Be sure to obtain the admin port and server instance port from the Release 2 Application Server 7 before installing Release 4 Application Server 8.

NOTE Upgrading third-party web containers, such as those from WebLogic and WebSphere can cause Portal Server to break because customizations made to these containers to support Portal Server are overwritten by the container upgrade.

In these cases you have to reinstall and re-configure Portal Server for the upgraded web container environments.

4. **Access Manager (Access Manager SDK).** Portal Server can be running in the same web container as Access Manager or in a different web container.

   - If Portal Server is running in a different web container from Access Manager, for example if Access Manager is running remotely, then upgrade Access Manager, or Access Manager SDK, from Release 2 to Release 4 using the procedure in "Upgrading Access Manager from Java ES Release 2" on page 224. If you are only upgrading Access Manager SDK, refer to "Upgrading Release 3 Access Manager SDK" on page 223 and set `DEPLOY_LEVEL = 3`.
   - If Portal Server is running in the same web container as Access Manager, and that web container is provided by Web Server, then upgrade Access Manager from Release 2 to Release 4 using the procedure in "Upgrading Release 2 Access Manager: Web Server Web Container" on page 225.
   - If Portal Server is running in the same web container as Access Manager, and that web container is provided by Application Server, then upgrade Access Manager from Release 2 to Release 4 using the procedure in "Upgrading Release 2 Access Manager: Application Server Web Container" on page 225, but be sure to use the scenario in which AS has been *freshly installed*.

Back Up Release 2 Portal Server Configuration Information

Upgrade of Portal Server to Release 4 does requires the re-configuration of Portal Server software. As a safety measure you can back up the following directories where configuration information is stored:

PortalServerConfig-base/

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. If you are using Web Server as a web container, no administrator password is required. However if you are using Application Server as a web container, you will need the Application Server administrator user ID and password.

Upgrading Release 2 Portal Server (Solaris)

This section discusses considerations that impact the upgrade procedure for Portal Server followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Portal Server software to Java ES Release 4 takes into account the following considerations:

- All Portal Server instances corresponding to the same installed Portal Server image are upgraded at the same time. All such instances should be shut down by shutting down the web container when patches are being applied to the installed image.
- The Release 4 Portal Server upgrade patches for Solaris OS are shown in the following table:

Table 17-6 Patches to Upgrade Portal Server to Release 4 on Solaris

Description	SPARC Solaris 8 & 9	X86 Solaris 9	Solaris 10
Portal Server sync-up	118195-07	118196-07	Doesn't apply
Portal Server core	118128-13	118129-13	Doesn't apply
Mobile Access core	119527-02	119528-02	Doesn't apply
Portal Server fixes	118950-15 (or higher)	118951-15 (or higher)	Doesn't apply

- The procedure for upgrading Portal Server on Solaris platforms depends upon whether Portal Server is deployed in a web container provided by Web Server or by Application Server. Hence there is a separate set of upgrade instructions below for each of these two web containers.

Upgrade Procedure (Solaris: Web Server)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 17-6.

   Be sure to download the exact patch revisions shown in Table 17-6, except for the Portal Server fixes, for which a later patch might be available.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   `su -`

3. Stop Portal Server by stopping its web container.

   WebServer-base`/https-`*instanceName*`/stop`

4. If you have not already done so, upgrade all shared components, the web container, and Access Manager (or Access Manager SDK).

   See "Upgrade Portal Server Dependencies" on page 327.

5. If not yet running, start Directory Server and Access Manager.

6. Apply the appropriate Portal Server patches in Table 17-6.

   Be sure to apply the patches in the order in which they are shown in Table 17-6, from top to bottom.

   `patchadd` *patch_ID*

7. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 7.

8. Re-configure Portal Server software:

   `ksh`

   ```
   $ cd PortalServer-base/lib
   $ ./upgradePS04Q205Q1
   ```

9. Restart Portal Server by restarting its web container.

   WebServer-base/https-*instanceName*/start

10. Re-deploy the Portal Server web application to your web container.

    PortalServer-base/bin/deploy redeploy

    The redeploy command redeploys content from *PortalServer-base*/web-src to /var/*PortalServer-base*/https-*hostName*/*deploy-dir*/web-apps. Any customizations to the Portal Server web application should therefore be first made to /web-src and then deployed to /web-apps. Any changes you might make under /web-apps should be replicated in /web-src *before* running the deploy command, or such changes will be overwritten.

11. Stop and restart the web container.

    While not required in all situations, restarting the web container ensures that Portal Server starts in a clean state.

Upgrade Procedure (Solaris: Application Server)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 17-6.

   Be sure to download the exact patch revisions shown in Table 17-6, except for the Portal Server fixes, for which a later patch might be available.

   Patches can be downloaded to /tmp from:
   http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

2. Log in as root or become superuser.

   su -

3. Make sure that Portal Server is no longer running in its Release 2 Application Server instance.

   AppServerConfig7-base/domains/*domainName*/*instanceName*/bin/stopserv

   In the above command, and in subsequent steps, the following conventions are used:

   - The default *domainName* is domain1
   - The default *instanceName* is server1

4. If you have not already done so, upgrade all shared components, the web container, and Access Manager (or Access Manager SDK).

   See "Upgrade Portal Server Dependencies" on page 327.

5. Make sure that the upgraded Access Manager is not running in its Release 4 Application Server instance.

   AppServer8-base/bin/asadmin stop-domain *domainName*

6. Make sure the Access Manager configuration file,

   AccessManagerConfig-base/config/AMConfig.properties

   contains the following property values:

   ```
   com.iplanet.am.notification.url=
       http://hostName:port/amserver/notificationservice
   com.sun.identity.webcontainer=IAS8.1
   com.iplanet.am.cookie.encode=true
   ```

   where *hostName*:*port* is the computer and port hosting the Access Manager instance.

7. Apply the appropriate Portal Server patches in Table 17-6.

   Be sure to apply the patches in the order in which they are shown in Table 17-6, from top to bottom.

   patchadd *patch_ID*

8. Confirm that the patch upgrade was successful:

   showrev -p | grep *patch_ID*

   The output should return the versions of patch IDs applied in Step 7.

9. Make sure the Portal Server configuration file,

   PortalServerConfig-base/PSConfig.properties

   contains the following property values, which reference the Application Server's Domain Administration Server (DAS) instance:

   ```
   DEPLOY_TYPE=SUNONE8
   DEPLOY_INSTANCE_DIR=AppServer8Config-base/domains/domainName
   DEPLOY_DOMAIN=AppServer8Config-base/domains/domainName
   DEPLOY_PRODUCT_DIR=AppServer8Config-base/domains/domainName
   DEPLOY_ADMIN_PROTOCOL=https
   DEPLOY_ADMIN_PORT=DAS_adminPort  (for example, default=4848)
   DEPLOY_ADMIN_HOST=DAS_hostName
   ```

LOAD_BALANCER_URL=http://*DAS_hostNam*e:*DAS_hostPort/*portal
DEPLOY_DOCROOT=*AppServer8Config-base*/domains/*domainName*/docroot
PS_PORT=*DAS_hostPort* (for example, default=80)
DEPLOY_DIR=*AppServer8-base*
PS_PROTOCOL=http

Assuming that the port values assigned to the fresh Release 4 Application Server 8 installation were the same as those of the Release 2 Application Server 7 installation, and that those were the default port values, then the default values shown above would apply.

10. Modify the PSconfig.properties file as follows:

DEPLOY_INSTANCE=*temporary_instanceName*

where *temporary_instanceName* is an unused temporary value.

11. Make sure that the DAS is running.

AppServer8-base/bin/asadmin start-domain --user *admin_ID*
--password *password domainName*

12. Perform the following commands:

cd *PortalServer-base*/bin
./multiserverinstance

A number of the questions asked by the multiserverinstance script use the values set in the PSConfig.properties file shown in Step 9 on page 332 as defaults, and the following instructions assume these defaults are correct.

Respond to the questions asked by the multiserverinstance script as follows:

1. Select option 1 for Create a new portalserver instance.
2. Select option 3 for Sun Java System Application Server 8.1.
3. Where is the Web Container installed? Hit return.
4. What is the domain name? Hit return.
5. What is the domain (DAS) path? Enter the same value that was shown as the default for question #4.
6. What is the Web Container instance path? Enter the same value as was entered for #5.
7. What is the Web Container administrator? Hit return.
8. What is the Web Container administration port? Hit return.
9. Is the Web Container administration port secure? Hit return.
10. Instance name? Enter a value of server.
11. Instance port? Enter the same value as was entered for the PS_PORT value in the PSConfig.properties file.

12. Is the instance port secure? Hit `return`.
13. What is the Web Container document root directory? Hit `return`.
14. What is the Application Server administration password? Enter password.
15. What is the Identity Server administration password? Enter password.

13. Modify the `PSconfig.properties` file as follows:

```
DEPLOY_INSTANCE=server
```

where the value `server` is the default instance name of the DAS instance.

14. Restart the DAS.

AppServer8-base`/bin/asadmin stop-domain` *domainName*

AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
`--password` *password domainName*

15. Deploy the Portal Server web application.

```
cd PortalServer-base/bin
./deploy redeploy
```

Ignore messages indicating there might be errors in `deploy.log`.

The redeploy command redeploys content from *PortalServer-base*`/web-src` to `/var/`*PortalServer-base*`/https-`*hostName*`/`*deploy-dir*`/web-apps`. Any customizations to the Portal Server web application should therefore be first made to `/web-src` and then deployed to `/web-apps`. Any changes you might make under `/web-apps` should be replicated in `/web-src` *before* running the `deploy` command, or such changes will be overwritten.

16. Re-configure Portal Server software:

```
ksh

$ cd PortalServer-base/lib
$ ./postinstall_PortletSamples
$ ./upgradePS04Q205Q1
```

Ignore `CLI137`-related errors and (un)deploy-related errors issued by the `upgradePS04Q205Q1` script.

17. Restart the DAS.

AppServer8-base/bin/asadmin stop-domain *domainName*

AppServer8-base/bin/asadmin start-domain --user *admin_ID* --password *password domainName*

While not required in all situations, restarting the web container ensures that Portal Server starts in a clean state.

18. Update the Portal Server Display Profile

a. Execute the following command:

PortalServer-base/bin/dpadmin list -g -u *amadminDN* -w *amadminPassword* /tmp/GlobalDP.xml

Where the value for *amadminDN* can be found in the property com.sun.identity.authentication.super.user in the *AccessManagerConfig-base*/config/AMConfig.properties file.

b. Open the file /tmp/GlobalDP.xml for editing

c. Modify the value of:

org.apache.xalan.xsltc.trax.TransformerFactoryImpl
to
com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl

d. Modify all occurrences of the value:

Sun JavaTM System Portal Server 6 2004Q2
to
Sun JavaTM System Portal Server 6 2005Q4

e. Execute the following command:

PortalServer-base/bin/dpadmin modify -g -u *amadminDN* -w *amadminPassword* /tmp/GlobalDP.xml

Where the value for *amadminDN* is the same as in Step a.

Upgrading Release 2 Portal Server (Linux)

This section discusses considerations that impact the upgrade procedure for Portal Server followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Portal Server software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 318), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Portal Server upgrade patches for Linux OS are shown in the following table:

Table 17-7 Patches to Upgrade Portal Server to Release 4 on Linux

Description	Patch ID and RPM names
Portal Server core	118020-16 `sun-portal-`*module*`-6.3-25.i386.rpm` where *module* is any of about 70 different software modules
Mobile Access core	119529-02 • `sun-mobileaccess-1.0-25.2.i386.rpm` • `sun-mobileaccess-config-1.0-25.2.i386.rpm`
Portal Server fixes	118952-15 (or higher) • `sun-portal-core-6.3-`*xx.y*`.i386.rpm` • `sun-portal-configurator-6.3-`*xx.y*`.i386.rpm` • `sun-portal-mobileaccess-6.3-`*xx.y*`.i386.rpm` • `sun-portal-desktop-6.3-`*xx.y*`.i386.rpm` • `sun-portal-sample-6.3-`*xx.y*`.i386.rpm` • `sun-portal-mobileaccess-config-6.3-`*xx.y*`.i386.rpm`

Upgrade Procedure (Linux: Web Server)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

1. Obtain the required patches using the patch numbers and RPM names from Table 17-7. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Portal Server by stopping its web container.

   WebServer-base`/https-`*instanceName*`/stop`

4. If you have not already done so, upgrade all shared components, the web container, and Access Manager (or Access Manager SDK).

   See "Upgrade Portal Server Dependencies" on page 327.

5. If not yet running, start Directory Server and Access Manager.

6. Apply the RPMs for Portal Server in Table 17-7.

   a. `cd /tmp`

      where `/tmp` is the directory to which you downloaded the patch in Step 1.

   b. Unzip the 118020 patch file, read the `README` file and run the following script:

      ```
      ./upgradeportalrpms
      ```

      The `upgradeportalrpms` script installs the RPMs and also ensures that the correct configurational changes occur as the result of the patch.

   c. Unzip the 119529 patch file, and run the `./update` script from within the directory that was created when the patch was unzipped.

   d. Unzip the 118952 patch file, and run the `./update` script from within the directory that was created when the patch was unzipped.

7. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal
   rpm -qa | grep sun-mobileaccess
   ```

   The new version numbers of the RPMs should be returned.

8. Re-configure Portal Server software:

   ksh

   ```
   $ cd PortalServer-base/lib
   $ ./upgradePS04Q205Q1
   ```

9. Edit the *PortalServer-base*/export/deploy.import file as follows:

   If the following is present:

   ```
   %JATO_LIB_DIR%/jato.tld %WEB_SRC_DIR%/WEB-INF/jato.tld
   %JATO_LIB_DIR%/jato.jar %WEB_SRC_DIR%/WEB-INF/lib/jato.jar
   ```

   Replace with:

   ```
   /usr/share/lib/jato/jato.tld %WEB_SRC_DIR%/WEB-INF/jato.tld
   /usr/share/lib/jato/jato.jar %WEB_SRC_DIR%/WEB-INF/lib/jato.jar
   ```

   In other words, replace %JATO_LIB_DIR% with /usr/share/lib/jato

10. Restart Portal Server by restarting its web container.

    WebServer-base/https-*instanceName*/start

11. Re-deploy the Portal Server web application to your web container.

    PortalServer-base/bin/deploy redeploy

    The redeploy command redeploys content from *PortalServer-base*/web-src to /var/*PortalServer-base*/https-*hostName*/*deploy-dir*/web-apps. Any customizations to the Portal Server web application should therefore be first made to /web-src and then deployed to /web-apps. Any changes you might make under /web-apps should be replicated in /web-src *before* running the deploy command, or such changes will be overwritten.

12. Stop and restart the web container.

    While not required in all situations, restarting the web container ensures that Portal Server starts in a clean state.

Upgrade Procedure (Linux: Application Server)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 17-7.

   Be sure to download the exact patch revisions shown in Table 17-7, except for the Portal Server fixes, for which a later patch might be available.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Make sure that Portal Server is no longer running in its Release 2 Application Server instance.

   AppServerConfig7-base`/domains/`*domainName*`/`*instanceName*`/bin/stopserv`

   In the above command, and in subsequent steps, the following conventions are used:

   - The default *domainName* is `domain1`
   - The default *instanceName* is `server1`

4. If you have not already done so, upgrade all shared components, the web container, and Access Manager (or Access Manager SDK).

   See "Upgrade Portal Server Dependencies" on page 327.

5. Make sure that the upgraded Access Manager is not running in its Release 4 Application Server instance.

   AppServer8-base`/bin/asadmin stop-domain` *domainName*

6. Make sure the Access Manager configuration file,

   AccessManagerConfig-base`/config/AMConfig.properties`

   contains the following property values:

   ```
   com.iplanet.am.notification.url=
       http://hostName:port/amserver/notificationservice
   com.sun.identity.webcontainer=IAS8.1
   com.iplanet.am.cookie.encode=true
   ```

   where *hostName*:*port* is the computer and port hosting the Access Manager instance.

7. Apply the RPMs for Portal Server in Table 17-7.

   a. `cd /tmp`

      where `/tmp` is the directory to which you downloaded the patch in Step 1.

   b. Unzip the 118020 patch file, read the `README` file and run the following script:

      `./upgradeportalrpms`

      The `upgradeportalrpms` script installs the RPMs and also ensures that the correct configurational changes occur as the result of the patch.

   c. Unzip the 119529 patch file, and run the `./update` script from within the directory that was created when the patch was unzipped.

   d. Unzip the 118952 patch file, and run the `./update` script from within the directory that was created when the patch was unzipped.

8. Confirm that the patch upgrade was successful:

   rpm -qa | grep sun-portal
   rpm -qa | grep sun-mobileaccess

   The new version numbers of the RPMs should be returned.

9. Edit the *PortalServer-base*`/export/deploy.import` file as follows:

   If the following is present:

   ```
   %JATO_LIB_DIR%/jato.tld %WEB_SRC_DIR%/WEB-INF/jato.tld
   %JATO_LIB_DIR%/jato.jar %WEB_SRC_DIR%/WEB-INF/lib/jato.jar
   ```

   Replace with:

   ```
   /usr/share/lib/jato/jato.tld %WEB_SRC_DIR%/WEB-INF/jato.tld
   /usr/share/lib/jato/jato.jar %WEB_SRC_DIR%/WEB-INF/lib/jato.jar
   ```

   In other words, replace `%JATO_LIB_DIR%` with `/usr/share/lib/jato`

10. Follow Step 9 on page 332 through Step 18 on page 335 under "Upgrade Procedure (Solaris: Application Server)."

Verifying the Upgrade

The upgrade of Portal Server to Release 4 is verified by confirming that the upgrade patches have been properly applied. The steps for this verification were included in "Upgrade Procedure (Solaris)" on page 319 and "Upgrade Procedure (Linux)" on page 321.

In addition, you can use the following command:

PortalServer-base/bin/version

See Table 17-3 on page 317 for output values.

Beyond these tests of the patch upgrade you can verify that what previously worked still works and that bug fixes of interest have actually been fixed.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure (Solaris: Application Server)” on page 331 and “Upgrade Procedure (Linux: Web Server)” on page 337.

Rolling Back the Upgrade

The upgrade of Portal Server from Release 2 to Release 4 cannot be rolled back.

Multiple Instance Upgrades

In some deployment architectures Portal Server is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Portal Server components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Portal Server, you can perform a rolling upgrade in which you upgrade the Portal Server instances sequentially without interrupting service. You upgrade each instance of Portal Server while the others remain running. You perform the upgrade of each instance as described in “Release 2 Portal Server Upgrade” on page 327.

Chapter 18

Portal Server Secure Remote Access

This chapter describes how to upgrade Portal Server Secure Remote Access to Java ES 2005Q4 (Release 4): Sun Java System Portal Server Secure Remote Access 6 2005Q4.

The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

NOTE File locations in this chapter are specified with respect to two directory paths referred to as *PortalServer-base and PortalServerConfig-base*. At least part of these paths might have been specified as an installation directory when Portal Server was initially installed. If not, the Java ES installer assigned a default value.

The default value of *PortalServer-base* depends on operating system platform:

- Solaris: `/opt/SUNWps`
- Linux: `/opt/sun/portal`

The default value of *PortalServerConfig-base* depends on operating system platform:

- Solaris: `/etc/opt/SUNWps`
- Linux: `/etc/opt/sun/portal`

Overview of Portal Server Secure Remote Access Upgrades

This section describes the following general aspects of Portal Server Secure Remote Access that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Portal Server Secure Remote Access

Java ES Release 4 Portal Server Secure Remote Access mostly represents bug fixes. There is no major new functionality with respect to Release 3.

Portal Server Secure Remote Access Upgrade Roadmap

Table 18-1 shows the supported Portal Server Secure Remote Access upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 18-1 Upgrade Paths to Java ES Release 4: Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4

Java ES Release	Portal Server Secure Remote Access Version	General Approach	Re-configuration Required
Release 3	Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q1	Direct upgrade: Performed by applying patches.	None
Release 2	Sun Java System Portal Server Secure Remote Access 6.3 2004Q2	No direct upgrade: Performed by upgrading first to Release 3 and then applying patches to upgrade to Release 4.	Configuration data

Table 18-1 Upgrade Paths to Java ES Release 4: Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4 *(Continued)*

Java ES Release	Portal Server Secure Remote Access Version	General Approach	Re-configuration Required
Release 1	Sun ONE Portal Server Secure Remote Access 6.2 (2003Q4)	No direct upgrade: Performed by upgrading first to Release 3 and then applying patches to upgrade to Release 4.	Configuration data
Pre-dates Java ES releases		No direct upgrade.	

Portal Server Secure Remote Access Data

The following table shows the type of data that could be impacted by an upgrade of Portal Server Secure Remote Access software.

Table 18-2 Portal Server Secure Remote Access Data Usage

Type of Data	Location	Usage
Configuration data	*PortalServerConfig-base/*	Configuration of Portal Server Secure Remote Access
Directory schema User data	Directory Server	Portal Server Secure Remote Access depends on user profile data that is stored in a directory.
Dynamic application data	None	Portal Server Secure Remote Access does not persistently store application data such as session state.

Compatibility Issues

Release 4 Portal Server Secure Remote Access does not introduce any interface changes. Portal Server Secure Remote Access is backwardly compatible with earlier versions.

Portal Server Secure Remote Access Dependencies

Portal Server Secure Remote Access dependencies on other Java ES components can impact the procedure for upgrading and re-configuring Portal Server Secure Remote Access software. Changes in Portal Server Secure Remote Access interfaces or functions, for example, could require upgraded version of components upon which Portal Server Secure Remote Access depends. The need to upgrade such components depends upon the specific upgrade path.

Portal Server Secure Remote Access has dependencies on the following Java ES components:

- **Shared components.** Portal Server Secure Remote Access has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Portal Server.** Portal Server Secure Remote Access provides secure remote access to Portal Server.
- **Access Manager (or Access Manager SDK).** Portal Server Secure Remote Access depends upon Access Manager to provide authentication and authorization services for end users, including single sign-on. If Access Manager is run on a remote computer, then Access Manager SDK must be available locally.
- **Directory Server.** Portal Server Secure Remote Access accesses user data stored in Directory Server. As a result, Portal Server Secure Remote Access upgrades might require extensions of directory schema.

Upgrading Portal Server Secure Remote Access from Java ES Release 3

This section includes information about upgrading Portal Server Secure Remote Access from Java ES 2005Q1 (Release 3) to Java ES Release 4. The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Portal Server Secure Remote Access to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. Re-configuration of Portal Server Secure Remote Access is not required.
- **Upgrade Dependencies.** While Portal Server Secure Remote Access has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Portal Server Secure Remote Access is compatible with the Release 3 version of these components. Upgrade of these shared components, except for Mobile Access Core (MA Core), is therefore optional with respect to upgrade of Portal Server Secure Remote Access to Release 4.

  In addition, Release 4 Portal Server Secure Remote Access is dependent upon Portal Server, Access Manager, and Directory Server as described in “Portal Server Secure Remote Access Dependencies” on page 346. Upgrade of Portal Server Secure Remote Access to Release 4 requires that Portal Server also be upgraded. However, the dependencies on Access Manager and Directory Server are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Portal Server Secure Remote Access to Release 4.
- **Backward Compatibility.** Release 4 Portal Server Secure Remote Access is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Portal Server Secure Remote Access to Release 3 is achieved by rolling back the patches applied during the upgrade. Patch rollback is not available on the Linux platform.

- **Platform Issues.** The general approach for upgrading Portal Server Secure Remote Access is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Portal Server Secure Remote Access Upgrade

This section describes how to perform an upgrade of Portal Server Secure Remote Access from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Portal Server Secure Remote Access you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Portal Server Secure Remote Access using the following commands:

PortalServer-base`/bin/gateway version`
PortalServer-base`/bin/rwproxyd version`
PortalServer-base`/bin/netletd version`

Table 18-3 Portal Server Secure Remote Access Version Verification Outputs

Java ES Release	Portal Server Version Number
Release 2	Earlier than Release 3
Release 3	`Thu Dec 16 03:30:34 PST 2004`
Release 4	Later than Release 3[1]

1. The only difference between Release 3 and Release 4 is a patch. You can check for the Release 4 patches shown in Table 18-5 on page 352 and Table 18-7 on page 362 using the Solaris `showrev -p | grep` *patch_ID* command and the Linux `rpm -qa sun-portal-core` command and look for the string "`25.12`" or greater.

Upgrade Portal Server Secure Remote Access Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, Portal Server has a hard upgrade dependency only on the Mobile Access Core (MA Core) shared component. Upgrading of otherJava ES Release 3 components upon which Portal Server depends is therefore optional.

However, if you choose to upgrade all Portal Server Secure Remote Access dependencies, they should be upgraded in the following order, all before you upgrade Portal Server Secure Remote Access. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, "Directory Server and Administration Server" on page 103.
3. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, "Access Manager" on page 203.
4. **Portal Server.** Instructions for upgrading Portal Server are provided in Chapter 17, "Portal Server" on page 311.

Back Up Release 3 Portal Server Secure Remote Access Configuration Information

Upgrade of Portal Server Secure Remote Access to Release 4 does not require the re-configuration of Portal Server Secure Remote Access software. However, as a safety measure you can back up the following directories where configuration information is stored:

PortalServerConfig-base/

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade.

Upgrading Release 3 Portal Server Secure Remote Access (Solaris)

This section discusses considerations that impact the upgrade procedure for Portal Server Secure Remote Access followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Portal Server Secure Remote Access software to Java ES Release 4 takes into account the following considerations:

- All Portal Server Secure Remote Access instances corresponding to the same installed Portal Server Secure Remote Access image are upgraded at the same time. All such instances should be shut down when patches are being applied to the installed image.
- The Release 4 Portal Server Secure Remote Access upgrade patches for Solaris OS are the same as those used to upgrade Portal Server and are shown in the following table:

Table 18-4 Patches[1] to Upgrade Portal Server Secure Remote Access on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Portal Server core	118950-12	118951-12
Portal Server localization	119425-08	119425-08
Portal Server localization configurator	118115-11	118115-11

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Portal Server Secure Remote Access on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 18-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway stop
   /etc/init.d/netletd stop
   /etc/init.d/rwproxyd stop
   ```

4. If you have not already done so, upgrade the MA Core shared component and any others you wish to upgrade.

   See "Upgrade Portal Server Secure Remote Access Dependencies" on page 349.

5. Apply the appropriate Portal Server patches in Table 18-4.

   Be sure to apply the Portal Server core patch before applying the two Portal Server localization patches.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Restart Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway start
   /etc/init.d/netletd start
   /etc/init.d/rwproxyd start
   ```

Upgrading Release 3 Portal Server Secure Remote Access (Linux)

This section discusses considerations that impact the upgrade procedure for Portal Server Secure Remote Access followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Portal Server Secure Remote Access software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 350), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Portal Server Secure Remote Access upgrade patches for Linux OS are the same as those used to upgrade Portal Server and are shown in the following table:

Table 18-5 Patches[1] to Upgrade Portal Server Secure Remote Access on Linux

Description	Patch ID and RPM names
Portal Server core	118952-12 • `sun-portal-core-6.3-25.12.i386.rpm` and a number of other RPMs for the Portal desktop and Portal Server mobile access.
Portal Server localization	119426-07 • `sun-portal-core-`*Locale*`-6.3-24.i386.rpm` and a large number of other RPMs for the Portal Server mobile access, configuration, identity, and other components.
Portal Server localization configurator	118116-08 • `sun-portal-l10n-configurator-6.3-24.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to Portal Server Secure Remote Access on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 18-4. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway stop
   /etc/init.d/netletd stop
   /etc/init.d/rwproxyd stop
   ```

4. If you have not already done so, upgrade the MA Core shared component and any others you wish to upgrade.

   See "Upgrade Portal Server Secure Remote Access Dependencies" on page 349.

5. Apply the RPMs for Portal Server core patch in Table 18-5.

   ```
   cd /tmp
   ```

   ```
   ./update
   ```

   The update script installs the RPMs and also ensures that the correct configurational changes occur as the result of the patch.

6. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal-gateway
   ```

   The new version numbers of the RPMs should be returned.

7. Apply the RPMs for the two Portal Server localization patches in Table 18-5.

   ```
   rpm -Fvh --replacefiles sun-portal-*-Locale-6.3-24.i386.rpm
   rpm -Fvh --replacefiles
        sun-portal-l10n-configurator-6.3-24.i386.rpm
   ```

8. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal-l10n-configurator-6.3-24
   ```

   The upgrade revision numbers of the RPMs should be returned.

9. Restart Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway start
   /etc/init.d/netletd start
   /etc/init.d/rwproxyd start
   ```

Verifying the Upgrade

The upgrade of Portal Server Secure Remote Access to Release 4 is verified by confirming that the upgrade patches have been properly applied. The steps for this verification were included in "Upgrade Procedure (Solaris)" on page 351 and "Upgrade Procedure (Linux)" on page 352.

In addition, you can use the following commands:

PortalServer-base`/bin/gateway version`
PortalServer-base`/bin/rwproxyd version`
PortalServer-base`/bin/netletd version`

See Table 18-3 on page 349 for output values.

Beyond these tests of the patch upgrade you can verify that what previously worked still works and that bug fixes of interest have actually been fixed.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 351 and "Upgrade Procedure (Linux)" on page 352.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Portal Server Secure Remote Access followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 4 of Portal Server Secure Remote Access is pretty much the reverse of the procedure for upgrading to Release 4. The re-configurations are rolled back and the patches are removed.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway stop
   /etc/init.d/netletd stop
   /etc/init.d/rwproxyd stop
   ```

3. Remove the patches in Table 18-4 on page 350.

   `patchrm` *patch_ID*

4. Restart Portal Server Secure Remote Access processes.

```
/etc/init.d/gateway start
/etc/init.d/netletd start
/etc/init.d/rwproxyd start
```

Multiple Instance Upgrades

In some deployment architectures Portal Server Secure Remote Access is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Portal Server Secure Remote Access components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Portal Server Secure Remote Access, you can perform a rolling upgrade in which you upgrade the Portal Server Secure Remote Access instances sequentially without interrupting service. You upgrade each instance of Portal Server Secure Remote Access while the others remain running. You perform the upgrade of each instance as described in “Release 3 Portal Server Secure Remote Access Upgrade” on page 348.

Upgrading Portal Server Secure Remote Access from Java ES Release 2

This section includes information about upgrading Portal Server Secure Remote Access from Java ES Release 2 to Java ES Release 4.

The upgrade is performed in two steps: by first upgrading Release 2 to Release 3 and then upgrading from Release 3 to Release 4. Because these two upgrade paths are distinct, this section will focus on the upgrade from Release 2 to Release 3.

Once the upgrade from Release 2 to Release 3 is complete, you can proceed with the upgrade from Release 3 to Release 4, covered in "Upgrading Portal Server Secure Remote Access from Java ES Release 3" on page 347.

This section covers the following topics regarding the upgrade from Release 2 to Release 3:

Introduction

When upgrading Java ES Release 2 Portal Server Secure Remote Access to Release 3, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 2 version. Re-configuration of Portal Server Secure Remote Access is also required and performed using an upgrade utility.
- **Upgrade Dependencies.** Portal Server Secure Remote Access has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), and all these need to be Upgraded to Release 3 because Java ES does not support a mixture of Release 2 and Release 3 components on a single computer.

  In addition, Release 3 Portal Server Secure Remote Access is dependent upon Portal Server, Access Manager, and Directory Server as described in "Portal Server Secure Remote Access Dependencies" on page 346. Portal Server Secure Remote Access has a hard upgrade dependency on Portal Server and on Access Manager (or Access Manager SDK) because it resides locally, and has a soft upgrade dependency on Directory Server, since it rarely resides locally.

- **Backward Compatibility.** Release 3 Portal Server Secure Remote Access is backwardly compatible with the Release 2 version.
- **Upgrade Rollback.** Rollback of the Release 3 upgrade of Portal Server Secure Remote Access to Release 2 can not be achieved once Portal Server Secure Remote Access re-configuration has been performed.
- **Platform Issues.** The general approach for upgrading Portal Server Secure Remote Access is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 2 Portal Server Secure Remote Access Upgrade

This section describes how to perform an upgrade of Portal Server Secure Remote Access from Java ES Release 2 to Java ES Release 3 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Portal Server Secure Remote Access you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Portal Server Secure Remote Access using the following commands:

PortalServer-base`/bin/gateway version`
PortalServer-base`/bin/rwproxyd version`
PortalServer-base`/bin/netletd version`

See Table 18-3 on page 349 for output values.

Upgrade Portal Server Secure Remote Access Dependencies

Java ES Release 3 does not support the coexistence of Release 2 and Release 3 shared components on a single computer.

You are therefore required to upgrade all local Java ES Release 2 components on which Portal Server Secure Remote Access depends to Release 3. For upgrade procedures see the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062).

When you upgrade all local Portal Server dependencies on a computer, they should be upgraded in the following order, all before you upgrade Portal Server Secure Remote Access.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 3 are provided in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062).
2. **Portal Server.** Portal Server Secure Remote Access is rarely dependent on a local Portal Server.
3. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 3 are provided in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062).
4. **Directory Server.** Portal Server is rarely dependent on a local Directory Server.

Back Up Release 2 Portal Server Secure Remote Access Configuration Information

Upgrade of Portal Server Secure Remote Access to Release 3 does requires the re-configuration of Portal Server Secure Remote Access software. As a safety measure you can back up the following directories where configuration information is stored:

PortalServerConfig-base/

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade.

Upgrading Release 2 Portal Server Secure Remote Access (Solaris)

This section discusses considerations that impact the upgrade procedure for Portal Server Secure Remote Access followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Portal Server Secure Remote Access software to Java ES Release 3 takes into account the following considerations:

- All Portal Server Secure Remote Access instances corresponding to the same installed Portal Server Secure Remote Access image are upgraded at the same time. All such instances should be shut down before patches are applied to the installed image.
- If the rewriter proxy, netlet proxy, and gateway processes are running on different computers, perform the upgrade procedure on computers hosting the rewriter and netlet proxy services before performing the upgrade on any computers hosting the gateway services.
- The Release 3 Portal Server Secure Remote Access upgrade patches for Solaris OS are the same as those used to upgrade Portal Server and are shown in the following table:

Table 18-6 Patches to Upgrade Portal Server Secure Remote Access to Release 3 on Solaris

Description	SPARC Solaris 8 & 9	X86 Solaris 9	Solaris 10
Portal Server sync-up	118195-07	118196-07	Doesn't apply
Portal Server core	118128-13	118129-13	Doesn't apply
Mobile Access core	118219-12	118219-12	Doesn't apply
Portal Server fixes	118950-15 (or higher)	118951-15 (or higher)	Doesn't apply

Upgrade Procedure (Solaris)

The procedure documented below applies to Portal Server Secure Remote Access on the computer where the upgrade is taking place.

If Portal Server Secure Remote Access is installed on the same computer as Portal Server, then follow the instructions found in "Upgrading Portal Server from Java ES Release 3" on page 315. Then proceed to Step 8 on page 360.

1. Obtain the required patches, based on Table 18-6.

   Be sure to download the exact patch revisions shown in Table 18-6

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway stop
   /etc/init.d/netletd stop
   /etc/init.d/rwproxyd stop
   ```

4. If you have not already done so, upgrade all shared components and Access Manager (or Access Manager SDK).

   See "Upgrade Portal Server Secure Remote Access Dependencies" on page 358.

5. If not yet running, start Directory Server and Access Manager (or Access Manager SDK).

6. Apply the appropriate Portal Server patches in Table 18-6.

   Be sure to apply the patches in the order in which they are shown in Table 18-6, from top to bottom.

   `patchadd` *patch_ID*

7. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

8. Re-configure Portal Server Secure Remote Access software:

   ```
   ksh

   $ cd PortalServer-base/lib
   $ ./upgradeSRA-04Q4-05Q1
   ```

9. Restart Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway start
   /etc/init.d/netletd start
   /etc/init.d/rwproxyd start
   ```

10. Re-configure Proxylet and Netlet services.

a. Logon to the Access Manager console (AMCONSOLE) as the admin user.

b. Remove Proxylet and Netlet services.

Under the Identity Management tab, select the Services option. This lists all the registered services on the left panel. From SRA Configuration, select the Proxylet and Netlet check boxes. Scroll to the top on the left panel and click the Remove button. This will remove the Proxylet and Netlet service from the ORG level.

To Verify this step manually, you may check your LDAP directory (under your organization) to make sure that the services (`srapProxylet`, `srapNetlet`) are really removed.

c. Add the services again.

Under Identity Management tab, select the Services option. Click the Add button under Services. This displays all the available services on the right panel. Choose proxylet and Netlet service check box and click OK. The newly added services will appear under SRA Configuration on your left panel.

d. Click the newly added services and build the template file. Click the Save button.

e. Add `/portal/netlet/jnlpclient.jar` and `/portal/netlet/netletjsse.jar` to non-Authenticated list of URLs under the gateway service. *

- Click the Service Configuration tab.
- Click the Gateway link under SRA Configuration. This lists all the available gateway profiles.
- Choose the appropriate profile by clicking on the link.
- Click the Security tab.
- Add `/portal/netlet/jnlpclient.jar` in the edit field under Non-authenticated URLs and click the Add button.
- Add `/portal/netlet/netletjsse.jar` in the edit field under Non-authenticated URLs and click the Add button.
- Click the Save button at the bottom of the page.

f. Restart your gateway server.

```
/etc/init.d/gateway stop
/etc/init.d/gateway start
```

11. Upgrade Portal Server Secure Remote Access from Release 3 to Release 4.

Follow the instructions in "Upgrading Portal Server Secure Remote Access from Java ES Release 3" on page 347.

Upgrading Release 2 Portal Server Secure Remote Access (Linux)

This section discusses considerations that impact the upgrade procedure for Portal Server Secure Remote Access followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Portal Server Secure Remote Access software to Java ES Release 3 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 350), except that the Linux Release 3 upgrade patches differ from the Solaris patches.

The Release 3 Portal Server Secure Remote Access upgrade patches for Linux OS are shown in the following table:

Table 18-7 Patches to Upgrade Portal Server Secure Remote Access to Release 3 on Linux

Description	Patch ID and RPM names
Portal Server core	118020-16 `sun-portal-`*`module`*`-6.3-25.i386.rpm` where *module* is any of about 70 different software modules
Mobile Access core	119515-01 • `sun-mobileaccess-1.0-25.i386.rpm` • `sun-mobileaccess-config-1.0-25.i386.rpm`
Portal Server fixes	118952-01 • `sun-portal-core-6.3-`*`xx.y`*`.i386.rpm` • `sun-portal-configurator-6.3-`*`xx.y`*`.i386.rpm` • `sun-portal-mobileaccess-6.3-`*`xx.y`*`.i386.rpm` • `sun-portal-desktop-6.3-`*`xx.y`*`.i386.rpm` • `sun-portal-sample-6.3-`*`xx.y`*`.i386.rpm` • `sun-portal-mobileaccess-config-6.3-`*`xx.y`*`.i386.rpm`

Upgrade Procedure (Linux)

The procedure documented below applies to Portal Server Secure Remote Access on the computer where the upgrade is taking place.

If Portal Server Secure Remote Access is installed on the same computer as Portal Server, then follow the instructions found in "Upgrading Portal Server from Java ES Release 3" on page 315. Then proceed to Step 8 on page 364.

1. Obtain the required patches using the patch numbers and RPM names from Table 18-6. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway stop
   /etc/init.d/netletd stop
   /etc/init.d/rwproxyd stop
   ```

4. If you have not already done so, upgrade all shared components and Access Manager (or Access Manager SDK).

   See "Upgrade Portal Server Secure Remote Access Dependencies" on page 358.

5. If not yet running, start Directory Server and Access Manager (or Access Manager SDK).

6. Apply the RPMs for Portal Server in Table 18-6.

   a. `cd /tmp`

   b. Unzip the 118020 patch file, read the `README` file and run the following script:

      ```
      ./upgradeportalrpms
      ```

      The update script installs the RPMs and also ensures that the correct configurational changes occur as the result of the patch.

   c. Unzip the 119515 patch file, and follow the instructions in its `README` file.

   d. Unzip the 118952 patch file, and follow the instructions in its `README` file.

7. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal-gateway
   rpm -qa | grep sun-mobileaccess
   ```

   The new version numbers of the RPMs should be returned.

8. Re-configure Portal Server Secure Remote Access software:

   ```
   ksh

   $ cd PortalServer-base/lib
   $ ./upgradeSRA-04Q4-05Q1
   ```

9. Restart Portal Server Secure Remote Access processes.

   ```
   /etc/init.d/gateway start
   /etc/init.d/netletd start
   /etc/init.d/rwproxyd start
   ```

10. Re-configure Proxylet and Netlet services.

    a. Logon to the Access Manager console (AMCONSOLE) as the admin user.

    b. Remove Proxylet and Netlet services.

       Under the Identity Management tab, select the Services option. This lists all the registered services on the left panel. From SRA Configuration, select the Proxylet and Netlet check boxes. Scroll to the top on the left panel and click the Remove button. This will remove the Proxylet and Netlet service from the ORG level.

       To Verify this step manually, you may check your LDAP directory (under your organization) to make sure that the services (`srapProxylet`, `srapNetlet`) are really removed.

    c. Add the services again.

       Under Identity Management tab, select the Services option. Click the Add button under Services. This displays all the available services on the right panel. Choose proxylet and Netlet service check box and click OK. The newly added services will appear under SRA Configuration on your left panel.

    d. Click the newly added services and build the template file. Click the Save button.

e. Add `/portal/netlet/jnlpclient.jar` and `/portal/netlet/netletjsse.jar` to non-Authenticated list of URLs under the gateway service. *

- Click the Service Configuration tab.
- Click the Gateway link under SRA Configuration. This lists all the available gateway profiles.
- Choose the appropriate profile by clicking on the link.
- Click the Security tab.
- Add `/portal/netlet/jnlpclient.jar` in the edit field under Non-authenticated URLs and click the Add button.
- Add `/portal/netlet/netletjsse.jar` in the edit field under Non-authenticated URLs and click the Add button.
- Click the Save button at the bottom of the page.

f. Restart your gateway server.

```
/etc/init.d/gateway stop
/etc/init.d/gateway start
```

11. Upgrade Portal Server Secure Remote Access from Release 3 to Release 4.

Follow the instructions in "Upgrading Portal Server Secure Remote Access from Java ES Release 3" on page 347.

Verifying the Upgrade

The upgrade of Portal Server Secure Remote Access to Release 3 is verified by confirming that the upgrade patches have been properly applied. The steps for this verification were included in "Upgrade Procedure (Solaris)" on page 351 and "Upgrade Procedure (Linux)" on page 352.

In addition, you can use the following commands:

PortalServer-base`/bin/gateway version`
PortalServer-base`/bin/rwproxyd version`
PortalServer-base`/bin/netletd version`

See Table 18-3 on page 349 for output values.

Beyond these tests of the patch upgrade you can verify that what previously worked still works and that bug fixes of interest have actually been fixed.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 359 and "Upgrade Procedure (Linux)" on page 363.

Rolling Back the Upgrade

The upgrade of Portal Server Secure Remote Access from Release 2 to Release 3 cannot be rolled back.

Multiple Instance Upgrades

In some deployment architectures Portal Server Secure Remote Access is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Portal Server Secure Remote Access components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Portal Server Secure Remote Access, you can perform a rolling upgrade in which you upgrade the Portal Server Secure Remote Access instances sequentially without interrupting service. You upgrade each instance of Portal Server Secure Remote Access while the others remain running. You perform the upgrade of each instance as described in "Release 3 Portal Server Secure Remote Access Upgrade" on page 348.

Chapter 19

Delegated Administrator

This chapter describes how to upgrade Delegated Administrator to Java ES 2005Q4 (Release 4): Sun Java System Communication Services Delegated Administrator 6.3 2005Q4. The chapter provides a general overview of upgrade issues and procedures for the different upgrade paths supported by Java ES Release 4. The chapter covers upgrades on both the Solaris and Linux operating systems:

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

The default value of *DelegatedAdmin-base* depends on operating system platform:

- Solaris: `/opt/SUNWcomm`
- Linux: `/opt/sun/comms/commcli`

Overview of Delegated Administrator Upgrades

This section describes the following general aspects of Delegated Administrator that impact upgrading to Java ES 2005Q4 (Release 4):

About Java ES Release 4 Delegated Administrator

Java ES Release 4 Delegated Administrator new features with respect to Release 3 include calendar service provisioning, mail group provisioning, improved UI navigation based on usability feedback, and various bug fixes.

Delegated Administrator Upgrade Roadmap

Table 19-1 shows the supported Delegated Administrator upgrade paths to Java ES Release 4. The table applies to both Solaris and Linux operating systems.

Table 19-1 Upgrade Paths to Java ES Release 4: Sun Java System Communication Services Delegated Administrator 6.3 2005Q4

Java ES Release	Delegated Administrator Version	General Approach	Re-configuration Required
Release 3	Sun Java System Communication Services Delegated Administrator 6 2005Q1	Direct upgrade: Performed by applying patches and running `config-commda` utility.	Configuration data
Release 2	Sun Java System Communication Services User Management Utility 1.1 (2004Q2)	Direct upgrade: Performed by applying patches and running `config-commda` utility.	Configuration data

Table 19-1 Upgrade Paths to Java ES Release 4: Sun Java System Communication Services Delegated Administrator 6.3 2005Q4 *(Continued)*

Java ES Release	Delegated Administrator Version	General Approach	Re-configuration Required
Release 1	User Management Utility (2003Q4)	Direct upgrade not certified: But can be performed by applying patches and running `config-commda` utility.	Configuration data
Pre-dates Java ES releases	iPlanet Delegated Administrator	No direct upgrade	

Delegated Administrator Data

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

Table 19-2 Delegated Administrator Data Usage

Type of Data	Location	Usage
Directory schema	Directory Server user/group directory	For attributes needed to support end users, organizations, and services schema
Configuration data	*DelegatedAdmin-base*`/data/WEB-INF/classes/sun/comm/cli/server/servlet/resource.properties`	Delegated Administrator server configuration and customizations
	DelegatedAdmin-base`/data/da/WEB-INF/classes/com/sun/comm/da/resources/daconfig.properties`	Delegated Administrator console configuration
	DelegatedAdmin-base`/data/da/WEB-INF/classes/com/sun/comm/da/resources/logger.properties`	Delegated Administrator console logging
	DelegatedAdmin-base`/data/da/WEB-INF/classes/com/sun/comm/da/resources/security.properties`	Delegated Administrator console configuration
Web container configuration	Web Server: `server.policy` and `server.xml` files in *WebServer-base*`/https-`*hostname*`/config` Application Server (Java ES Release 3 and 4): `server.policy` and `domain.xml` files in *AppServer8Config-base*`/domains/`*domainName*`/config` Application Server (Java ES Release 2): `server.policy` and `server.xml` files in *AppServer7Config-base*`/domains/`*domainName*`/config`	Configuration of Delegated Administrator web container instance.

Compatibility Issues

Release 4 Delegated Administrator introduces graphical user interface changes but is backwardly compatible with earlier versions.

Delegated Administrator Dependencies

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

Delegated Administrator has dependencies on the following Java ES components:

- **Shared components.** Delegated Administrator has dependencies on specific Java ES shared components (see Table 1-6 on page 40).
- **Web Container.** Delegated Administrator depends upon web container services, which can be provided either by Java ES Web Server or Java ES Application Server.
- **Directory Server.** Delegated Administrator stores application and user data in Directory Server.
- **Directory Preparation Tool.** Delegated Administrator uses the Directory Preparation Tool to prepare the directory to support Delegated Administrator user provisioning functions. As a result, Delegated Administrator upgrades might depend upon preparation of the directory to support new functions.
- **Access Manager (Access Manager SDK).** Delegated Administrator depends upon Access Manager to register services and to make entries into Directory Server.

Upgrading Delegated Administrator from Java ES Release 3

This section includes information about upgrading Delegated Administrator from Java ES 2005Q1 (Release 3) to Java ES Release 4. The section covers the following topics:

Introduction

When upgrading Java ES Release 3 Delegated Administrator to Release 4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Release 3 version. Re-configuration of Delegated Administrator is achieved by running the `config-commda` configuration utility.
- **Upgrade Dependencies.** While Delegated Administrator has dependencies on a number of Java ES shared components (see Table 1-6 on page 40), Release 4 Delegated Administrator is compatible with the Release 3 versions of these components. Upgrade of these shared components is therefore optional with respect to upgrade of Delegated Administrator to Release 4.

  In addition, Release 4 Delegated Administrator is dependent upon a web container and on Access Manager, as described in "Delegated Administrator Dependencies" on page 370. These are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Delegated Administrator to Release 4. (However, if Access Manager is upgraded, then Delegated Administrator must be upgraded also.)

  However, Release 4 Delegated Administrator has a hard upgrade dependency on Directory Preparation Tool; Release 4 Directory Preparation Tool is required to prepare Directory Server for user provisioning operations.
- **Backward Compatibility.** Release 4 Delegated Administrator is backwardly compatible with the Release 3 version.
- **Upgrade Rollback.** Rollback of the Release 4 upgrade of Delegated Administrator to Release 3 is not supported.

- **Platform Issues.** The general approach for upgrading Delegated Administrator is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Release 3 Delegated Administrator Upgrade

This section describes how to perform an upgrade of Delegated Administrator from Java ES Release 3 to Java ES Release 4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Delegated Administrator you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Delegated Administrator by entering the following command:

DelegatedAdmin-base`/bin/commadmin -V`

Table 19-3 Delegated Administrator Version Verification Outputs

Java ES Release	Delegated Administrator Version Number
Release 2	`User Management Utility 1.1`
Release 3	`User Management Utility 6 2005Q1`
Release 4	`Delegated Administrator 6.3-0.09`

Apply Necessary Operating System Patches

On Solaris 10 operating system platforms, you need to apply an operating system patch to perform the Delegated Administrator upgrade procedure (see "Required Operating System Patches" on page 31).

Upgrade Delegated Administrator Dependencies

It is generally recommended that all Java ES components on a computer system (and in a computing environment) be upgraded to Java ES Release 4. However, Delegated Administrator has a hard upgrade dependency only on Directory Preparation Tool. Upgrading of other Java ES Release 3 components upon which Delegated Administrator depends is therefore optional.

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

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in "Upgrading Java ES Shared Components" on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, "Directory Server and Administration Server" on page 103.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 6, "Web Server" on page 137 and Chapter 9, "Application Server" on page 175, respectively.
4. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 4 are provided in Chapter 11, "Access Manager" on page 203.
5. **Directory Preparation Tool.** Release 4 Directory Preparation Tool needs to have been run against Directory Server before using Release 4 Delegated Administrator. If Directory Preparation Tool has not already been run against Directory Server, upgrade Directory Preparation Tool to Release 4 and use it to modify and extend the schema of Directory Server (see Chapter 12, "Directory Preparation Tool" on page 231 for procedures).

Back Up Delegated Administrator Data

The Delegated Administrator upgrade from Release 3 to Release 4 requires re-configuration of Delegated Administrator. It is a good idea to back up configuration data as a safety precaution, and to back up any Release 3 graphical user interface customizations.

Obtain Required Configuration Information and Passwords

You should know the following information about your currently installed version:

- Access Manager administrator user ID and password
- Access Manager internal LDAP password
- Top level administrator user ID and password
- Application Server administrator user ID and password, if you are using Application Server as a web container

Upgrading Release 3 Delegated Administrator (Solaris)

This section discusses considerations that impact the upgrade procedure for Delegated Administrator followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Delegated Administrator software to Java ES Release 4 takes into account the following considerations:

- Delegated Administrator should not be used while patches are being applied to the installed image.
- When re-configuring Delegated Administrator, it should be deployed to the same web container as Access Manager.
- The Release 4 Delegated Administrator upgrade patch for Solaris OS is shown in the following table:

Table 19-4 Patches[1] to Upgrade Delegated Administrator on Solaris

Description	SPARC Solaris 8, 9, & 10	X86 Solaris 9 & 10
Delegated Administrator	119777-09	119778-09

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to Delegated Administrator on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 19-4.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   `su -`

3. If you have not already done so, upgrade all shared components, the web container, and Access Manager (or Access Manager SDK).

   See "Upgrade Delegated Administrator Dependencies" on page 373.

4. Apply the appropriate Delegated Administrator patches in Table 19-4.

   `patchadd` *patch_ID*

5. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 4.

6. Reconfigure Delegated Administrator.

   a. Make sure Directory Server is running.

   b. Make sure the web container running Access Manager and Delegated Administrator is running.

      To start the web container:

      Web Server:
      WebServer-base/https-*instanceName*/start

      Application Server:
      AppServer8-base`/bin/asadmin start-domain` *domainName*
      `--user` *admin_ID* `--password` *password*

      where Access Manager and Delegated Administrator are deployed in a server instance in the *domainName* domain. The default *domainName* is `domain1` and the default server instance is `server1`.

c. Run the Delegated Administrator configuration utility.

DelegatedAdmin-base/sbin/config-commda

For details of how to use this utility, see Chapter 3 Configuring Delegated Administrator of the *Sun Java System Communications Services 6 2005Q4 Delegated Administration Guide (*http://docs.sun.com/doc/819-2658).

7. Stop and restart the Delegated Administrator web container.

Web Server:
WebServer-base / https-*instanceName* / stop
WebServer-base / https-*instanceName* / start

Application Server:
AppServer8-base/bin/asadmin stop-domain *domainName*
AppServer8-base/bin/asadmin start-domain *domainName*
--user *admin_ID* --password *password*

Upgrading Release 3 Delegated Administrator (Linux)

This section discusses considerations that impact the upgrade procedure for Delegated Administrator followed by a description of the procedure itself.

Upgrade Considerations (Linux)

The upgrade of Delegated Administrator software to Java ES Release 4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 374), except that the Linux Release 4 upgrade patches differ from the Solaris patches.

The Release 4 Delegated Administrator upgrade patch for Linux OS is shown in the following table:

Table 19-5 Patches[1] to Upgrade Delegated Administrator on Linux

Description	Patch ID and RPM names
Delegated Administrator	119779-09 • sun-commcli-client-1.1-11.9.i386.rpm • sun-commcli-server-1.1-11.9.i386.rpm

1. Patch revision numbers are the minimum required for upgrade to Java ES Release 4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to Delegated Administrator on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 3 to Java ES Release 4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 19-5. Use this information to obtain the version numbers for the RPM.

   Patches can be downloaded to `/tmp` from:
   `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

2. Log in as root or become superuser.

   ```
   su -
   ```

3. If you have not already done so, upgrade all shared components, the web container, and Access Manager (or Access Manager SDK).

   See "Upgrade Delegated Administrator Dependencies" on page 373.

4. Apply the RPMs for Delegated Administrator in Table 19-5.

   ```
   rpm -Fvh sun-commcli-client-1.1-11.9.i386.rpm
   rpm -Fvh sun-commcli-server-1.1-11.9.i386.rpm
   ```

5. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-commcli
   ```

   The new version numbers of the RPMs should be returned.

6. Reconfigure Delegated Administrator.

   a. Make sure the web container running Access Manager and Delegated Administrator is running.

      To start the web container:

      Web Server:
      WebServer-base / https-*instanceName* / start

      Application Server:
      AppServer8-base/bin/asadmin start-domain *domainName*
      --user *admin_ID* --password *password*

      where Access Manager and Delegated Administrator are deployed in a server instance in the *domainName* domain. The default *domainName* is domain1 and the default server instance is server1.

   b. Run the Delegated Administrator configuration utility.

      DelegatedAdmin-base/sbin/config-commda

      For details of how to use this utility, see Chapter 3 Configuring Delegated Administrator of the *Sun Java System Communications Services 6 2005Q4 Delegated Administration Guide (*http://docs.sun.com/doc/819-2658).

7. Stop and restart the Delegated Administrator web container.

   Web Server:
   WebServer-base / https-*instanceName* / stop
   WebServer-base / https-*instanceName* / start

   Application Server:
   AppServer8-base/bin/asadmin stop-domain *domainName*
   AppServer8-base/bin/asadmin start-domain *domainName*
   --user *admin_ID* --password *password*

Verifying the Upgrade

You can verify successful upgrade of Delegated Administrator as follows:

1. Check the version number.

   DelegatedAdmin-base`/bin/commadmin -V`

   See Table 19-3 on page 372 for output values.

2. Log in to the Delegated Administrator console using the top level administrator user ID and password specified during re-configuration of Delegated Administrator.

   `http://`*hostName*`:`*port*`/da/DA/Login`

   where *hostName*:*port* are values provided during re-configuration of Delegated Administrator.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 375 and "Upgrade Procedure (Linux)" on page 377.

Rolling Back the Upgrade (Solaris)

Rollback of Delegated Administrator is not supported. Changes made during the upgrade procedure, such as entries in Directory Server or in deploying Delegated Administrator into the web container cannot easily be backed out.

Upgrading Delegated Administrator from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Delegated Administrator to Release 4 is the same as that for upgrading Release 3 Delegated Administrator to Release 4, with a couple of exceptions, noted below.

Upgrade Delegated Administrator Dependencies

The pre-upgrade tasks for upgrading Java ES Release 2 Delegated Administrator to Release 4 are similar to those for upgrading Release 3 Delegated Administrator to Release 4, with the exception that the upgrade of Delegated Administrator dependencies should include the upgrading to Release 4 of all shared components (see Table 1-6 on page 40) and all locally-resident product components upon which Delegated Administrator depends.

When upgrading Delegated Administrator dependencies, they should be upgraded in the following order, all before you upgrade Delegated Administrator. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 4 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 51.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 4 are provided in Chapter 4, “Directory Server and Administration Server” on page 103.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in “Upgrading Web Server from Java ES Release 2” on page 147 and “Upgrading Application Server from Java ES Release 2” on page 188, respectively.
4. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 4 are provided in “Upgrading Access Manager from Java ES Release 2” on page 224.
5. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Delegated Administrator, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in “Upgrading Directory Preparation Tool from Java ES Release 2” on page 240.

Release 2 Delegated Administrator Upgrade

The procedure for upgrading Delegated Administrator from Release 2 to Release 4 depends on the web container in which you are deploying Delegated Administrator software.

Upgrading Release 2 Delegated Administrator: Web Server Web Container

To upgrade Release 2 Delegated Administrator to Release 4, when deploying into a Web Server web container that has been upgraded to Release 4, follow the instructions in "Upgrading Release 3 Delegated Administrator (Solaris)" on page 374 or "Upgrading Release 3 Delegated Administrator (Linux)" on page 376, except substitute Release 2 wherever Release 3 is referenced.

Upgrading Release 2 Delegated Administrator: Application Server Web Container

To upgrade Release 2 Delegated Administrator to Release 4, when deploying into a Application Server web container that has been upgraded to Release 4, you first follow the instructions in "Upgrading Release 3 Delegated Administrator (Solaris)" on page 374 or "Upgrading Release 3 Delegated Administrator (Linux)" on page 376, except substitute Release 2 wherever Release 3 is referenced.

The Release 2 Application Server instance in which Delegated Administrator was originally deployed (*instanceName*), when upgraded to Release 4, was migrated under a node agent created by the upgrade process.

Upgrade of Delegated Administrator in this situation requires you need to change Step 6 "Reconfigure Delegated Administrator." on page 375 (Solaris) or page 378 (Linux) as follows:

5. Reconfigure Delegated Administrator.

a. Modify the *AccessManagerConfig-base*`/config/AMConfig.properties` file.

Replace the following line:

```
com.sun.identity.webcontainer=IAS7.0
```

with:

```
com.sun.identity.webcontainer=IAS8.1
```

b. Make sure the upgraded Application Server instance, in which Delegated Administrator is deployed (*instanceName*), is running.

To start the Application Server instance, you start the Domain Administration Server (DAS) and the node agent under which the instance was migrated:

AppServer8-base/bin/asadmin start-domain --user *admin_ID* --password *password domainName*

AppServer8-base/bin/asadmin start-node-agent --user *admin_ID* --password *password nodeagentName*

In the above commands, and in subsequent steps, the following conventions are used:

- *nodeAgentName* has the form *hostName_domainName.*
- The default *domainName* is domain1
- The default *instanceName* is server1

c. Undeploy the commcli Delegated Administrator web application from the Application Server instance (*instanceName*).

AppServer8-base/bin/asadmin undeploy --secure=false --user *admin* --password *password* --target *instanceName* commcli

d. Run the Delegated Administrator configuration utility.

DelegatedAdmin-base/sbin/config-commda

For details of how to use this utility, see Chapter 3 Configuring Delegated Administrator of the *Sun Java System Communications Services 6 2005Q4 Delegated Administration Guide (*http://docs.sun.com/doc/819-2658).

Specify the following parameters:

- When asked for the Access Manager host and port, specify the port for the DAS instance (default=8080).
- When asked where to deploy the Delegated Administrator console and Delegated Administrator server, also specify the DAS instance information (default port=8080), not the information for the upgraded Application Server instance (*instanceName*).

These parameter values will cause Delegated Administrator to be redeployed to the DAS instance. This is not standard, but it works.

e. Copy the Access Manager classpath information for `classpath-prefix` and `classpath-suffix` from the `domain.xml` file for the *instanceName* instance*:*

AppServer8Config-base`/nodeagents/`*nodeagentName*`/`*instanceName*`/config/domain.xml`

to the `domain.xml` file for the DAS (default `name=server`):

AppServer8Config-base`/domains/`*domainName*`/config/domain.xml`

where the default *domainName* is `domain1`.

This step can also be performed from the Application Server administration console.

f. Restart the DAS.

The DAS is the server instance to which Delegated Administrator has been re-deployed.

AppServer8-base`/bin/asadmin stop-domain --user` *admin_ID*
`--password` *password domainName*

AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
`--password` *password domainName*

Appendix A

Java Enterprise System Release Contents

This appendix lists the contents of the various Java Enterprise System releases. It contains the following sections:

Java ES 2003Q4 (Release 1)

This section lists the contents of Java Enterprise System 2003Q4.

Release 1 Installer-Selectable Components

The Sun Open Network Environment (Sun ONE) and Sun Cluster component products provide infrastructure services needed to support distributed enterprise applications. These are the component products:

- Sun Cluster 3.1 and Sun Cluster Agents for Sun ONE
- Sun ONE Administration Server 5.2
- Sun ONE Application Server 7, Update 1
- Sun ONE Calendar Server 6.0
- Sun ONE Directory Server 5.2
- Sun ONE Directory Proxy Server 5.2
- Sun ONE Identity Server 6.1
- Sun ONE Instant Messaging 6.1
- Sun ONE Message Queue 3.0.1 Service Pack 2
- Sun ONE Messaging Server 6.0
- Sun ONE Portal Server 6.2
- Sun ONE Portal Server, Secure Remote Access 6.2
- Sun ONE Web Server 6.1

Release 1 Shared Components

Shared components provide the local services and technology support upon which the component products depend. When you install component products, the Java ES installer automatically installs the shared components required if they are not already installed.

Java Enterprise System includes these shared components:

- ANT (Jakarta ANT Java / XML-based build tool)
- Apache Commons Logging
- ICU (International Components for Unicode)
- J2SE™ platform 1.4.1_06 (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Sun ONE Application Framework)
- JavaHelp™ Runtime
- JAXM (Java API for XML Messaging) Client Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries)
- JAX-RPC (Java APIs for XML-based Remote Procedure Call)
- JSS (Java Security Services)
- KT search engine
- LDAP C Language SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- XML C Library (libxml)

NOTE Perl is also required on your system for Application Server and Directory Server, but is not installed automatically as a Java ES shared component.

Java ES 2004Q2 (Release 2)

This section lists the contents of Java Enterprise System 2004Q2.

Release 2 Installer-Selectable Components

Component products provide infrastructure services needed to support distributed enterprise applications. When you install Java Enterprise System on a particular host, you choose which component products to install on that host based on your overall deployment architecture.

Java Enterprise System 2005Q4 includes the following component products:

Communication & Collaboration Services

- Sun Java System Messaging Server 6 2004Q2
- Sun Java System Calendar Server 6 2004Q2
- Sun Java System Instant Messaging 6 2004Q2
- Sun Java System Portal Server 2004Q2
- Sun Java System Portal Server Mobile Access 2004Q2
- Sun Java System Portal Server Secure Remote Access 2004Q2
- Sun Java System Communications Express 6 2004Q2

Web & Application Services

- Sun Java System Application Server 7.0 Update 3 (Standard and Platform Editions)
- Sun Java System Web Server 6 2004Q2 Update 1 Service Pack 2
- Sun Java System Message Queue 3.5 SP1 (Platform and Enterprise Editions)

Directory & Identity Services

- Sun Java System Identity Server 6.2 2004Q2, including
  Sun Java System Communications Services 6 2004Q2 User Management Utility
- Sun Java System Directory Server 5 2004Q2
- Sun Java System Directory Proxy Server 5 2004Q2

Availability Services

- Sun Cluster 3.1 4/04 and Sun Cluster Agents for Sun Java System

Administrative Services

- Sun Java System Administration Server 5 2004Q2
- Sun Remote Services Net Connect 3.5

Note that Sun Cluster, Sun Cluster Agents, and Sun Remote Services Net Connect are not available on the Linux operating system.

Release 2 Shared Components

Shared components provide the local services and technology support upon which the component products depend. When you install component products, the Java ES installer automatically installs the shared components required if they are not already installed.

Java Enterprise System 2005Q4 includes these shared components:

- Ant (Jakarta ANT Java/XML-based build tool)
- Apache Commons Logging
- Apache SOAP (Simple Object Access Protocol)
- ICU (International Components for Unicode)
- J2SE™ platform 1.4.2_04 (Java 2 Platform, Standard Edition)
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Application Framework)
- JavaHelp™ Runtime
- JAXB (Java Architecture for XML Binding)
- JAXM (Java API for XML Messaging) Client Runtime

- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries)
- JAX-RPC (Java APIs for XML-based Remote Procedure Call)
- JCAPI (Java Calendar API)
- JSS (Java Security Services)
- KT search engine
- LDAP C Language SDK
- LDAP Java SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- Perl LDAP, including NSPERL
- SAAJ (SOAP with Attachments API for Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector
- XML C Library (libxml)

Java ES 2005Q1 (Release 3)

This section lists the contents of Java Enterprise System 2005Q1.

Release 3 Installer Selectable Components

In the component selection page of the Java ES installer, the selectable components are grouped by the services they help to provide. The following list also shows the subcomponents that are installed with each component.

Communication & Collaboration Services

- Sun Java System Messaging Server 6 2005Q1
- Sun Java System Calendar Server 6 2005Q1
- Sun Java System Instant Messaging 7 2005Q1
  - Instant Messaging Server Core; includes server and multiplexor software
  - Instant Messaging Resources
  - Access Manager Instant Messaging Service
- Sun Java System Portal Server 6 2005Q1
- Sun Java System Portal Server Secure Remote Access 6 2005Q1
  - Secure Remote Access Core
  - Gateway
  - Netlet Proxy
  - Rewriter Proxy
- Sun Java System Communications Express 2005Q1
- Sun Java System Directory Preparation Tool

Web & Application Services

- Sun Java System Application Server Enterprise Edition 8.1 2005Q1
  - Domain Administration Server
  - Application Server Node Agent
  - Command Line Administration Tool
  - Load Balancing Plugin

    Can be used with either Web Server or Apache Web Server, selectable at configuration. Default is Web Server.
  - PointBase
  - Sample Applications
- Sun Java System Web Server 6 2005Q1 Update 1 Service Pack 4
- Sun Java System Message Queue 3 2005Q1

Directory & Identity Services

- Sun Java System Access Manager 6.3 2005Q1

  Delegated Administrator provisioning tools for Portal Server and Messaging Server are automatically installed with Access Manager.
  - Identity Management and Policy Services Core (includes Delegated Administrator Utility)
  - Access Manager Administration Console
  - Common Domain Services for Federation Management
  - Access Manager SDK
- Sun Java System Directory Server 5 2005Q1
- Sun Java System Directory Proxy Server 5 2005Q1

Availability Services

- Sun Cluster 3.1 9/04
  - Sun Cluster Core
- Sun Cluster Agents for Sun Java System
  - HA/Scalable Sun Java System Web Server
  - HA Sun Java System Message Queue
  - HA Sun Java System Portal Server
  - HA Sun Java System Administration Server
  - HA Sun Java System Directory Server
  - HA Sun Java System Messaging Server
- HADB (used for high availability session storage)

Administrative Services

- Sun Java System Administration Server 5 2005Q1
- Sun[SM] Remote Services Net Connect 3.1.1

NOTE Sun Cluster, Sun Cluster Agents, and Sun Remote Services Net Connect are not available on the Solaris 10 or Linux operating systems.

Sun Remote Services Net Connect is not available on the Solaris x86 platform.

Release 3 Shared Components

Shared components provide the local services and technology support for the selectable components. When you install Java ES components, the installer automatically installs the shared components required if they are not already installed.

This release of Java ES includes these shared components:

- Ant (Jakarta ANT Java / XML-based build tool)
- Apache SOAP (Simple Object Access Protocol) Runtime
- Berkeley Database
- Common agent container
- ICU (International Components for Unicode)
- J2SE™ (Java 2 Platform, Standard Edition) platform 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail ™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)

- Perl LDAP, including NSPERL
- SAAJ (SOAP with Attachments API for Java)
- SAML (Security Assertions Markup Language)
- SASL (Simple Authentication and Security Layer)
- SNMP (Simple Network Management Protocol) Peer
- Sun Explorer Data Collector (Solaris only)
- Sun Java Monitoring Framework
- Sun Java Web Console
- Tomcat Servlet JSP Container
- XML C Library (`libxml`)
- WSCL (Web services Common Library)

Java ES 2005Q4 (Release 4)

This section lists the contents of Java Enterprise System 2005Q4.

Release 4 Installer-Selectable Components

In the component selection page of the Java ES installer, the selectable components are grouped by the services they help to provide. The following list also shows the subcomponents that are installed with each component.

Communication & Collaboration Services

- Sun Java System Messaging Server 6.2 2005Q4
- Sun Java System Calendar Server 6.2 2005Q4
- Sun Java System Instant Messaging 7.0.1 2005Q4
  - Instant Messaging Server Core; includes server and multiplexor software
  - Instant Messaging Resources
  - Access Manager Instant Messaging Service
- Sun Java System Portal Server 6.3.1 2005Q4
- Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4
  - Secure Remote Access Core
  - Gateway
  - Netlet Proxy
  - Rewriter Proxy
- Sun Java System Communications Express 6.2 2005Q4
- Sun Java System Directory Preparation Tool 6.3 2005Q4
- Sun Java System Communications Services Delegated Administrator 6.3 2005Q4
  - Delegated Administrator Console and Utility
  - Delegated Administrator Server

Web & Application Services

- Sun Java System Application Server Enterprise Edition 8.1 2005Q4
  - Domain Administration Server
  - Application Server Node Agent
  - Command Line Administration Tool
  - Load Balancing Plugin

    Can be used with either Web Server or Apache Web Server, selectable at configuration. Default is Web Server.
  - PointBase Database
  - Sample Applications
- Sun Java System Web Server 6.1 Service Pack 5 2005Q4
- Sun Java Web Proxy Server 4.0.1 2005Q4
- Sun Java System Message Queue Enterprise Edition 3.6 SP3 2005Q4
- Sun Java Service Registry 3.0

Directory & Identity Services

- Sun Java System Access Manager 7.0 2005Q4
  - Identity Management and Policy Services Core
  - Access Manager Administration Console
  - Common Domain Services for Federation Management
  - Access Manager SDK
- Sun Java System Directory Server 5.2 2005Q4
- Sun Java System Directory Proxy Server 5.2 2005Q4

Availability Services

- Sun Cluster 3.1 8/05
  - Sun Cluster Core
  - Sun Cluster Agents for Sun Java System
    - HA Sun Java System Directory Server
    - HA Sun Java System Administration Server
    - HA/Scalable Sun Java System Web Server
    - HA Sun Java System Message Queue
    - HA Sun Java System Application Server
    - HA Sun Java System Messaging Server
    - HA Sun Java System Calendar Server
    - HA Sun Java System Instant Messaging
- High Availability Session Store (HADB) 4.4.2

Administrative Services

- Sun Java System Administration Server 5.2 2005Q4

NOTE Sun Cluster, Sun Cluster Agents, and Sun Remote Services Net Connect are not available on the Solaris 10 or Linux operating systems.

Sun Remote Services Net Connect is not available on the Solaris x86 platform.

Release 4 Shared Components

Shared components provide the local services and technology support for the selectable components. When you install Java ES components, the installer automatically installs the shared components required if they are not already installed.

This release of Java ES includes these shared components:

- ANT (Jakarta ANT Java / XML-based build tool)
- ACL (Apache Commons Logging)
- BDB (Berkeley Database)
- CAC (Common agent container)
- Derby Database
- ICU (International Components for Unicode)
- IM-SDK (Instant Messenger SDK)
- J2SE™ (Java 2 Platform, Standard Edition) platform 5.0
- JAF (JavaBeans™ Activation Framework)
- JATO (Java Studio Enterprise Web Application Framework)
- JavaHelp™ Runtime
- JavaMail™ Runtime
- JAXB (Java Architecture for XML Binding) Runtime
- JAXP (Java API for XML Processing)
- JAXR (Java API for XML Registries) Runtime
- JAX-RPC (Java API for XML-based Remote Procedure Call) Runtime
- JCAPI (Java Calendar API)
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- KTSE (KT Search Engine)
- LDAP C SDK
- LDAP Java SDK

- MA (Mobile Access) Core
- NSPR (Netscape Portable Runtime)
- NSS (Network Security Services)
- SAAJ (SOAP runtime with Attachments API for Java)
- SASL (Simple Authentication and Security Layer)
- SEDC (Sun Explorer Data Collector, Solaris only)
- MFWK (Java ES Monitoring Framework)
- SJWC (Sun Java Web Console)
- WSCL (Web services Common Library)

Index

C

D

H

I

J

K

L

M

N

P

S

T

U

W